Calibre-Benutzerhandbuch

Release 8.7.0

Kovid Goyal

Juli 18, 2025

Inhaltsverzeichnis

1 Die grafische Benutzeroberfläche 3

2 Hinzufügen Ihrer bevorzugten Nachrichtenwebsite 33

3 eBook-Betrachter 53

4 eBook Konvertierung 61

5 Bearbeiten von eBooks 79

6 Der Calibre Inhalteserver 113

7 Vergleichen von eBooks 121

8 Bearbeiten von eBook Metadaten 125

9 Häufig gestellte Fragen (FAQ) 129

10 Tutorials 153

11 Das calibre:// URL Schema 251

12 Anpassen von Calibre 255

13 Befehlszeilenschnittstelle 303

14 Einrichten einer Calibre-Entwicklungsumgebung 345

15 Digitale Rechteverwaltung (DRM) 375

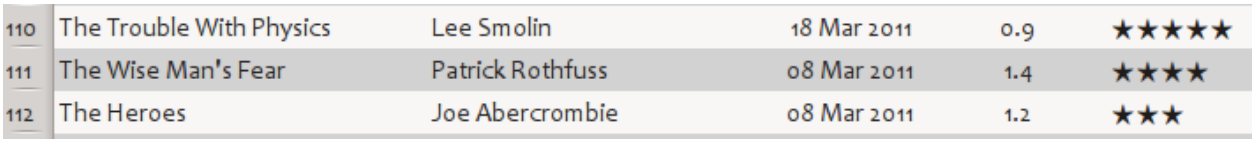
16 Glossar 377

Python-Modulindex 379

Stichwortverzeichnis 381

Calibre ist ein e-Book Bibliotheksverwalter. Er kann E-Bücher in den meisten gängigen E-Book-Formaten anzeigen, konvertieren und katalogisieren. Er kann auch mit vielen E-Book Lesegeräten kommunizieren. Er kann ins Internet gehen und Metadaten für Ihre Bücher holen. Er kann Zeitungen herunterladen und in E-Books umwandeln, um sie bequem zu lesen. Er ist plattformübergreifend und läuft unter Linux, Windows und macOS.

Sie haben Calibre gerade gestartet. Was müssen Sie jetzt machen? Bevor Calibre etwas mit ihren eBooks anfangen kann, muss es sie zuerst einmal kennen. Nehmen Sie einige eBook-Dateien und ziehen Sie diese mit der Maus nach Calibre ins Hauptfenster oder klicken Sie auf die Schaltfläche „Bücher hinzufügen“ und suchen Sie nach eBooks, mit denen Sie arbeiten möchten. Sobald diese Bücher hinzugefügt wurden, sind diese in der Hauptansicht, ähnlich wie hier angezeigt, sichtbar:



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	★★★

Nachdem Sie eine Reihe von der Bücher nach Herzenslust hinzugefügt haben, möchten Sie vielleicht eins davon lesen. Um dies zu können, müssen Sie das Buch in ein Format konvertieren, welches Ihr Lesegerät versteht. Falls Calibre zum ersten Mal ausgeführt wurde, startet der *Willkommensassistent* und wird Calibre für Ihr Lesegerät einrichten. Die Konvertierung ist ein Kinderspiel. Wählen Sie einfach das Buch, welches Sie konvertieren möchten, aus und klicken Sie auf die Schaltfläche „Bücher konvertieren“. Ignorieren Sie zunächst alle Optionen und klicken auf „OK“. Das kleine Symbol in der rechten unteren Ecke beginnt sich zu drehen. Sobald es sich nicht mehr dreht, ist das Buch fertig konvertiert. Klicken Sie auf „Bücher öffnen“, um das Buch zu lesen.

Wenn Sie das Buch auf Ihrem Reader lesen möchten, verbinden Sie den Reader mit dem Computer, warten Sie bis Calibre den Reader erkannt hat (10-20 Sekunden) und klicken Sie dann auf die Schaltfläche „An Gerät senden“. Sobald sich das kleine Symbol unten rechts nicht mehr dreht, können Sie den Reader vom Computer trennen und mit dem Lesen anfangen! Falls Sie das Buch im vorherigen Schritt nicht konvertiert hatten, wird Calibre es automatisch vor dem Übertragen in das für das Lesegerät verständliche Format konvertieren.

Um die weitergehenden Möglichkeiten zu nutzen, sollten sie zuvor mehr über *Die grafische Benutzeroberfläche* (die Titelseite 3) lesen. Für noch mehr Leistung und Vielseitigkeit ist das Erlernen der *Befehlszeilenschnittstelle* (die Titelseite 303) nützlich. Ebenfalls hilfreich sind Empfehlungen der *Häufig gestellte Fragen (FAQ)* (die Titelseite 129).

Wenn Sie weitere Fragen haben, mit anderen Benutzern über Calibre diskutieren oder um Hilfe bei spezifischen Fragen bitten möchten, stehen Ihnen Foren und andere Hilfe[1] zur Verfügung.

Abschnitte

[1] https://calibre-ebook.com/help

KAPITEL 1

Die grafische Benutzeroberfläche

The Graphical User Interface *(GUI)* provides access to all library management and e-book format conversion features. The basic workflow for using calibre is to first add books to the library from your hard disk. calibre will automatically try to read metadata from the books and add them to its internal database. Once they are in the database, you can perform various *Aktionen* (die Titelseite 4) on them that include conversion from one format to another, transfer to the reading device, viewing on your computer, and editing metadata. The latter includes modifying the cover, description, and tags among other details. Note that calibre creates copies of the files you add to it. Your original files are left untouched.

Die Oberfläche ist in verschiedene Abschnitte aufgeteilt:

- *Aktionen* (die Titelseite 4)
- *Einstellungen* (die Titelseite 11)
- *Kataloge* (die Titelseite 11)
- *Suchen & Sortieren* (die Titelseite 12)
- *Die Suchoberfläche* (die Titelseite 12)
- *Speichern von Suchen* (die Titelseite 17)
- *Searching the full text of all books* (die Titelseite 18)
- *Virtuelle Bibliotheken* (die Titelseite 19)
- *Temporarily marking books* (die Titelseite 19)
- *Extrahieren von Metadaten aus dem Dateinamen* (die Titelseite 19)
- *Über dieses Buch* (die Titelseite 20)
- *Schlagwortbrowser* (die Titelseite 22)
- *Titelbildraster* (die Titelseite 25)
- *Titelbildbrowser* (die Titelseite 26)

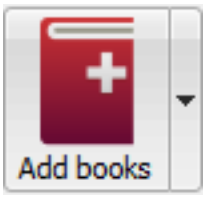
- *Adding notes for authors, series, etc.* (die Titelseite 27)
- *Schnellansicht* (die Titelseite 28)
- *Aufträge* (die Titelseite 29)
- *Tastaturkurzbefehle* (die Titelseite 29)

1.1 Aktionen

The actions toolbar provides convenient shortcuts to commonly used actions. If you right-click the buttons, you can perform variations on the default action. Please note that the actions toolbar will look slightly different depending on whether you have an e-book reader attached to your computer.

- *Bücher hinzufügen* (die Titelseite 4)
- *Metadaten bearbeiten* (die Titelseite 5)
- *Bücher konvertieren* (die Titelseite 6)
- *Bücher öffnen* (die Titelseite 6)
- *An Gerät senden* (die Titelseite 7)
- *Nachrichten abrufen* (die Titelseite 7)
- *Bibliothek* (die Titelseite 8)
- *Gerät* (die Titelseite 9)
- *Bücher exportieren* (die Titelseite 9)
- *Verbinden/Teilen* (die Titelseite 10)
- *Bücher entfernen* (die Titelseite 10)

1.1.1 Bücher hinzufügen



Die Aktion *Bücher hinzufügen* hat sieben Variationen, die mit einem Rechtsklick auf die Schaltfläche aufgerufen werden können.

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* (die Titelseite 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* (die Titelseite 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. **Add multiple books from archive (ZIP/RAR)**: Allows you to add multiple e-books that are stored inside the selected ZIP or RAR files. It is a convenient shortcut that avoids having to first unzip the archive and then add the books via one of the above two options.
4. **Add empty book (Book Entry with no formats)**: Allows you to create a blank book record. This can be used to then manually fill out the information about a book that you may not have yet in your collection.
5. **Von ISBN hinzufügen**: Sie können eines oder mehrere Bücher hinzufügen, indem Sie deren ISBN-Nummern eingeben.
6. **Dateien zum ausgewählten Buch hinzufügen**: Sie können hierüber Dateien zu einem bestehenden Buch in Ihrer Bibliothek hinzufügen oder diese aktualisieren.
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* (die Titelseite 128) for details.
8. **Add an empty file to selected book records**: Allows you to add an empty file of the specified format to the selected book records.

The *Add books* action can read metadata from a wide variety of e-book formats. In addition, it tries to guess metadata from the filename. See the *Extrahieren von Metadaten aus dem Dateinamen* (die Titelseite 19) section, to learn how to configure this.

Um ein zusätzliches Format zu einem bestehenden Buch hinzufügen zu können, nutzen sie eine der drei Möglichkeiten:

1. Ziehen Sie die Datei in den Buchdetailsbereich auf der rechten Seite des Hauptfensters
2. Right click the *Add books* button and choose *Add files to selected books*.
3. Click the *Add books* button in the top right area of the *Edit metadata* dialog, accessed by the *Metadaten bearbeiten* (die Titelseite 5) action.

1.1.2 Metadaten bearbeiten

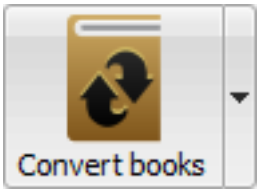
Die Aktion *Metadaten bearbeiten* hat vier Variationen, die mit einem Rechtsklick auf die Schaltfläche aufgerufen werden können.

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. **Metadaten gleichzeitig bearbeiten**: Sie können gemeinsame Metadatenfelder für eine große Anzahl von Büchern gleichzeitig bearbeiten. Es bearbeitet alle Bücher, die Sie in der *Bibliotheksansicht* (die Titelseite 12) ausgewählt haben.
3. **Metadaten und Titelbilder herunterladen**: Lädt Metadaten und Titelbilder (falls verfügbar) für die Bücher, die in der Buchliste ausgewählt wurden.

4. **Bucheinträge zusammenführen**: Diese Option gibt Ihnen die Möglichkeit Metadaten und Formate von zwei oder mehr Büchern zusammenzuführen. Die Datensätze der Büchern, die nicht zuerst angeklickt wurden, entweder löschen oder behalten.
5. **Manage data files**: Manage the extra data files associated with the selected books.

For more details, see *Bearbeiten von eBook Metadaten* (die Titelseite 125).

1.1.3 Bücher konvertieren



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* (die Titelseite 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.

In den meisten Fällen sollte die Konvertierung eine einfache Ein-Klick Angelegenheit sein. Wenn Sie mehr über den Konvertierungsprozess erfahren möchten, finden Sie unter *eBook Konvertierung* (die Titelseite 61) weitere Informationen.

Die Aktion *Bücher konvertieren* hat drei Variationen, die mit einem Rechtsklick auf die Schaltfläche aufgerufen werden können.

1. **Convert individually**: Allows you to specify conversion options to customize the conversion of each selected e-book.
2. **Mehrere gleichzeitig konvertieren**: Sie können die Optionen der Konvertierung einmalig für alle ausgewählte eBooks anpassen und dann alle auf einmal konvertieren.
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 *Erstellen von AZW3-, EPUB- und MOBI-Katalogen* (die Titelseite 244).

1.1.4 Bücher öffnen

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 An Gerät senden

Die Aktion *An Gerät senden* hat acht Variationen, die mit einem Rechtsklick auf die Schaltfläche aufgerufen werden können.

1. **An Hauptspeicher senden**: Die ausgewählten Bücher werden in den Hauptspeicher des eBook-Readers übertragen.
2. **An Speicherkarte (A) senden**: Die ausgewählten Bücher werden auf die Speicherkarte (A) des eBook-Readers übertragen.
3. **An Speicherkarte (B) senden**: Die ausgewählten Bücher werden auf die Speicherkarte (B) des eBook-Readers übertragen.
4. **Bestimmtes Format senden an**: Die ausgewählten Bücher werden an den ausgewählten Speicherort des eBook-Readers im angegeben Format übertragen.
5. **Gerät auswerfen**: trennt das Gerät von Calibre und meldet den eBook-Reader sicher vom Computer ab.
6. **Standardübertragungsart festlegen**: Sie können angeben, welche der Optionen, 1 bis 5 oben oder 7 unten, die Standardaktion sein soll, wenn Sie auf die Hauptschaltfläche anklicken.
7. **Senden und aus Bibliothek löschen**: Die ausgewählten Bücher werden an den ausgewählten Speicherort des eBook-Readers übertragen und dann aus der Bibliothek **gelöscht**.
8. **Anmerkungen abrufen (experimentell)**: Überträgt die Anmerkungen, die Sie vielleicht im eBook auf ihrem eBook-Reader gemacht haben, an die Kommentar-Metadaten des Buchs in der Calibre-Bibliothek.

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 *Die Calibre-Vorlagensprache* (die Titelseite 160).

1.1.6 Nachrichten abrufen

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.

Die Aktion *Nachrichten herunterladen* verwendet einfache Schemata (10-15 Zeilen Code) für jede Nachrichtenwebsite. Für weitere Informationen zur Erstellung von Schemata für eigene Nachrichten-Quellen lesen Sie bitte das Tutorial zum *Hinzufügen Ihrer bevorzugten Nachrichtenwebsite* (die Titelseite 33).

Die Aktion *Nachrichten herunterladen* hat drei Variationen, die mit einem Rechtsklick auf die Schaltfläche aufgerufen werden können.

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 *Hinzufügen Ihrer bevorzugten Nachrichtenwebsite* (die Titelseite 33).

3. **Alle geplanten Nachrichtenquellen herunterladen**: Veranlasst Calibre sofort mit downloaden aller geplanten Nachrichtenquellen zu beginnen.

1.1.7 Bibliothek

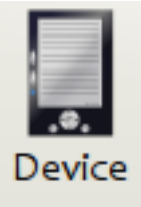
Die Aktion *Bibliothek* ermöglicht es, eine Bibliothek zu erstellen, zwischen ihnen zu wechseln, deren Name zu verändern oder sie zu löschen. In Calibre können Sie beliebig viele Bibliotheken erstellen. Sie können Bibliotheken nach beliebigen Kriterien, wie z. B. Belletristik-, Sachbuch-, Fremdsprachen-, Projektbibliothek erstellen oder nach jeder beliebigen anderen Strukturen definieren. Bibliotheken sind die höchsten Organisationsstrukturen innerhalb Calibre. Jede Bibliothek hat seine eigenen Bücher, Tags, Kategorien und Speicherort.

1. **Bibliothek wechseln/neu erstellen …**: Ermöglicht es: a) eine bereits vorhandene Bibliothek von einem anderen Ort in Calibre einzubinden, b) eine leere Bibliothek an einem neuen Speicherort zu erstellen oder c) eine aktuelle Bibliothek zu einem neuen Speicherort zu verschieben.
2. **Schneller Wechsel**: Ermöglicht das Umschalten zwischen Bibliotheken, die in Calibre erstellt oder eingebunden wurden.
3. **Bibliothek umbenennen**: Ermöglicht die Umbenennung einer Bibliothek.
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. **Bibliothekswartung**: Ermöglicht die Überprüfung der Datenintegrität der aktuellen Bibliothek und die Wiederherstellung von Metadaten der aktuellen Bibliotheksdatenbank aus der internen Metadatensicherung.

> **Bemerkung**
>
> 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.

Bücher können zwischen verschiedenen Bibliotheken durch Rechtsklick auf das Buch und Auswahl der Aktion *In Bibliothek kopieren* kopiert oder verschoben werden (sobald Sie mehr als eine Bibliothek erstellt oder eingebunden haben).

1.1.8 Gerät



Die Aktion *Gerät* ermöglicht die Anzeige von Bücher, die sich im Hauptspeicher oder auf integrierten Speicherkarten befinden oder die Möglichkeit das Gerät auszuwerfen (von Calibre und Computer sicher trennen). Dieses Symbol wird automatisch in der Hauptwerkzeugleiste von Calibre angezeigt, sobald ein unterstütztes Gerät angeschlossen wird. Wenn mehrere Speicher im Gerät vorhanden sind, werden mehrere Symbole angezeigt. Klicken Sie auf das Symbol, um die im Gerät gespeicherten Bücher zu sehen. Sie können auch mit der Maus Bücher aus Ihrer Calibre-Bibliothek auf das Symbol ziehen, um sie auf Ihr Gerät übertragen. Umgekehrt können Sie Bücher vom Gerät auf das Bibliothekssymbol in der Symbolleiste ziehen, um Bücher von Ihrem Gerät in die Calibre-Bibliothek zu übertragen.

1.1.9 Bücher exportieren

Die Aktion *Bücher exportieren* hat fünf Variationen, die durch Rechtsklick auf die Schaltfläche abgerufen werden können.

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 *Die Calibre-Vorlagensprache* (die Titelseite 160).

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

Für 1. und 2. werden für jedes ausgewählte Buch alle verfügbaren Formate sowie Metadaten auf Datenträger gespeichert. Metadaten werden in einer OPF-Datei gespeichert. Gespeicherte Bücher können erneut ohne Verlust an Informationen in die Bibliothek mithilfe der Aktion *Bücher hinzufügen* (die Titelseite 4) importiert werden.

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 Verbinden/Teilen

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. **Mit Ordner verbinden**: Ermöglicht die Verbindung zu jedem beliebigen Ordner auf Ihrem Computer und stellt den Ordner wie ein Gerät zur Nutzung aller Möglichkeiten von Calibre zur Verfügung. Dies ist hilfreich, falls Ihr Gerät nicht von Calibre unterstützt wird jedoch als USB-Festplatte/Massenspeicher zugänglich ist.
2. **Inhalteserver starten**: Calibres integrierten Webserver starten. Sie können (falls ausgewählt) auf Ihre Calibre-Bibliothek über den Webbrowser via Internet zugreifen, sobald dieser gestartet wurde. Der Zugriff auf den Webserver wird über *Einstellungen → Versand → Netzwerkserver* eingestellt.
3. **E-Mail-Versand von Büchern einrichten**: Ermöglicht den Austausch von Büchern und Newsfeeds per E-Mail. Nach der Einrichtung von E-Mail-Adressen für diese Option sendet Calibre Nachrichten und Buchaktualisierungen an die eingegebene E-Mail-Adressen. Wie Calibre E-Mails versendet, wird über *Einstellungen → Versand → Bücherversand per E-Mail* eingestellt. Nachdem Sie eine oder mehrere E-Mail-Adressen eingerichtet haben, wird dieser Menüeintrag durch Menüeinträge Bücher an die konfigurierten E-Mail-Adressen senden ersetzt.

1.1.11 Bücher entfernen

The *Remove books* action **deletes books permanently**, so use it with care. It is *context sensitive*, i.e. it depends on which *catalog* (die Titelseite 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 *Metadaten bearbeiten* (die Titelseite 5) action. Remove books also has five variations which can be accessed by doing a right-click on the button.

1. **Ausgewählte Bücher entfernen**: Löscht **dauerhaft** alle Bücher, die in der Buchliste ausgewählt wurden.
2. **Bestimmte Dateiformate aus ausgewählten Büchern entfernen**: Löscht **dauerhaft** eBook-Dateien eines bestimmten Formats aus Büchern, die aus der Buchliste ausgewählt wurden.
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. **Titelbilder der ausgewählten Bücher entfernen**: löscht **dauerhaft** Titelbild-Dateien aus Bücher, die aus der Buchliste ausgewählt wurden.
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.

Bemerkung

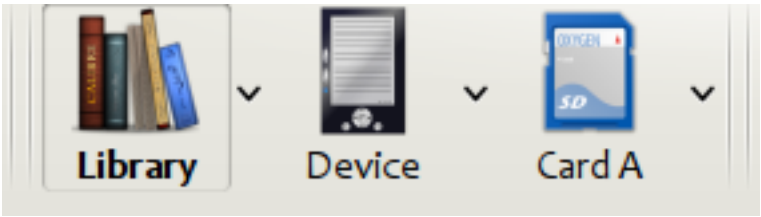
Note that when you use *Remove books* to delete books from your calibre library, the book record is deleted, but the books are temporarily stored, for a few days, in a trash folder. You can undo the delete by right clicking the *Remove books* button and choosing to *Restore recently deleted* books.

1.2 Einstellungen

Über die Aktion *Einstellungen* können Sie die vielfältigen Einstellungen und das Verhalten von Calibre anpassen. Sie hat vier Variationen, die durch Rechtsklick auf die Schaltfläche aufgerufen werden können.

1. **Verhalten von Calibre ändern**: Hierüber haben Sie Zugriff auf die vielfältigen Einstellungen und das Verhalten von Calibre. Ein Klick auf die Schaltfläche *Einstellungen* führt die gleiche Aktion aus.
2. **Run Welcome wizard**: Allows you to start the *Welcome wizard* which appeared the first time you started calibre.
3. **Calibre durch Erweiterungen verbessern**: Öffnet ein neues Fenster mit zur Verfügung stehenden Anwender-Erweiterungen für Calibre. Diese Erweiterungen werden von Drittanbietern zur Erweiterung von calibres Funktionsumfang entwickelt.
4. **Neustart im Fehlerdiagnosemodus**: Aktiviert den Fehlerdiagnosemodus, mit dessen Hilfe Calibres Entwickler zur Lösung von aufgetretenen Problemen im Programm unterstützen werden können. Diese Option sollte nur aktiviert werden, wenn ein Entwickler Sie dazu auffordert.

1.3 Kataloge



Ein *Katalog* ist eine Sammlung von Büchern. Calibre kann zwei Arten von Katalogen verwalten:

1. **Bibliothek**: Dies ist eine Sammlung von Büchern, die in Ihrer Calibre-Bibliothek auf dem Computer gespeichert wurde.
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 Suchen & Sortieren

Im Abschnitt Suche & Sortieren können Sie vielfältige leistungsstarke Aktionen innerhalb der Büchersammlungen ausführen.

- Sie können diese durch Klicken auf die Spaltentitel nach Titel, Autor, Datum, Bewertung, etc. sortieren. Auch Untersortierungen, d. h. über mehrere Spalten sortieren, sind möglich. Wenn Sie beispielsweise auf den Titel- und danach der Autor-Spalte klicken, werden die Bücher zuerst nach Autoren sortiert und danach alle Einträge des gleichen Autors nach Titel sortiert.
- 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 *Aktionen* (die Titelseite 4) on sets of books. To select multiple books you can either:
  - Bei gedrückter `Strg`-Taste nacheinander auf die gewünschten Bücher klicken.
  - Mit gedrückter `Umschalt`-Taste auf das erste und letzte Buch einer Reihe von hintereinander liegenden Büchern klicken.
- Sie können die angezeigten Felder mithilfe des Dialogs *Einstellungen* (die Titelseite 11) nach ihren Wünschen gestalten.
- To perform complex multiple column based sub-sorting add the *Sort by* tool to a toolbar via *Preferences → Toolbars & menus*.

1.5 Die Suchoberfläche

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

```
Asimov Foundation format:lrf
```

Diese Suche wird alle Bücher in der Bibliothek finden, die `Asimov` und `Foundation` in ihren Metadaten enthalten und als LRF-Format verfügbar sind. Einige weitere Beispiele:

```
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[2]), 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.
- `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.

[2] https://en.wikipedia.org/wiki/Regular_expression

- `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[3] 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.

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`

[3] https://docs.python.org/library/re.html

- `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

Die Syntax für die Suche nach einem Datum ist:

```
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
```

Um mögliche Probleme mit übersetzten Zeichenketten bei nicht englischen Calibre-Versionen zu vermeiden, sind die Bezeichnungen `_today`, `_yesterday`, `_thismonth`, und `_daysago` immer verfügbar. Diese werden nicht übersetzt.

Searching dates and numeric values with relational comparisons

Datums- und numerische Felder unterstützen die relationalen Operatoren = (gleich), > (größer als), >= (größer als oder gleich), < (kleiner), <= (kleiner als oder gleich), und != (ungleich). Bewertungsfelder sind numerisch. Beispielsweise findet die Suche `rating:>=3` alle Bücher mit der Bewertung 3 oder höher.

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* (die Titelseite 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 `vl` is used to search for books in a Virtual library. 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.

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* (die Titelseite 13). Examples:

- `identifiers:true` findet alle Bücher mit einer beliebigen Kennung.
- `identifiers:false` findet Bücher ohne Kennung.
- `identifiers:123` sucht nach Büchern mit beliebiger Kennung, die den Wert *123* beinhalten.
- `identifiers:=123456789‘` sucht nach Bücher mit beliebiger Kennung und einem Wert gleich *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` findet Bücher, deren Typ ein *i* und deren Wert eine *1* beinhalten.

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 *Die Calibre-Vorlagensprache* (die Titelseite 160) 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* (die Titelseite 13) described above for the various search types. You must quote the entire search string if there are spaces anywhere in it.

Beispiele

- `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

1.6 Speichern von Suchen

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

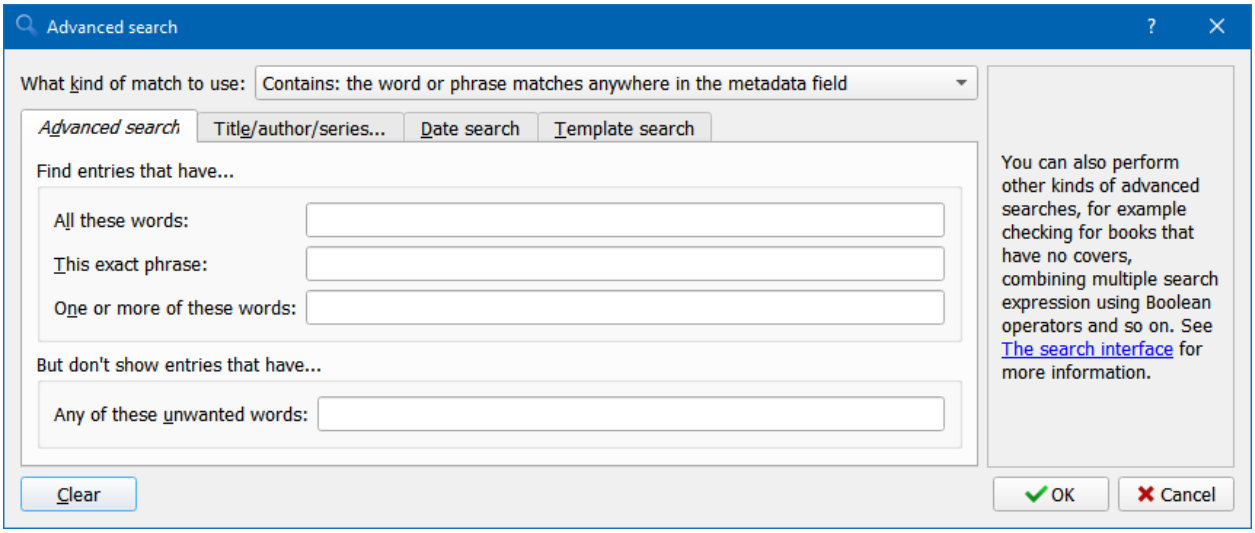
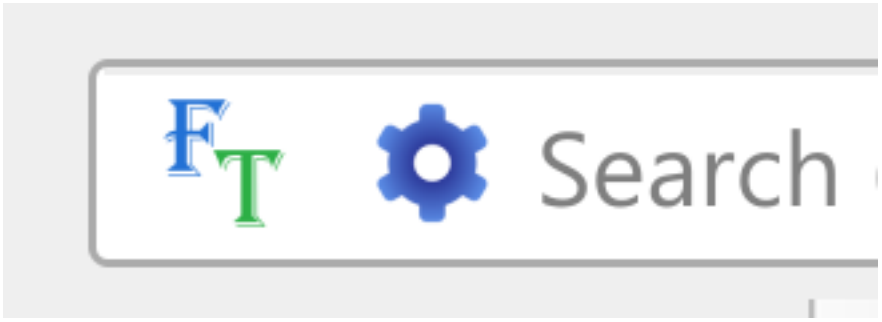


Abb. 1: *Erweiterte Suche Dialog*

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 Virtuelle Bibliotheken

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: *Virtuelle Bibliotheken* (die Titelseite 248).

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 Extrahieren von Metadaten aus dem Dateinamen

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 *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214)). 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
```

es wird also davon ausgegangen, dass alle Zeichen bis zum ersten - der Titel des Buches ist und die nachfolgenden Zeichen zum Autor des Buches gehören. Als Beispiel der Dateiname:

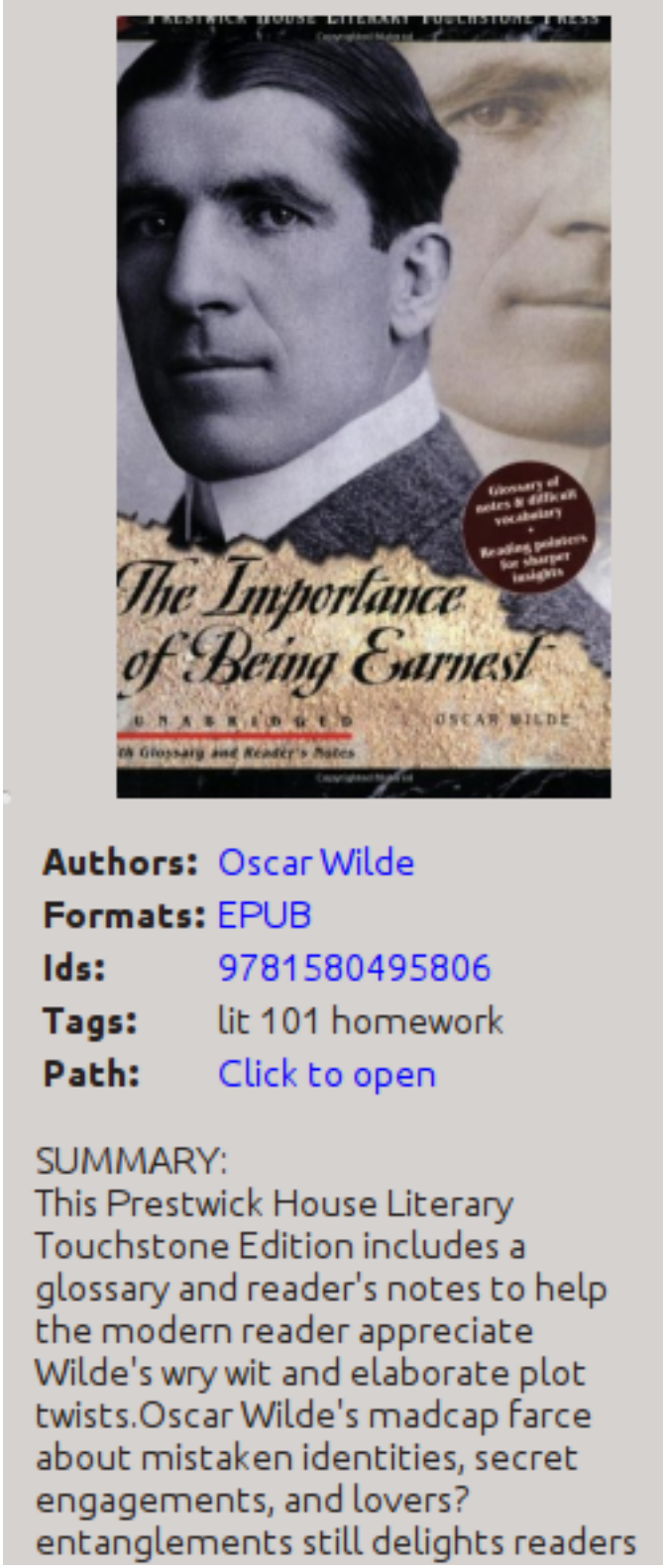
```
Foundation and Earth - Isaac Asimov.txt
```

interpretiert daraus den Titel: Foundation and Earth und den Autor: Isaac Asimov.

Tipp

Der zuvor genannte reguläre Ausdruck schlägt fehl, wenn der Dateiname keinen Bindestrich enthält.

1.11 Über dieses Buch



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.

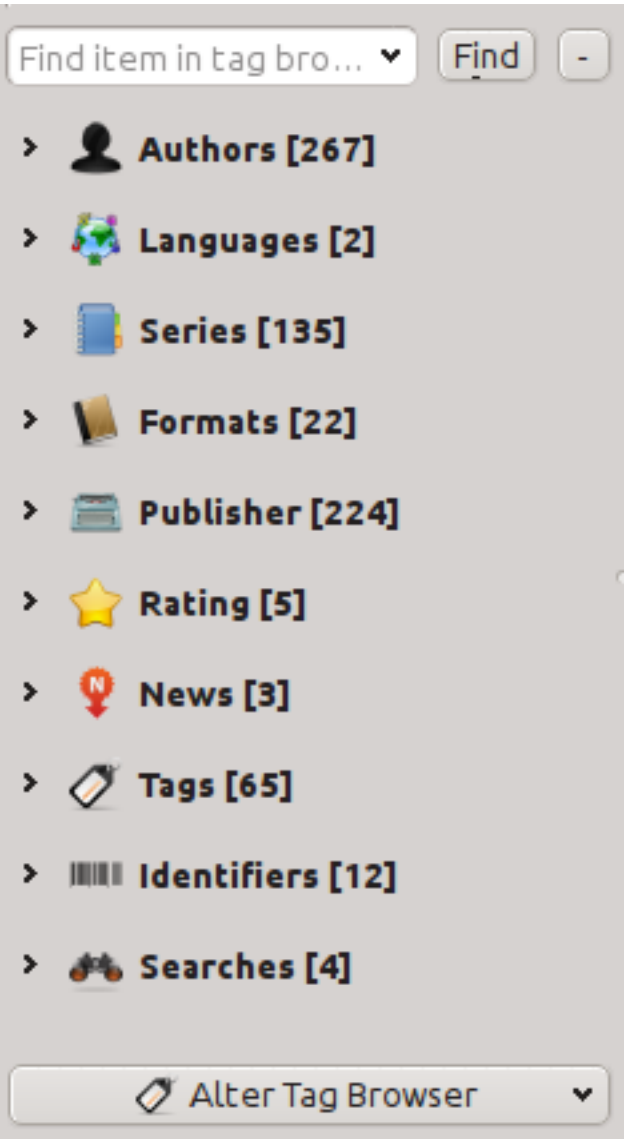
You can change the cover of the book by simply drag and dropping an image onto the Book details panel. If you wish to edit the cover image in an external program, simply right click on it and choose *Open with*.

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 Schlagwortbrowser



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. „Alle zu der Kategorie passende Elemente“ durch ein einziges grünes Pluszeichen gekennzeichnet.
2. „Alle zu der Kategorie oder den zugehörigen Unterkategorien passende Elemente“ durch zwei grüne Pluszeichen gekennzeichnet.
3. „Alle zu der Kategorie nicht passende Elemente „ durch ein einzelnes rotes Minuszeichen angezeigt.
4. „Alle zu der Kategorie oder den zugehörigen Unterkategorien nicht passende Elemente“ durch zwei rote Minus-Zeichen angezeigt.

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“* (die Titelseite 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).

You can use your own icons for categories and values in categories. To change the icon for a category, right-click on the category (the outer-level item) and choose *Change (category name) category icon*. A dialog will open where you can pick an image to be used as the icon. To restore the icon to its default choose *Restore (category name) default icon*.

To choose icons for values in categories, right-click on a value then choose *Manage icon for (value name)*. You will see a list of choices:

- *Choose an icon for this value but not its children*. A dialog will open where you choose an icon for the value. Children of that value will not inherit that icon.

- *Choose an icon for this value and its children*. A dialog will open where you choose an icon for the value. Any children that don't have their own specified icon will inherit this icon.
- *Use the existing icon for this value but not its children*. This option is offered if the value already has an icon that is inherited by the value's children. Selecting it will make the icon apply to the value but not its children.
- *Use the existing icon for this value and its children*. This option is offered if the value already has an icon that is not inherited by the value's children. Selecting it will make the icon apply to the value and its children.
- *Use the default icon for this value*. This option is offered if the item has an icon. It removes the icon from the value and any children inheriting the icon. The default icon is what is specified below.
- *Reset all value icons to the default icon*. This option removes all item value icons for the category. It does not remove a template if one exists. There is no undo.
- *Use/edit a template to choose the default value icon*. This option permits you to provide a calibre template that returns the name of an icon file to be used as a default icon. The template can use two variables:
  - `category`: the lookup name of the category, for example `authors`, `series`, `#mycolumn`.
  - `value`: the value of the item within the category.
  - `count`: the number of books with this value. If the value is part of a hierarchy then the count includes the children.
  - `avg_rating`: the average rating for books with this value. If the value is part of a hierarchy then the average includes the children.

  Book metadata such as title is not available. Template database functions such as *book_count* (die Titelseite 187) and *book_values* (die Titelseite 188) will work, but the performance might not be acceptable. The following template functions will work in the GUI but won't work in the content server: *connected_device_name* (die Titelseite 195), *connected_device_uuid* (die Titelseite 195), *current_virtual_library_name* (die Titelseite 196), *is_marked* (die Titelseite 197), and *virtual_libraries* (die Titelseite 198).

  In the GUI, Python templates have full access to the calibre database. In the content server, Python templates have access to new API (see API documentation for the database interface[4]) but not the old API (LibraryDatabase).

  For example, this template specifies that any value in the clicked-on category beginning with *History* will have an icon named `flower.png`:

```
program:
  if substr($value, 0, 7) == 'History' then 'flower.png' fi
```

  If a template returns the empty string (`''`) then the category icon will be used. If the template returns a file name that doesn't exist then no icon is displayed.
- *Use the category icon as the default*. This option specifies that the icon used for the category should be used for any value that doesn't otherwise have an icon. Selecting this option removes any template icon specification.

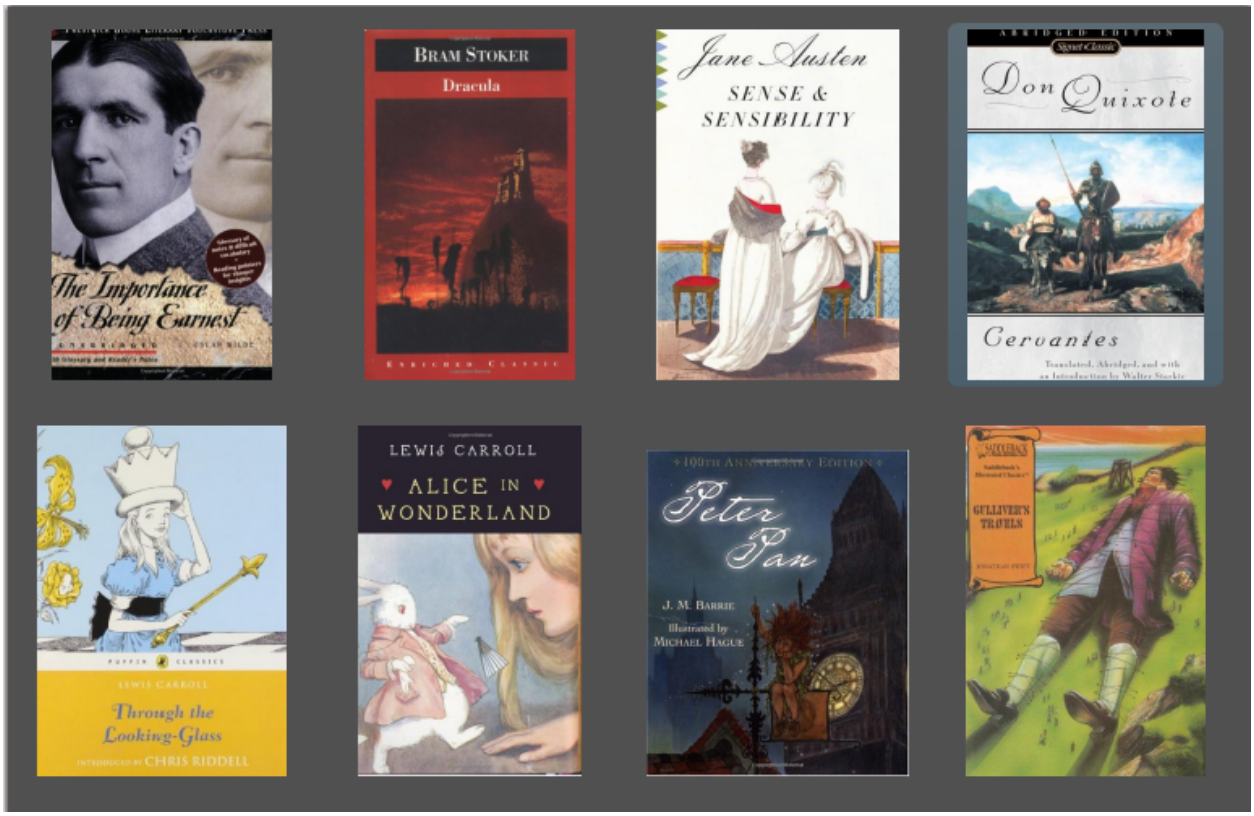
The icon is chosen using the following hierarchy:

1. The icon specified for the value, if it exists.
2. The icon specified for a parent node found by walking up the tree, if one exists.
3. The icon from a template, if a template exists and it returns a non-empty string.
4. The default category icon, which always exists.

Icons are per-user, not per-library, stored in the calibre configuration folder. Icons for item values are stored in the `tb_icons` subfolder. Icons used by templates are in the `template_icons` subfolder of `tb_icons`.

[4] https://manual.calibre-ebook.com/db_api.html

1.13 Titelbildraster



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 Titelbildbrowser

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.

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 Schnellansicht

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.

Einige Beispiele zur Schnellansicht: Schnell sehen, welche anderen Bücher:

- have some tag(s) applied to the currently selected book,
- haben die gleiche Serie, wie das aktuelle Buch,
- haben die gleichen Werte einer benutzerdefinierten Spalte wie das aktuelle Buch,
- wurden von einem der gleichen Autoren wie vom aktuellen Buch geschieben,
- 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.

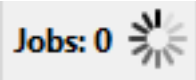
`Shift-` or `Ctrl-` double-click on a book in the Quickview window to open the edit metadata dialog on that book in the calibre window. The edited book will be Quickview'ed when you close the edit metadata dialog.

Ob eine Schnellansicht für eine Spalte besteht, wird im Tooltip der Spaltenanzeige angegeben, sobald man den Mauszeiger über die Spaltenüberschriften bewegt. Ebenfalls kann man dies durch einen Rechtsklick auf die Spaltenüberschrift und der auswählbaren Menüoption „Schnellansichtsspalten" erkennen. Wird diese ausgewählt, entspricht diese Option der Schnellansicht durch Drücken von „Q" in der aktuellen Zelle.

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 Aufträge



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 Tastaturkurzbefehle

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* (die Titelseite 57) which can be customised in the viewer *Preferences*.

Bemerkung

Hinweis: Die Calibre-Tastaturkurzbefehle benötigen keine Steuerungstaste (Strg, Alt, Shift etc.), solange darauf nicht explizit hingewiesen wird. Sie müssen nur die entsprechende Buchstabetaste drücken, z. B. „E“ für „Editieren“.

Tab. 1: Keyboard shortcuts for the main calibre program

Tastaturkurzbefehl	Aktion
F2 (Enter for macOS)	Metadaten des in der Buchliste markierten Feldes bearbeiten.
A	Bücher hinzufügen
Umschalt+	Add formats to the selected books
C	Convert selected books
D	An Gerät senden
Entf	Ausgewählte Bücher entfernen
E	Metadaten der ausgewählten Bücher bearbeiten
G	Bücher beziehen
I	Buchdetails anzeigen
K	Inhaltsverzeichnis bearbeiten
M	Ausgewählte Bucheinträge zusammenfügen
Alt+M	Ausgewählte Bucheinträge zusammenfügen, Originale behalten
O	Speicherort öffnen
P	Bücher perfektionieren
S	Bücher exportieren
T	Buch bearbeiten
V	Bücher öffnen

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Tastaturkurzbefehl	Aktion
Shift+V	View last read book
Alt+V/ Cmd+V for macOS	Spezifisches Format ansehen
Alt+Ums	Auftragsliste umschalten
Alt+Ums	Umschalten auf Titelbildbrowser
Alt+Ums	Toggle Book details panel
Alt+Ums	Umschalten auf Schlagwortbrowser
Alt+Ums	Umschalten auf Titelbildraster
Alt+A	Bücher anzeigen vom gleichen Autor wie das aktuelle Buch
Alt+T	Bücher anzeigen mit den gleichen Schlagwörtern wie das aktuelle Buch
Alt+P	Bücher anzeigen vom gleichen Verlag wie das aktuelle Buch
Alt+Ums	Bücher anzeigen der gleichen Serie wie das aktuelle Buch
/, Strg+F	Focus the Search bar
Umschalt+	Open the Advanced search dialog
Shift+A	Toggle the Search bar
Esc	Aktuelle Suche löschen
Umschalt+	Buchliste fokussieren
Strg+Es	Virtuelle Bibliothek leeren
Alt+Esc	Zusätzliche Einschränkungen leeren
Strg+*	Create a temporary Virtual library based on the current search
Strg+Re	Select the next Virtual library tab
Strg+Li	Select the previous Virtual library tab
N oder F3	Nächstes Buch finden, das der aktuellen Suche entspricht (funktioniert nur, falls das Kontrollkästchen Hervorhebung in den Sucheinstellungen aktiviert ist)
Umschalt+ oder Umschalt+	Vorheriges Buch finden, das der aktuellen Suche entspricht (funktioniert nur, falls das Kontrollkästchen Hervorhebung in den Sucheinstellungen aktiviert ist)
Strg+D	Metadaten und Titelseiten herunterladen
Strg+R	Calibre neu starten
Strg+Um	Calibre im Fehlerdiagnosemodus neu starten
Umschalt+	Leere Bücher zu Calibre hinzufügen
Strg+M	Markierungsstatus der ausgewählten Bücher umschalten
Ctrl+/ or Ctrl+Al	Open the popup to search the full text of all books in the library
Q	Schnellansichtfenster öffnen, um Bücher aus zugehörigen Serien, mit zugehörigen Schlagwörtern usw. anzuzeigen.
Shift+Q	Geöffneten Schnellansichtsbereich fokussieren
Shift+S	Suche im Schnellansichtsbereich durchführen

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Tastaturkurzbefehl	Aktion
F5	Aktuelle Sortierung wiederholen
Strg+Q	Calibre beenden
X	Toggle auto scroll of the book list
Ctrl+Al	Restrict the displayed books to only those books that are in a category currently displayed in the *Tag browser*
B	Browse annotations (highlights and bookmarks) made in the calibre E-book viewer for all books in the library
Ctrl+Sh	Browse notes associated with authors/series/tags/etc.
Alt+Shi	Toggle the layout between wide and narrow views

KAPITEL 2

Hinzufügen Ihrer bevorzugten Nachrichtenwebsite

calibre has a powerful, flexible and easy-to-use framework for downloading news from the Internet and converting it into an e-book. The following will show you, by means of examples, how to get news from various websites.

Folgen Sie den Beispielen in der unten stehenden Liste, um ein Verständnis von der Verwendung des Frameworks zu erhalten:

- *Vollautomatisches Abrufen* (die Titelseite 33)
  - *Der Calibre-Blog* (die Titelseite 34)
  - *bbc.co.uk* (die Titelseite 35)
- *Anpassen des Abrufprozesses* (die Titelseite 35)
  - *Verwenden der Druckversion von bbc.co.uk* (die Titelseite 35)
  - *Ersetzen von Artikel-styles* (die Titelseite 36)
  - *Manipulieren von heruntergeladenen Inhalten* (die Titelseite 37)
  - *Beispiel aus der Praxis* (die Titelseite 37)
- *Tipps zum Entwickeln neuer Schemata* (die Titelseite 40)
- *Weiterführende Literatur* (die Titelseite 41)
- *API-Dokumentation* (die Titelseite 41)

2.1 Vollautomatisches Abrufen

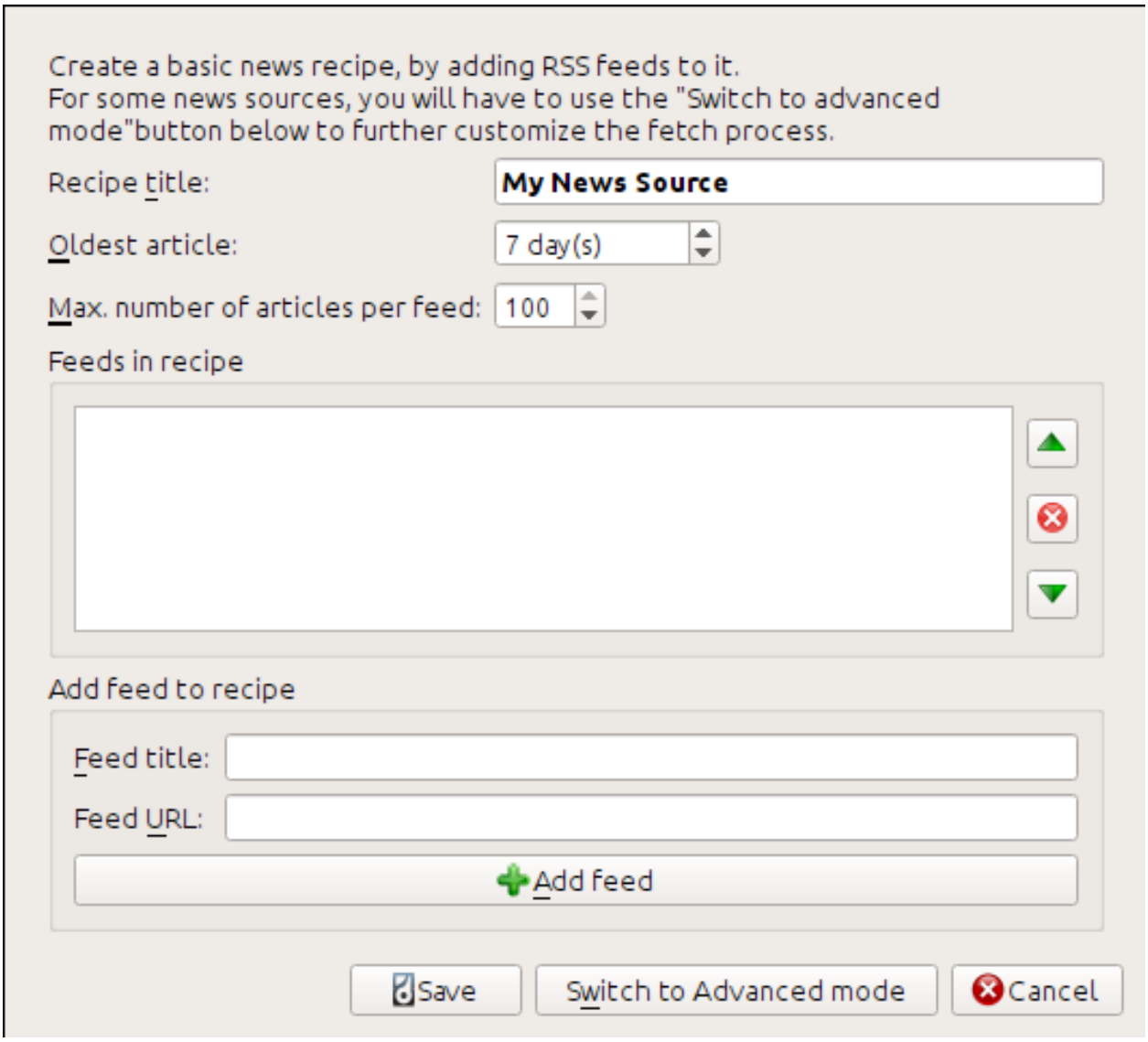
Wenn Ihre Nachrichtenquelle einfach genug aufgebaut ist, könnte Calibre in der Lage sein, sie vollautomatisch und nur unter Angabe ihrer URL abzurufen. Calibre sammelt alle Informationen die für das Herunterladen einer Nachrichtenquelle nötig sind in einem *recipe*. Um Calibre eine neue Nachrichtenquelle beizubringen, müssen Sie für diese also ein *recipe* erstellen. Hier nun einige Beispiele:

2.1.1 Der Calibre-Blog

The calibre blog is a blog of posts that describe many useful calibre features in a simple and accessible way for new calibre users. In order to download this blog into an e-book, we rely on the *RSS* feed of the blog:

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

I got the RSS URL by looking under „Subscribe to“ at the bottom of the blog page and choosing *Posts → Atom*. To make calibre download the feeds and convert them into an e-book, you should right click the *Fetch news* button and then the *Add a custom news source* menu item and then the *New Recipe* button. A dialog similar to that shown below should open up.



First enter `Calibre Blog` into the *Recipe title* field. This will be the title of the e-book that will be created from the articles in the above feeds.

The next two fields (*Oldest article* and *Max. number of articles*) allow you some control over how many articles should be downloaded from each feed, and they are pretty self explanatory.

To add the feeds to the recipe, enter the feed title and the feed URL and click the *Add feed* button. Once you have added the feed, simply click the *Save* button and you're done! Close the dialog.

To test your new *recipe*, click the *Fetch news* button and in the *Custom news sources* sub-menu click *calibre Blog*. After a couple of minutes, the newly downloaded e-book of blog posts will appear in the main library view (if you have your reader connected, it will be put onto the reader instead of into the library). Select it and hit the *View* button to read!

The reason this worked so well, with so little effort is that the blog provides *full-content RSS* feeds, i.e., the article content is embedded in the feed itself. For most news sources that provide news in this fashion, with *full-content* feeds, you don't need any more effort to convert them to e-books. Now we will look at a news source that does not provide full content feeds. In such feeds, the full article is a webpage and the feed only contains a link to the webpage with a short summary of the article.

2.1.2 bbc.co.uk

Let's try the following two feeds from *The BBC*:

1. News Front Page: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_page/rss.xml
2. Science/Nature: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/science/nature/rss.xml

Follow the procedure outlined in *Der Calibre-Blog* (die Titelseite 34) above to create a recipe for *The BBC* (using the feeds above). Looking at the downloaded e-book, we see that calibre has done a creditable job of extracting only the content you care about from each article's webpage. However, the extraction process is not perfect. Sometimes it leaves in undesirable content like menus and navigation aids or it removes content that should have been left alone, like article headings. In order, to have perfect content extraction, we will need to customize the fetch process, as described in the next section.

2.2 Anpassen des Abrufprozesses

When you want to perfect the download process, or download content from a particularly complex website, you can avail yourself of all the power and flexibility of the *recipe* framework. In order to do that, in the *Add custom news sources* dialog, simply click the *Switch to Advanced mode* button.

The easiest and often most productive customization is to use the print version of the online articles. The print version typically has much less cruft and translates much more smoothly to an e-book. Let's try to use the print version of the articles from *The BBC*.

2.2.1 Verwenden der Druckversion von bbc.co.uk

The first step is to look at the e-book we downloaded previously from *bbc.co.uk* (die Titelseite 35). At the end of each article, in the e-book is a little blurb telling you where the article was downloaded from. Copy and paste that URL into a browser. Now on the article webpage look for a link that points to the „Printable version". Click it to see the print version of the article. It looks much neater! Now compare the two URLs. For me they were:

Artikel-URL
: https://news.bbc.co.uk/2/hi/science/nature/7312016.stm

Druckversion-URL
: https://newsvote.bbc.co.uk/mpapps/pagetools/print/news.bbc.co.uk/2/hi/science/nature/7312016.stm

Es sieht also so aus als müssten wir, um die Druckversion zu erhalten, jeder Artikel-URL folgendes Präfix geben:

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/newsonlir
```

You can see that the fields from the *Basic mode* have been translated to Python code in a straightforward manner. We need to add instructions to this recipe to use the print version of the articles. All that's needed is to add the following two lines:

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

This is Python, so indentation is important. After you've added the lines, it should look like:

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/newsonlir

    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[5] see the tutorial[6].

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.

2.2.2 Ersetzen von Artikel-styles

In the previous section, we saw that the font size for articles from the print version of *The BBC* was too small. In most websites, *The BBC* included, this font size is set by means of *CSS* stylesheets. We can disable the fetching of such stylesheets by adding the line:

[5] https://www.python.org
[6] https://docs.python.org/tutorial/

```
no_stylesheets = True
```

Das Schema sieht nun so aus:

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/
```

Die neue Version sieht ziemlich gut aus. Falls Sie ein Perfektionist sind, dann sollten Sie den nächsten Abschnitt lesen, der das Ändern der heruntergeladenen Inhalte erläutert.

2.2.3 Manipulieren von heruntergeladenen Inhalten

calibre contains very powerful and flexible abilities when it comes to manipulating downloaded content. To show off a couple of these, let's look at our old friend the *The BBC* (die Titelseite 37) recipe again. Looking at the source code (*HTML*) of a couple of articles (print version), we see that they have a footer that contains no useful information, contained in

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

Das kann entfernt werden durch die Ergänzung von:

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

to the recipe. Finally, lets replace some of the *CSS* that we disabled earlier, with our own *CSS* that is suitable for conversion to an e-book:

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

With these additions, our recipe has become „production quality".

This *recipe* explores only the tip of the iceberg when it comes to the power of calibre. To explore more of the abilities of calibre we'll examine a more complex real life example in the next section.

2.2.4 Beispiel aus der Praxis

A reasonably complex real life example that exposes more of the *API* of `BasicNewsRecipe` is the *recipe* for *The New York Times*

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

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
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)
                 if not a:
                     continue
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                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'))
```

Wir sehen mehrere neue Funktionen in *recipe*. Erstens haben wir:

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

This sets the displayed time on the front page of the created e-book to be in the format, `Day, Day_Number Month, Year`. See *timefmt* (die Titelseite 51).

Then we see a group of directives to cleanup the downloaded *HTML*:

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

Diese ersetzen alles vor der ersten `<h1>` Markierung und alles nach der ersten Markierung deren id `Fußzeile` ist. Siehe *remove_tags* (die Titelseite 50), *remove_tags_before* (die Titelseite 50), *remove_tags_after* (die Titelseite 50).

Die nächste interessante Funktion ist:

```
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()* (die Titelseite 43) 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[7] to understand the code in `get_browser`.

The next new feature is the *calibre.web.feeds.news.BasicNewsRecipe.parse_index()* (die Titelseite 44) method. Its job is to go to https://www.nytimes.com/pages/todayspaper/index.html and fetch the list of articles that appear in *todays* 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[8] to parse the daily paper webpage. You can also use other, more modern parsers if you dislike BeautifulSoup. calibre comes with lxml[9] and html5lib[10], 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)
```

The final new feature is the *calibre.web.feeds.news.BasicNewsRecipe.preprocess_html()* (die Titelseite 45) method. It can be used to perform arbitrary transformations on every downloaded HTML page. Here it is used to bypass the ads that the nytimes shows you before each article.

2.3 Tipps zum Entwickeln neuer Schemata

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* (die Titelseite 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[11] to share it with other calibre users.

[7] https://mechanize.readthedocs.io/en/latest/
[8] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[9] https://lxml.de/
[10] https://github.com/html5lib/html5lib-python
[11] https://www.mobileread.com/forums/forumdisplay.php?f=228

Bemerkung

Für macOS sind die Komandozeilentools in Calibre integriert. Sollte Calibre z.B. in `/Applications` installiert sein, dann finden Sie die Tools unter `/Applications/calibre.app/Contents/MacOS/`.

Siehe auch

ebook-convert **(die Titelseite 326)**
The command line interface for all e-book conversion.

2.4 Weiterführende Literatur

To learn more about writing advanced recipes using some of the facilities, available in `BasicNewsRecipe` you should consult the following sources:

API documentation **(die Titelseite 41)**
Dokumentation der `BasicNewsRecipe`-Klasse und aller ihrer wichtigen Methoden und Felder.

BasicNewsRecipe[12]
Der Quelltext des `BasicNewsRecipe`

Integrierte Schemata[13]
Der Quellcode für die in Calibre eingebauten Schemata

The calibre recipes forum[14]
Lots of knowledgeable calibre recipe writers hang out here.

2.5 API-Dokumentation

2.5.1 API Dokumentation für Rezepte

Das API für das Schreiben von Rezepten wird definiert durch *BasicNewsRecipe* (die Titelseite 41)

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

Base class that contains logic needed in all recipes. By overriding progressively more of the functionality in this class, you can make progressively more customized/powerful recipes. For a tutorial introduction to creating recipes, see *Hinzufügen Ihrer bevorzugten Nachrichtenwebsite* (die Titelseite 33).

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()* (die Titelseite 45).

classmethod image_url_processor(*baseurl*, *url*)

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

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:

[12] https://github.com/kovidgoyal/calibre/blob/master/src/calibre/web/feeds/news.py
[13] https://github.com/kovidgoyal/calibre/tree/master/recipes
[14] https://www.mobileread.com/forums/forumdisplay.php?f=228

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

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

Convenience method to take a BeautifulSoup[15] `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: BeautifulSoup[16] `Tag`

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*)

Causes the recipe download system to abort the download of this recipe, displaying a simple feedback message to the user.

add_toc_thumbnail(*article*, *src*)

Call this from populate_article_metadata with the src attribute of an <img> 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.

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.

Parameter

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*)

Create a generic cover for recipes that don't have a cover

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[17]. 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[18].

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

Return a browser instance used to fetch documents from the web. By default it returns a mechanize[19] browser instance that supports cookies, ignores robots.txt, handles refreshes and has a random common user agent.

To customize the browser override this method in your sub-class as:

```
def get_browser(self, *a, **kw):
    br = super().get_browser(*a, **kw)
    # Add some headers
    br.addheaders += [
        ('My-Header', 'one'),
        ('My-Header2', 'two'),
    ]
    # Set some cookies
    br.set_cookie('name', 'value')
    br.set_cookie('name2', 'value2', domain='.mydomain.com')
    # Make a POST request with some data
    br.open('https://someurl.com', {'username': 'def', 'password': 'pwd'}).
↪read()
    # Do a login via a simple web form (only supported with mechanize␣
↪browsers)
    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
```

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.

Rückgabe
A floating point number, the delay in seconds.

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[20] 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.

Parameter

- **url** – Die zu folgende URL
- **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()* (die Titelseite 43). 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()* (die Titelseite 44).

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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                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()* (die Titelseite 42).

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).

Parameter

- **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.

Parameter

- **oeb** – Ein OEBBook-Objekt
- **opts** – Konvertierungsoptionen

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.

Parameter

- **soup** – A BeautifulSoup[21] instance containing the downloaded *HTML*.
- **first_fetch** – Wahr, wenn dies die erste Seite eines Artikels ist.

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 BeautifulSoup[22] 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.

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 BeautifulSoup[23] 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: Eine Liste mit Titeln.

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.

articles_are_obfuscated = False

Set to True and implement *get_obfuscated_article()* (die Titelseite 44) 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"
```

browser_type = 'mechanize'

The simulated browser engine to use when downloading from servers. The default is to use the Python mechanize browser engine, which supports logging in. However, if you don't need logging in, consider changing this to either ‚webengine' which uses an actual Chromium browser to do the network requests or ‚qt' which uses the Qt Networking backend. Both ‚webengine' and ‚qt' support HTTP/2, which mechanize does not and are thus harder to fingerprint for bot protection services.

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, 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 x 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)/compress_news_images_auto_size bytes if possible by reducing the quality level, where w x 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)/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[24]. 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()* (die Titelseite 44) 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()* (die Titelseite 43) 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']
```

wird alle URLs, die *ads.doubleclick.net* beinhalten, löschen.

Only one of *BasicNewsRecipe.match_regexps* (die Titelseite 48) or *BasicNewsRecipe.filter_regexps* (die Titelseite 48) should be defined.

handle_gzip = True

Set to False if you do not want to use gzipped transfers with the mechanize browser. 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'}
```

Um statt dessen URLs zu nutzen, setzen Sie es auf:

```
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* (die Titelseite 50). 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* (die Titelseite 48) or *BasicNewsRecipe.filter_regexps* (die Titelseite 48) 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* (die Titelseite 49)

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 *regexp* 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

recipe_specific_options = None

Specify options specific to this recipe. These will be available for the user to customize in the Advanced tab of the Fetch News dialog or at the ebook-convert command line. The options are specified as a dictionary mapping option name to metadata about the option. For example:

```
recipe_specific_options = {
    'edition_date': {
        'short': 'The issue date to download',
        'long':  'Specify a date in the format YYYY-mm-dd to download the␣
↪issue corresponding to that date',
        'default': 'current',
    }
}
```

When the recipe is run, self.recipe_specific_options will be a dict mapping option name to the option value specified by the user. When the option is unspecified by the user, it will have the value specified by ‚default‘. If no default is specified, the option will not be in the dict at all, when unspecified by the user.

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()* (die Titelseite 43). 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[25] 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* (die Titelseite 50). 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* (die Titelseite 50). 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()* (die Titelseite 42) to work with the URL scheme of your particular website.

reverse_article_order = False

Reverse the order of articles in each feed

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* (die Titelseite 47) > 0

summary_length = 500

Maximale Anzahl an Zeichen in der kurzen Beschreibung

```
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

Zeitüberschreitung in Sekunden beim Abruf von Dateien vom Server

title = 'Nachrichtenquelle unbekannt'

Der Titel, der für das eBook verwendet werden soll

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.

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

KAPITEL 3

eBook-Betrachter

Calibre hat einen integrierten eBook-Betrachter, mit dem Sie alle gängigen eBook-Formate lesen können. Der Betrachter ist sehr anpassbar und hat viele zusätzliche Funktionen

- *Starten des eBook-Betrachters* (die Titelseite 54)
- *Navigation durch ein eBook* (die Titelseite 54)
- *Hervorgehobener Text* (die Titelseite 55)
- *Vorlesen* (die Titelseite 55)
- *Searching the text* (die Titelseite 55)
- *Von Verweis zu Verweis unter Nutzung der Tastatur* (die Titelseite 56)
- *Anpassen des Erscheinungsbilds für Ihr Lesevergnügen* (die Titelseite 56)
- *Wörterbuchsuche* (die Titelseite 56)
- *Kopieren von Text und Bildern* (die Titelseite 57)
- *Vergrößerung von Bildern* (die Titelseite 57)
- *Syncing with a paper edition of the current book* (die Titelseite 57)
- *Tastaturkurzbefehle* (die Titelseite 57)
- *Der Inhalt kann nicht neu formatiert werden* (die Titelseite 59)
- *Designing your book to work well with the calibre E-book viewer* (die Titelseite 60)

3.1 Starten des eBook-Betrachters

Ein oder mehrere Bücher werden in Calibre mit dem E-Book Viewer (Betrachter) zur Ansicht geöffnet indem das oder die Bücher angewählt werden und der *View* Button geklickt wird. Das geht eben so mit einem Doppelklick auf das Buch oder einem klick mit der rechten Maustaste und der Auswahl des Viewers. Daraufhin wird das Buch in je einem E-Book Viewer Fenster geöffnet. Unter Windows gibt es für den Viewer einen eigenen Startmenüeintrag. Unter macOS, können Sie das Viewer Icon an das Dock anpinnen und dann von dort starten. Unter Linux gibt es einen Starter im Desktop Menü oder auch das Kommando **`ebook-viewer`**.

3.2 Navigation durch ein eBook

Umblättern in einem Buch können Sie auf mehrere Arten

- Mit der Maus den linken oder rechten Rand anklicken
- Tastendruck auf `spacebar`, `page up`, `page down` oder die Pfeiltasten
- Bei einem Touchscreen auf den Text tippen oder nach links oder rechts wischen

Ein Kontextmenü mit Einstellungen des Betrachters öffnen Sie auf foldende Arten

- Rechtsklick auf den Text
- Tastendruck auf die `Esc` oder `Menu` Taste
- Bei einem Touchscreen tippen Sie in das obere Drittel des Bildschrims

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 Lesezeichen

Lesen Sie gerade in einem Buch und schließen das Betrachter Fenster, dann wird die Leseposition vermerkt. Öffnen Sie das Buch dann später wieder, dann wird Ihnen genau diese Stelle wieder angezeigt. Sie können aber auch Lesezeichen selbst setzen. Nutzen Sie den *Bookmarks* Button im Kontextmenü oder die Taste `Ctrl+B`. Ist Ihr Buch im EPUB -Format, dann werden diese Lesezeichen direkt in die EPUB Datei geschrieben. Teilen Sie das E-Book oder kopieren es dann bleiben die Lesezeichen darin enthalten. Dieses Verhalten können Sie unter dem Punkt *Miscellaneous* im Kontextmenü abschalten.

3.2.2 Inhaltsverzeichnis

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 Navigieren nach Position

Im Gegensatz zu Büchern aus Papier haben E-Books keine wirkliche Seiteneinteilung. Um dennoch eine Stelle genau angeben zu können nutzen Sie im Kontextmenü den Eintrag *Go to → Location* .

Sie können diese Positionsabgabe nutzen um Teile des Buches eindeutig zu indentifzieren. Teilen Sie das Buch dann mit anderen, können Sie leicht jede Stelle referenzieren. Finden kann man das alles im Kontextmenü unter *Go to → Location* .

Da ist ein Link (URL) den Sie in die Zwischenablage übernehmen können und den Sie dann in andere Programme oder Dokumente einfügen. Klicken Sie dort dann auf diesen Link, öffnet sich der Calibre E-Book Viewer (Betrachter) mit dem geöffnetem Buch an der richtigen Stelle.

Sollten Sie auf Verweise in einem E-Book klicken, springen Sie wie bei Fuß- oder Endnoten zu verschiedenen Stellen im Buch. Sie können auch die *Back* und *Forward* Buttons links oben im Betrachter Menü nutzen, die genau das tun was Sie erwarten, wie in einem Webbrowser.

3.2.4 Verweismodus

Calibre hat einen nützlichen *Reference mode*. Klicken Sie im kontextmenü auf *Reference mode* unm Ihn zu aktivieren. Haben Sie das getan, dann erscheint vor jedem Absatz eine Nummer auf die man verweisen kann.

Diese Nummer kann verwendet werden um genau zu speziellen Stellen im Buch zu gelangen. Oder auf die man in anderen Werken verweisen kann. Gezielt eine solche Nummer anspringen können Sie indem Sie im Kontextmenü *Go to function* anwählen und die entsprechende Nummer dort eingeben.

3.3 Hervorgehobener 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.

Wählen Sie im Kontextmeü *Highlights* um in einem weiteren Fenster eine Liste aller Hervorhebungen des Buches nach Kapitel sortiert zu sehen.

Um schnell auf alle Hervorhebungen zuzugreifen können Sie in der Bibliothek den Button *View* mit Rechtsklick anwählen und dort dann *Browse annotations* auswählen. Um die Notizen zu den Hervorhebungen anzuzeigen.

Und zum Schluß: Wenn Sie Inhalte des Calibre Content server im Betrachter nutzen dann können Sie alle Anmerkungen, Hervorhebungen und Notizen syncronisieren indem Sie das im Kontextmenü unter *Preferences → Miscellaneous* einstellen und einen Benutzernamen dort angeben. Durch Eingabe von `*` wird mit allen Benutzern syncronisiert.

3.4 Vorlesen

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 or sentence being currently read is highlighted. Speech is synthesized from the text using either the Piper[26] neural text-to-speech engine or your operating system services for text-to-speech. You can change the backend and the voice being used by clicking the gear icon in the bar that is displayed while *Read aloud* is active.

Sie können der Leiste, die beim markieren von Text erscheint auch einen *Read aloud* Button hinzufügen, damit markierter Text schnell vorgelesen werden kann. Die Einstellung finden Sie im Kontextmenü des Betrachters unter *Selection behavior*.

Bemerkung

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.

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:

[26] https://github.com/rhasspy/piper

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* (die Titelseite 214).

3.6 Von Verweis zu Verweis unter Nutzung der Tastatur

Der E-Book Betrachter hat einen *Hints mode* der es erlaubt von Verweise im Text anzuspringen ohne die Maus nutzen zu müssen. Drücken Sie die Tastenkombination `Alt+F` . Daraufhin werden alle Verweise im Text markiert und mit einer Nummer oder einem Buchstaben darüber angezeigt. Drücken Sie die entsprechende Taste um zum Verweis zu gelangen. Mit der `Esc` Taste brechen Sie diesen *Hints mode* Modus ab.

Sind allerdings mehr als fünf Verweise auf dem Bildschirm sichtbar, dann haben einige davon mehrere Buchstaben oder Zahlen darüber. Drücken Sie dann einfach beide Tasten nacheinander. Oder auch die erste Ziffer und dann Enter um die zweite einzugeben. Die Taste `Backspace` ist aktiv, falls Sie sich vertippt haben.

3.7 Anpassen des Erscheinungsbilds für Ihr Lesevergnügen

Die Textgröße können Sie schnell ändern indem Sie im Kontextmenü *Font size* wählen oder mit der Tastatur `Ctrl++` oder `Ctrl+-` nutzen oder halten Sie die `Ctrl` Taste und nutzen Sie das Mausrad.

Für das Anpassen der Farben nutzen Sie *Colors* sim Kontextmenü des Betrachters.

Ebenso kann die Anzahl der (Buch-) Seiten pro Bildschirm und auch die Breite und Höhe im Kontextmenü unter *Page layout* gewählt werden.

Auch Überschriften können hinzugefügt werden. Dort kann zum Beispiel „überige Zeit zum Lesen“, „Aktuelles Kapitel“, „Leseposition“ und Anderes angezeigt werden. Das stellen Sie unter *Headers and footers* im Kontexmenü ein.

Noch weitere Anpassungen können Sie mit *Styles* vornehmen. An der Stelle kann auch ein Hintergrundbild für den Text oder ein Stylesheet gesetz werden, das dann für jedes Buch genutzt wird. Hier können dann Sachen wie Absatzformate, Text Ausrichtungen dun anderes eingestellt werden. Beispiele sehen Sie im Calibre Forum unter: <https://www.mobileread.com/forums/showthread.php?t=51500>`_.

3.8 Wörterbuchsuche

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 Kopieren von Text und Bildern

Bilder und Text können einfach mit der Maus markiert werden und dann mit *Copy* in die Zwischenablage übernommen werden. Von dort kann der Inhalt einfach in andere Programme oder Dokumente übernommen werden.

3.10 Vergrößerung von Bildern

Um ein Bild größer oder in Bildschirmgröße anzuzeigen klicken Sie doppelt darauf oder bei Touchscreen tippen Sie lange darauf. Ein Rechtsklick auf ein Bild gibt Ihnen *View image* zur Auswahl.

3.11 Syncing with a paper edition of the current book

Some e-books, that have corresponding print editions, include metadata that marks the start of each paper page. For such e-books, the viewer allows you to jump to a particular paper edition page via the *Go to* button in the viewer controls. You can also optionally display the paper page corresponding to the current location in the book's headers or footers via the viewer settings, by adding *Pages from paper edition* to either the header or the footer.

3.12 Tastaturkurzbefehle

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:

Tab. 1: Keyboard shortcuts for the calibre E-book viewer

Tastaturkurzbefehl	Aktion
`Home`, `Ctrl+Ar` `Ctrl+Ar`	Scroll to the start of the current file in a multi file book
`Ctrl+Hc`	Zum Buchanfang scrollen
`Ctrl+En`	Zum Buchende scrollen
`End`, `Ctrl+Ar` `Ctrl+Ar`	Scroll to the end of the current file in a multi file book
`ArrowUp`	Rückwärts scrollen, weich im fortlaufenden Modus und bildschirmweise im Seitenmodus
`ArrowDown`	Vorwärts scrollen, weich im fortlaufenden Modus und bildschirmweise im Seitenmodus
`ArrowLeft`	Ein wenig nach links scrollen im fortlaufenden Modus oder um eine Seite im Seitenmodus
`ArrowRight`	Ein wenig nach rechts scrollen im fortlaufenden Modus oder um eine Seite im Seitenmodus

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Tastaturkurzbefehl	Aktion
PageUp, Shift+S	Bildschirmweise rückwärts scrollen
PageDow Space-bar	Bildschirmweise vorwärts scrollen
Ctrl+Pa	Zum vorherigen Abschnitt scrollen
Ctrl+Pa	Zum nächsten Abschnitt scrollen
Alt+Arr	Zurück
Alt+Arr	Weiter
Ctrl+T	Toggle Table of Contents
Ctrl+S	Vorlesen
Alt+P	Change settings quickly by creating and switching to *profiles*
Alt+f	Links mit Tastenkürzeln folgen
Ctrl+C	In Zwischenablage kopieren
Alt+C	Aktuellen Abschnitt in die Zwischenablage kopieren
Ctrl+Sh	Aktuellen Abschnitt als calibre:// Link (URL) in die Zwischenablage kopieren
/, Ctrl+f, Cmd+f	Suche starten
F3, Enter	Nächste suchen
Shift+F Shift+E	Vorherige suchen
Ctrl+Pl Me-ta+Plus	Schrift vergrößern
Ctrl+Mi Me-ta+Minu	Schrift verkleinern
Ctrl+0	Standardschriftgröße wiederherstellen
Ctrl+]	Mehr Seiten pro Bildschirm
Ctrl+[	Weniger Seiten pro Bildschirm
Ctrl+Al	Automatische Anzahl von Seiten pro Bildschirm
F11, Ctrl+Sh	Vollbildmodus an/aus
Strg+M	Für das Textlayout zwischen Seitenmodus und fortlaufendem Modus umschalten
Ctrl+W	Ein- / Ausschalten der Scollbar
Ctrl+X	Verweismodus an-/ausschalten
Ctrl+B	Lesezeichen ein-/ausblenden
Ctrl+Al	Neues Lesezeichen
Ctrl+N, Ctrl+E	Zeige die Metadaten des Buches

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Tastaturkurzbefehl	Aktion
Ctrl+Al	Buch neu laden
Ctrl+Al	
Ctrl+Sh	Nach der aktuellen Auswahl ein Word vorwärts springen
Ctrl+Sh	Nach der aktuellen Auswahl ein Word rückwärts gehen
Shift+A	Ändern Sie die aktuelle Auswahl um ein Zeichen vorwärts
Shift+A	Ändern der aktuellen Auswahl rückwärts um ein Zeichen
Shift+A	Ändern Sie die aktuelle Auswahl um eine Zeile vorwärts
Shift+H	Erweitern der aktuellen Auswahl bis zum Anfang der Zeile
Shift+E	Erweitern Sie die aktuelle Auswahl bis zum Ende der Zeile
Ctrl+A	Select all
Shift+A	Die aktuelle Auswahl um eine Zeile nach hinten verändern
Ctrl+Sh	Die aktuelle Auswahl um einen Absatz weiterschalten
Ctrl+Sh	Ändern der aktuellen Auswahl rückwärts um einen Absatz
Esc, Menu-Key	Zeige die E-Book-Anzeige Steuerung
Ctrl+Cc	Zeige die E-Book-Anzeige Einstellungen
Ctrl+Es Me-ta+Esc, Me-ta+Comm	
Ctrl+G, ;, :	Im Buch zu einem angegebenen Ort oder einer angegebenen Position springen
Ctrl+Sp	Ein- / Aus automatisches scrollen
Alt+Arr	Automatisches scrollen schneller
Alt+Arr	Automatisches scrollen langsamer
Ctrl+I	Inspektor ein-/ausblenden
Ctrl+L	Nachschlagebereich ein-/ausblenden
Ctrl+Q (Cmd+Q on ma-cOS)	Beenden
Ctrl+P	Buch als PDF drucken
Ctrl+F1	Ein / Aus Werkzeuge (Toolbar)
Ctrl+H	EIn / Aus der Hervorhebungen Leiste
Strg+D	Dieses Buch bearbeiten

3.13 Der Inhalt kann nicht neu formatiert werden

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 }
```

3.14 Designing your book to work well with the calibre E-book viewer

The calibre E-book 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[27]. The calibre E-book 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.

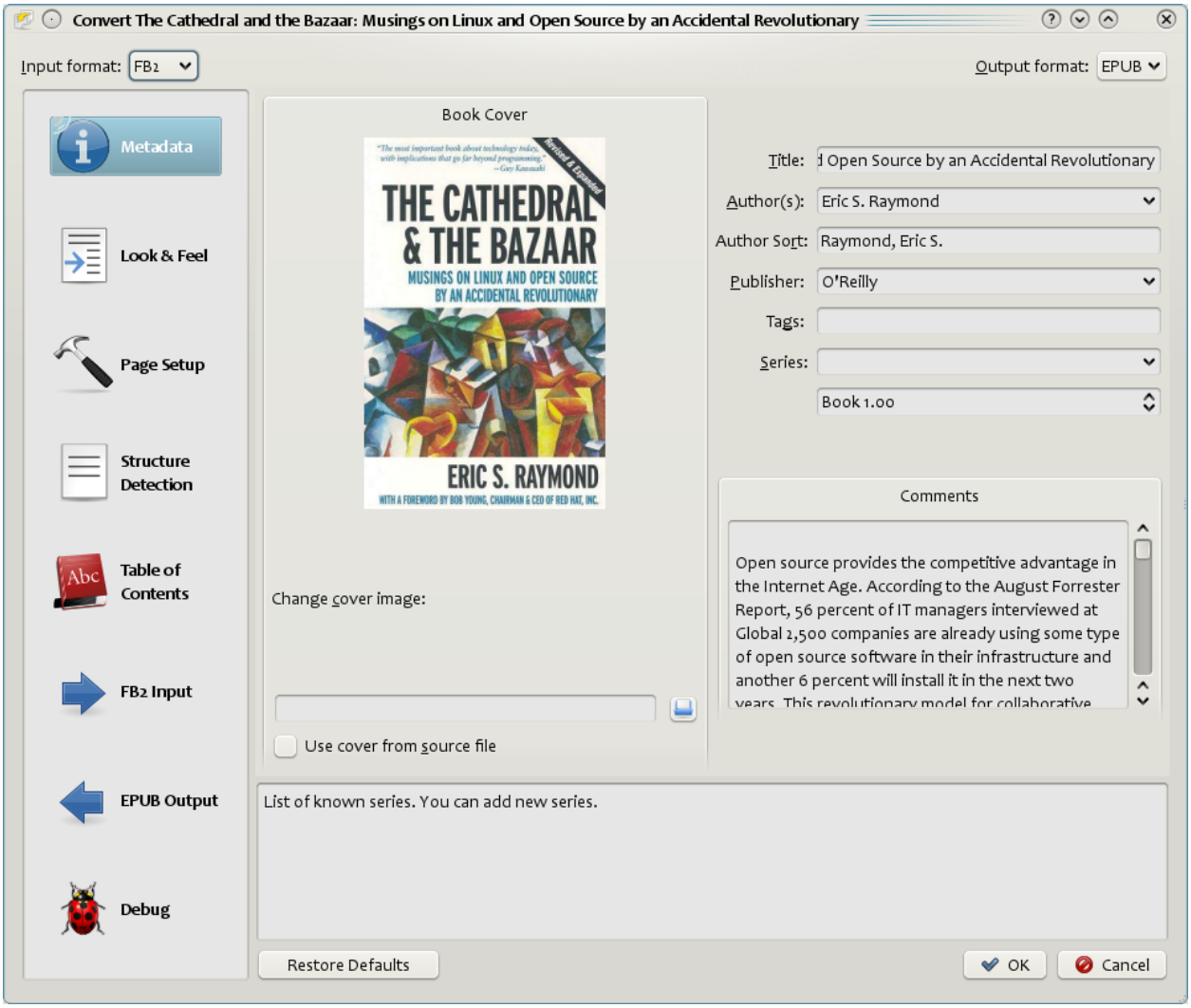
[27] https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties

KAPITEL 4

eBook Konvertierung

Calibre verfügt über ein Konvertierungssystem, das so konzipiert ist, dass es sehr einfach zu bedienen ist. Normalerweise fügen Sie einfach ein Buch zu Calibre hinzu, klicken auf „Konvertieren“ und calibre wird sich bemühen, eine Ausgabe zu erzeugen, die so nah wie möglich an der Eingabe liegt. Calibre akzeptiert jedoch eine sehr große Anzahl von Eingabeformaten, von denen nicht alle so gut wie andere für die Konvertierung in E-Books geeignet sind. Im Falle solcher Eingabeformate oder wenn Sie einfach eine größere Kontrolle über das Konvertierungssystem wünschen, bietet Calibre eine Vielzahl von Optionen zur Feinabstimmung des Konvertierungsprozesses. Beachten Sie jedoch, dass das Konvertierungssystem von Calibre kein Ersatz für einen ausgewachsenen E-Book-Editor ist. Um E-Books zu bearbeiten, empfehle ich, sie zunächst mit Calibre in EPUB oder AZW3 zu konvertieren und dann die *Edit book*-Funktion zu verwenden, um sie in eine perfekte Form zu bringen. Sie können dann das bearbeitete E-Book in Calibre als Input für die Konvertierung in andere Formate verwenden.

Dieses Dokument bezieht sich hauptsächlich auf die Konvertierungseinstellungen, wie sie im Konvertieren-Dialog unten dargestellt werden. Alle diese Einstellungen sind auch über die Befehlszeilenschnittstelle zur Konvertierung verfügbar, wie in *ebook-convert* (die Titelseite 326) dokumentiert. In Calibre können Sie für jede einzelne Einstellung Hilfe erhalten, indem Sie Ihren Mauszeiger über diese bewegen. Es erscheint dann eine Kurzinfo, welche die Einstellung beschreibt.



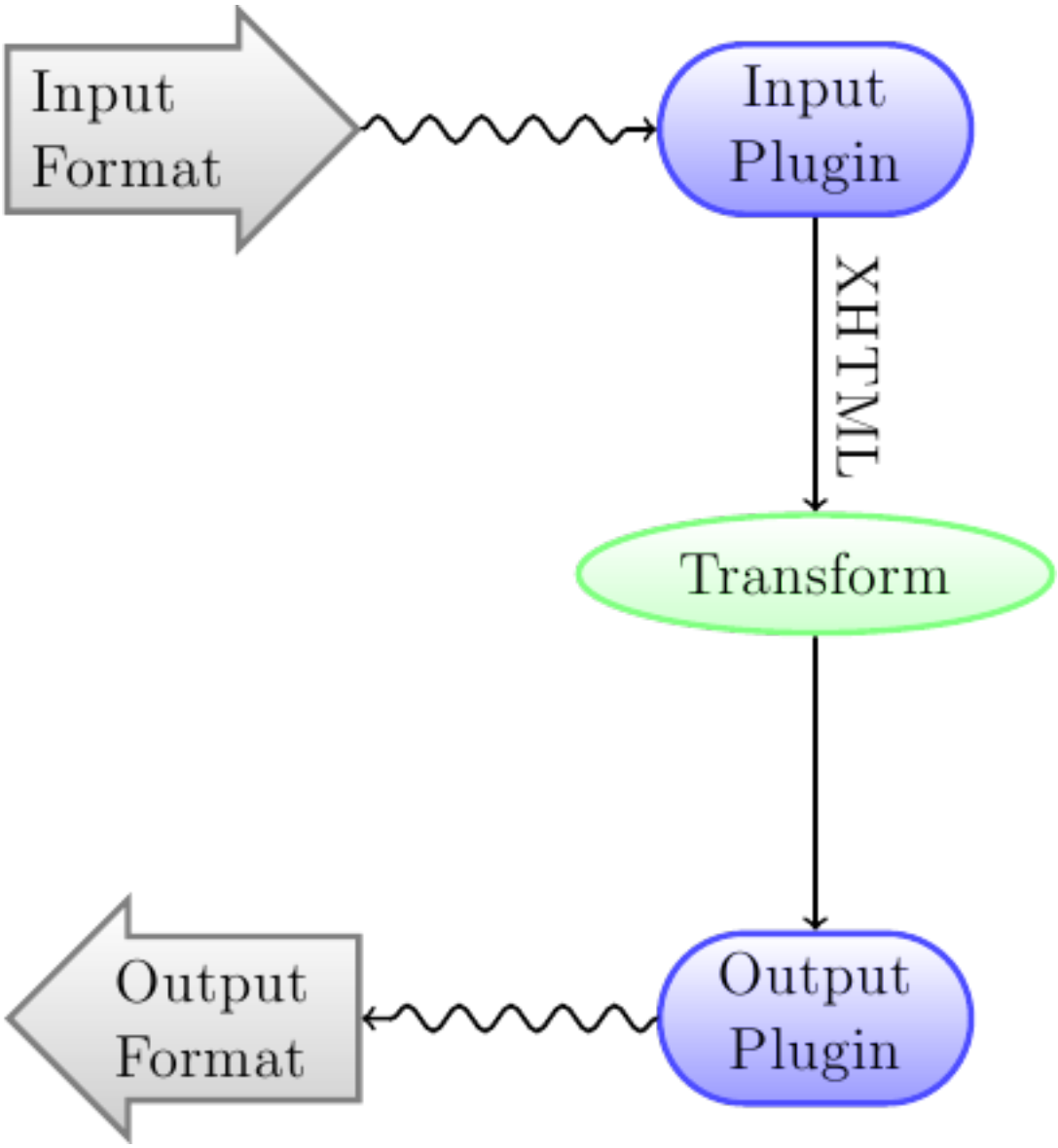
Inhalt

- *Einführung* (die Titelseite 63)
- *Erscheinungsbild* (die Titelseite 64)
- *Seiteneinrichtung* (die Titelseite 67)
- *Heuristische Verarbeitung* (die Titelseite 67)
- *Suchen & Ersetzen* (die Titelseite 69)
- *Struktur-Erkennung* (die Titelseite 69)
- *Inhaltsverzeichnis* (die Titelseite 70)
- *Verwenden von Bildern als Kapitelüberschriften beim Konvertieren von HTML-Quelldokumenten* (die Titelseite 72)
- *Verwenden von Tag-Attributen als Text für Inhaltsverzeichniseinträge* (die Titelseite 72)
- *How options are set/saved for conversion* (die Titelseite 72)

- *Formatspezifische Tipps* (die Titelseite 73)

4.1 Einführung

Zuerst sollten Sie verstehen, wie der Konvertierungsprozess strukturiert ist. Schematisch läuft eine Konvertierung wie folgt ab:



Das Eingabeformat wird zunächst durch das entsprechende *Eingabe-Plugin* in XHTML umgewandelt. Dieses HTML wird dann *transformiert*. Im letzten Schritt wird das verarbeitete XHTML durch das entsprechende *Output-Plugin* in das angegebene Ausgabeformat umgewandelt. Die Ergebnisse der Konvertierung können je nach Eingabeformat sehr unterschiedlich ausfallen. Manche Formate lassen sich viel besser konvertieren als andere. Eine Liste der besten Quellformate für die Konvertierung finden Sie *hier* (die Titelseite 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:

Tab. 1: Stadien des Konvertierungsprozesses

Ordner	Beschreibung
input	This contains the HTML output by the Input plugin. Use this to debug the Input plugin.
parsed	The result of pre-processing and converting to XHTML the output from the Input plugin. Use to debug structure detection.
Struktur	Post structure detection, but before CSS flattening and font size conversion. Use to debug font size conversion and CSS transforms.
abgeschlossen	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 Erscheinungsbild

Inhalt

- *Schriftarten* (die Titelseite 65)
- *Text* (die Titelseite 66)
- *Layout* (die Titelseite 66)
- *Styling* (die Titelseite 66)
- *Styles umwandeln* (die Titelseite 67)
- *Transform HTML* (die Titelseite 67)

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

4.2.1 Schriftarten

Eines der angenehmsten Merkmale der e-reading Erfahrung ist die Möglichkeit Schriftgrößen einfach an individuelle Bedürfnisse und Lichtverhältnisse anpassen zu können. Calibre hat raffinierte Algorithmen um sicher zu stellen, dass alle ausgegebenen Bücher konsistente Schriftgrößen haben, unabhängig davon welche Schriftgrößen im Quelldokument angegeben sind.

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 *Seiteneinrichtung* (die Titelseite 67)). However, you can override this here in case the default is not suitable for you.

Die *Font size key* option ermöglicht Ihnen einzustellen, wie Schriftgrößen außer der Basisschriftgröße skaliert werden. Der Schriftgrößenskalierungsalgorithmus arbeitet mit einem Schriftgrößenschlüssel, der einfach eine kommagetrennte Liste an Schriftgrößen ist. Der Schriftgrößenschlüssel gibt an, wie viele „Stufen“ größer oder kleiner als die Basisschriftgröße eine gegebene Schriftgröße sein soll. Die Idee dahinter ist, dass es eine begrenzte Anzahl an Schriftgrößen in einem Dokument geben sollte. Zum Beispiel eine Größe für den Hauptteil, ein paar Größen für verschiedene Stufen an Überschriften und ein paar Größen für Super-/Subskripts und Fußnoten. Der Schriftgrößenschlüssel ermöglicht Callibre die Schriftgrößen aus den Quelldokumenten in verschiedene Kategorien einzuteilen, die den unterschiedlichen logischen Schriftgrößen entsprechen.

Lassen Sie uns ein Beispiel geben. Angenommen das Quelldokument, dass wir konvertieren wurde von jemandem mit perfektem Sehvermögen erstellt und hat eine Grund-Schriftgröße von 8pt. Dies bedeutet das der Großteil des Texts im Dokument eine Größe von 8pt hat, während Überschriften etwas größer sind (beispielsweise 10 und 12pt) und Fußnoten etwas kleiner mit 6pt. Wenn wir nun die folgenden Einstellungen verwenden:

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

Das Zieldokument wird eine Basisschriftgröße von 12pt, Überschriften von 14 und 16pt und Fußnoten von 8pt haben. Nehmen wir an wir möchten die größte Überschriftengröße hervorheben und die Fußnoten auch etwas vergrößern. Um dies zu erreichen sollte der Schriftgrößenschlüssel wie folgt geändert werden:

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

Die größten Überschriften werden nun 18pt groß, während die Fußnoten 9pt groß werden. Sie können mit diesen Einstellungen herumspielen um zu versuchen ein Optimum zu finden, indem Sie den Schriftgrößenskalierungsdialog nutzen, auf den über die kleine Schaltfläche neben der *Font size key* Einstellung zugegriffen werden kann.

Die Schriftgrößenskalierung der Konvertierung kann auch deaktiviert werden, falls Sie die Schriftgrößen des Eingabedokuments beibehalten möchten.

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* (die Titelseite 66).

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 Text

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, *Unsmarted 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 *Wie konvertiere ich meine Datei mit nicht-englischen Zeichen oder Anführungszeichen?* (die Titelseite 130) for more on encoding issues.

4.2.3 Layout

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 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* (die Titelseite 66):

```
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 Styles umwandeln

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 Seiteneinrichtung

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 Heuristische Verarbeitung

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 <img> 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 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 Suchen & Ersetzen

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 leaves 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 *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214).

4.6 Struktur-Erkennung

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 Kapitel und Seitenumbrüche

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 location 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* (die Titelseite 158) 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 *Einführung* (die Titelseite 63) 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.

Die Standardeinstellung zur Erkennung von Seitenumbrüchen ist:

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

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

Bemerkung

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

4.6.2 Verschiedenes

In dieser Sektion gibt es noch einige weitere Optionen.

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.

Erstes Bild entfernen
: Manche Quelldateien, die Sie konvertieren, enthalten das Titelbild als Teil des Buches, anstatt als separates Bild. Falls Sie ein weiteres Titelbild in Calibre angeben, hat das konvertierte Buch zwei Titelbilder. Diese Option entfernt einfach das erste Bild des Quelldokuments und stellt auf diese Weise sicher, dass das konvertierte Buch nur ein einziges Titelbild hat, und zwar das welches in Calibre angegeben wurde.

4.7 Inhaltsverzeichnis

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.

Bemerkung

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 *Struktur-Erkennung* (die Titelseite 69) 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 *Einführung* (die Titelseite 63) for how to get access to this XHTML. Also read the *XPath-Tutorial* (die Titelseite 158), 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
```

Warnung

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 Verwenden von Bildern als Kapitelüberschriften beim Konvertieren von HTML-Quelldokumenten

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"><img src="chapter2.jpg" /></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 Verwenden von Tag-Attributen als Text für Inhaltsverzeichniseinträge

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.

Bemerkung

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 Formatspezifische Tipps

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 Microsoft Word-Dokumente konvertieren

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

Bemerkung

There is a demo .docx file[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.

Ältere .doc-Dateien

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 TXT-Dokumente konvertieren

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.

[28] https://calibre-ebook.com/downloads/demos/demo.docx

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
Nimmt an, dass jede Zeile ein Absatz ist:

```
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 though 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[29].

Formatting style: None
Applies no special formatting to the text, the document is converted to HTML with no other changes.

[29] https://daringfireball.net/projects/markdown/syntax

4.11.3 PDF-Dokumente konvertieren

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 *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214).

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 und Inhaltsverzeichnisse werden nicht unterstützt
- 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 OCRed text behind them. In such cases calibre uses the OCRed 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.

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
```

Die .cbc Datei enthält dann:

```
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 ODT-Dokumente konvertieren

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 to small, and with good aspect-ratio) as the cover image.

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.

[30] https://calibre-ebook.com/downloads/demos/demo.epub
[31] https://calibre-ebook.com/downloads/demos/demo.zip

4.11.7 Konvertieren zu 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.

Kopf- und Fußzeilen

Sie können beliebige Kopf- und Fußzeilen auf jeder Seite des PDF einfügen, indem Sie Kopf-und Fußzeilenvorlagen festlegen. Vorlagen sind einfach Schnipsel von HTML Code, die an den Positionen von Kopf- und Fußzeile dargestellt werden. Um zum Beispiel Seitenzahlen zentriert unten auf jeder Seite und in Grün anzuzeigen nutzen Sie die folgende Fußzeilenvorlage:

```
<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>
```

Dies wird den Titel auf der linken Seite und den Autor an der rechten Seite anzeigen, in einer Schriftgröße kleiner als der Haupttext .

Sie können auch den aktuellen Abschnitt in Vorlagen nutzen, wie nachfolgend gezeigt:

```
<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
- `_WIDTH_PIXELS_` - the width of the header/footer area in pixels
- `_HEIGHT_PIXELS_` - the height of the header/footer area in pixels

Bemerkung

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.

Ausdruckbares Inhaltsverzeichnis

Es kann auch ein ausdruckbares Inhaltsverzeichnis am Ende des PDF eingefügt werden, das die Seitenzahlen für jeden Abschnitt auflistet. Dies ist sehr sinnvoll, wenn Sie die PDF-Datei auf Papier ausdrucken möchten. Falls Sie die PDF-Datei auf einem elektronischen Gerät verwenden möchten, dann bietet die PDF-Gliederung diese Funktionalität und wird standardmäßig erstellt.

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>
```

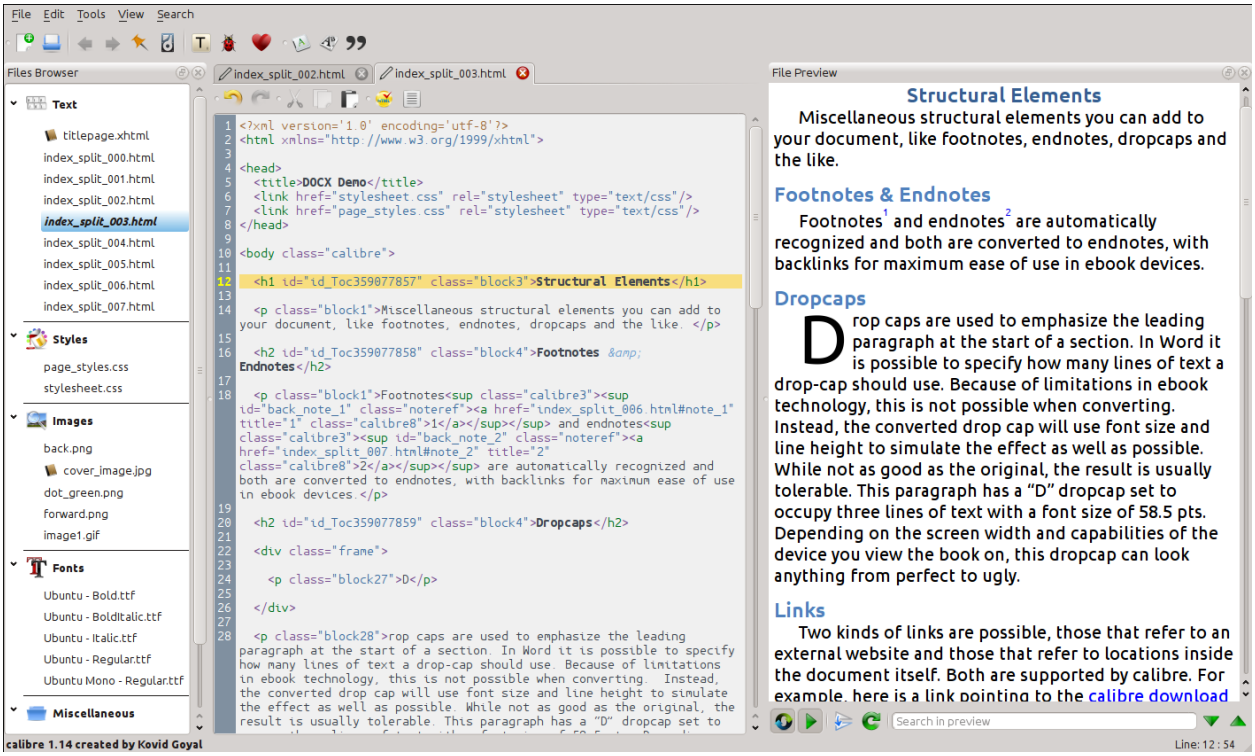
Then, in the PDF output section of the conversion dialog, turn on the option to *Use page margins from the document being converted*. Now all pages generated from this HTML file will have `10pt` margins.

KAPITEL 5

Bearbeiten von eBooks

calibre has an integrated e-book editor that can be used to edit books in the EPUB, KEPUB (Kobo) 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.

Sie können den Editor verwenden, indem Sie in Calibre auf ein Buch rechtsklicken und *Buch bearbeiten* auswählen.



Inhalt

- *Grundlagen* (die Titelseite 81)
- *Der Dateibrowser* (die Titelseite 83)
  - *Umbenennen von Dateien* (die Titelseite 84)
  - *Zusammenführen von Dateien* (die Titelseite 84)
  - *Ändern der Textdateireihenfolge* (die Titelseite 84)
  - *Kennzeichnen des Titelbildes* (die Titelseite 85)
  - *Löschen von Dateien* (die Titelseite 85)
  - *Exportieren von Dateien* (die Titelseite 85)
  - *Hinzufügen neue Bilder, Schriftarten usw. oder Erstellen neuer, leerer Dateien* (die Titelseite 85)
  - *Ersetzen von Dateien* (die Titelseite 85)
  - *Effizientes Verknüpfen von Stylesheets mit HTML-Dateien* (die Titelseite 85)
- *Suchen & Ersetzen* (die Titelseite 86)
  - *Gespeicherte Suchen* (die Titelseite 86)
  - *Funktionsmodus* (die Titelseite 86)
  - *Suchen (HTML-tags ignorieren)* (die Titelseite 87)
- *Automatisierte Werkzeuge* (die Titelseite 87)
  - *Bearbeiten des Inhaltsverzeichnisses* (die Titelseite 87)
  - *Checking the book* (die Titelseite 88)
  - *Hinzufügen eines Titelbildes* (die Titelseite 89)
  - *Einbetten referenzierter Schriftarten* (die Titelseite 89)
  - *Reduzieren eingebetteter Schriftarten* (die Titelseite 89)
  - *Typografisieren von Satzzeichen* (die Titelseite 89)
  - *Umwandeln von CSS-Eigenschaften* (die Titelseite 89)
  - *Entfernen unbenutzter CSS-Regeln* (die Titelseite 90)
  - *Reparieren von HTML* (die Titelseite 90)
  - *Verschönern von Dateien* (die Titelseite 90)
  - *Einfügen eines Inline-Inhaltsverzeichnisses* (die Titelseite 90)
  - *Festlegen der Semantik* (die Titelseite 90)
  - *Filtern von style-Informationen* (die Titelseite 91)
  - *Buchstrukturen upgraden* (die Titelseite 91)
- *Sicherungspunkte* (die Titelseite 91)
- *Die Echtzeit-Dateivorschau* (die Titelseite 93)
  - *Aufteilen von HTML-Dateien* (die Titelseite 94)
- *Der Echtzeit-CSS-Bereich* (die Titelseite 95)

- *Verschiedene Werkzeuge* (die Titelseite 96)
  - *Die Inhaltsverzeichnis-Ansicht* (die Titelseite 96)
  - *Überprüfen der Rechtschreibung von Wörtern im Buch* (die Titelseite 96)
  - *Einfügen von Sonderzeichen* (die Titelseite 98)
  - *Die Codeinspektor-Ansicht* (die Titelseite 99)
  - *Überprüfen externer Verweise* (die Titelseite 99)
  - *Externe Ressourcen werden heruntergeladen* (die Titelseite 99)
  - *Ordnen von Dateien in Ordner nach Typ* (die Titelseite 99)
  - *Importieren von Dateien in anderen eBook-Formaten als EPUB* (die Titelseite 99)
  - *Das Berichte-Werkzeug* (die Titelseite 109)
- *Besondere Funktionen des Code-Editors* (die Titelseite 110)
  - *Syntaxhervorhebung* (die Titelseite 110)
  - *Kontextspezifische Hilfe* (die Titelseite 110)
  - *Automatisches Vervollständigen* (die Titelseite 111)
  - *Snippets* (die Titelseite 111)

5.1 Grundlagen

Bemerkung

A video tour of the calibre E-book editor is available here[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[33] and CSS Tutorial[34].

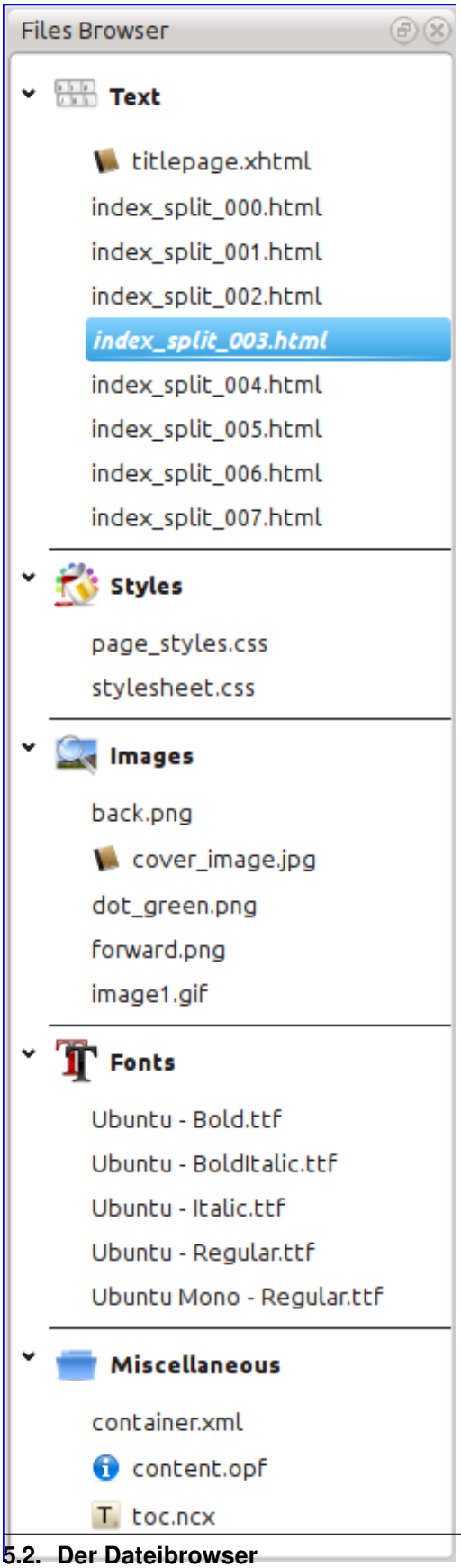
Alle Änderungen, die Sie im Editor an HTML- oder CSS-Dateien vornehmen, werden direkt rechts im Vorschaubereich angezeigt. Wenn Sie mit dem Aussehen der Änderungen zufrieden sind, klicken Sie auf die Schaltfläche „Speichern“ oder verwenden Sie *Datei → Speichern*, um Ihre Änderungen im eBook zu speichern.

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.

Das ist das normale Vorgehen im Editor – Eine Datei öffnen, Änderungen vornehmen, Vorschau ansehen und Speichern. Der Rest dieses Handbuches wird sich mit den verschiedenen Werkzeugen und Funktionen auseinandersetzen, die es Ihnen erlauben, bestimmte Aufgaben effizient zu lösen.

[32] https://calibre-ebook.com/demo#tutorials
[33] https://www.w3schools.com/Html/default.asp
[34] https://www.w3schools.com/css/default.asp

5.2 Der Dateibrowser



Der *Dateibrowser* gibt Ihnen einen Überblick über die verschiedenen Dateien innerhalb des Buches, das Sie bearbeiten. Die Dateien sind nach Kategorien sortiert, mit Textdateien (HTML) am Anfang, gefolgt von Stylesheets (CSS), Bildern und so weiter. Klicken Sie einfach auf eine Datei, um mit dem Bearbeiten zu beginnen. Es können nur HTML-, CSS- und Bilddateien bearbeitet werden. Die Textdateien werden in der gleichen Reihenfolge angezeigt, in der Sie das Buch lesen würden. Alle anderen Dateien sind alphabetisch sortiert.

Wenn Sie die Maus über einen Eintrag bewegen, können Sie die Größe und am unteren Fensterrand den vollständigen Pfad dieser Datei im Buch sehen. Beachten Sie, dass Dateien innerhalb von eBooks komprimiert sind, sodass sich die Größe des fertigen Buches nicht aus der Summe der einzelnen Dateigrößen ergibt.

Einige Dateien in einem eBook haben eine spezielle Bedeutung. Diese haben normalerweise ein Symbol neben ihrem Namen, das auf die spezielle Bedeutung hinweist. Im Bild links können Sie sehen, dass die Dateien *cover_image.jpg* und *titlepage.xhtml* das Symbol für ein Titelbild haben, was bedeutet, dass es sich um das Titelbild des Buches und um die Titelseite handelt. In ähnlicher Weise hat die *content.opf* Datei ein Metadaten-Symbol, was bedeutet, dass die Metadaten des Buches in dieser Datei gespeichert sind. Die *toc.ncx* Datei wird durch ein „T"-Symbol gekennzeichnet, was für das Inhaltsverzeichnis steht.

Sie können viele Aktionen auf einzelne Dateien anwenden, indem Sie diese rechtsklicken.

5.2.1 Umbenennen von Dateien

Sie können eine einzelne Datei umbenennen, indem Sie sie rechtsklicken und *Umbenennen* auswählen. Das Umbenennen einer Datei aktualisiert automatisch alle Verweise und Referenzen auf die Datei im gesamten Buch. Alles, was Sie also tun müssen, ist lediglich die Vergabe eines neuen Namens. Calibre kümmert sich um den 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.

Zum Schluss: Sie können die Dateinamenerweiterung aller ausgewählten Dateien gleichzeitig ändern. Markieren Sie mehrere Dateien, wie oben beschrieben, und wählen Sie nach einem Rechtsklick *Dateinamenerweiterung für ausgewählte Dateien ändern*.

5.2.2 Zusammenführen von Dateien

Manchmal will man zwei HTML-Dateien oder zwei CSS-Dateien zusammenführen. Es kann ja nützlich sein, alles in einer einzigen Datei zu haben. Seien Sie aber vorsichtig: Wenn Sie zu viel Inhalt in eine einzige Datei packen, kann es Leistungsprobleme beim Betrachten des Buchs in normalen eBook-Readern geben.

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 Ändern der Textdateireihenfolge

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 Kennzeichnen des Titelbildes

Normalerweise haben eBooks ein Titelbild. Dieses Bild wird im Dateibrowser mit einem braunen Buch-Symbol neben der Bildbezeichnung angezeigt. Wenn Sie ein anderes Titelbild möchten, können Sie es per Rechtsklick markieren und danach *Als Titelbild markieren* auswählen.

Zusätzlich haben EPUB-Dateien eine *Titelseite*. Eine Titelseite ist eine HTML-Datei, die als Cover bzw. Titelseite fungiert. Sie können eine HTML-Datei mit einem Rechtsklick als Titelseite festlegen, wenn Sie EPUBs bearbeiten. Achten Sie darauf, dass die so markierte Datei nur die Titelbild-Informationen enthält. Wenn sich dort noch andere Inhalte befinden, wie zum Beispiel das erste Kapitel, dann gehen diese verloren, wenn der Nutzer die EPUB-Datei in Calibre in ein anderes Format umwandelt. Der Grund ist, dass Calibre beim Konvertieren davon ausgeht, dass die markierte Titelseite ausschließlich das Titelbild enthält.

5.2.5 Löschen von Dateien

Sie können Dateien löschen, indem Sie diese entweder rechtsklicken oder indem Sie sie auswählen und dann auf Löschen klicken. Das Löschen entfernt auch alle Verweise zu der Datei aus der OPF-Datei, um Ihnen diese Arbeit abzunehmen. Aber andernorts bleiben die Verweise bestehen. Sie können das Werkzeug „Buch prüfen“ verwenden, um alle Verweise zu finden und sie zu löschen oder zu ersetzen.

5.2.6 Exportieren von Dateien

Sie können eine Datei aus einem Buch heraus auf Ihren Computer exportieren. Das ist nützlich wenn Sie isoliert mit speziellen Programmen an dieser Datei arbeiten wollen. Für den Export rechtsklicken Sie auf die Datei und wählen *exportieren*.

Sobald Sie mit dem Bearbeiten der exportierten Datei fertig sind, können Sie diese in das Buch re-importieren, indem Sie erneut auf die Datei rechtsklicken und *Ersetzen durch die Datei …* auswählen. Damit können Sie die Datei im Buch mit der zuvor exportierten Datei ersetzen.

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 Hinzufügen neue Bilder, Schriftarten usw. oder Erstellen neuer, leerer Dateien

Sie können ein neues Bild, eine Schriftart, ein Stylesheet usw. von Ihrem Computer in das Buch einfügen, indem Sie auf *Datei → Neue Datei* klicken. Dabei können Sie entweder eine Datei importieren, indem Sie auf *Quelldatei importieren* klicken, oder Sie legen eine neue, leere HTML-Datei oder ein Stylesheet an. Geben Sie dafür einfach einen Dateinamen in das Feld für die neue Datei ein.

Sie können auch mehrere Dateien auf einmal in das Buch einfügen, indem Sie Datei → Dateien in das Buch importieren auswählen.

5.2.8 Ersetzen von Dateien

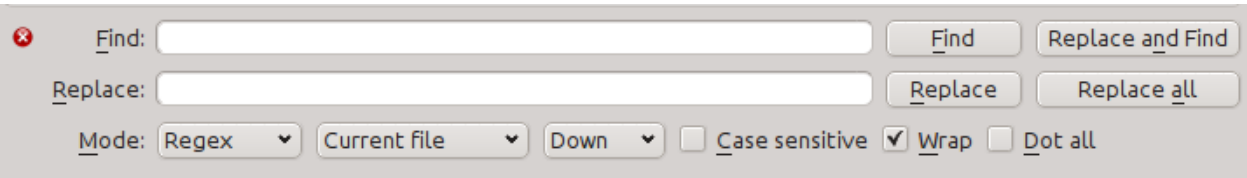
Sie können ganz einfach bestehende Dateien im Buch ersetzen, indem Sie auf die Datei rechtsklicken und „Ersetzen durch die Datei …“ wählen. Damit werden automatisch alle Links und Verweise aktualisiert, falls die neue Datei einen anderen Namen hat als die alte Datei.

5.2.9 Effizientes Verknüpfen von Stylesheets mit HTML-Dateien

Praktischerweise können Sie mehrere HTML-Dateien im Dateibrowser auswählen, dann rechtsklicken und „Stylesheets verknüpfen“ anwählen, damit Calibre automatisch <link>-Tags für diese Stylesheets in alle ausgewählten HTML-Dateien einfügt.

5.3 Suchen & Ersetzen

„Buch bearbeiten“ besitzt eine sehr leistungsstarke „Suchen und Ersetzen“-Funktion. Sie können damit einen Text in der aktuellen Datei suchen und ersetzen, aber auch über alle Dateien hinweg oder nur in einem markierten Teil der aktuellen Datei. Sie können eine einfache Suche durchführen oder reguläre Ausdrücke (Regex) verwenden. Um die Verwendung von regulären Ausdrücken zu lernen, schlagen Sie unter *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214) nach.



Starten Sie Suchen und Ersetzen über den Menüeintrag *Suchen → Suchen/Ersetzen* (falls Sie dabei sind, eine HTML- oder CSS-Datei zu bearbeiten).

Geben Sie den zu suchenden Text in das Feld „Suchen“ und dessen Ersatz in das Feld „Ersetzen“ ein. Sie können die entsprechenden Schaltflächen anklicken, um die nächste Übereinstimmung zu finden, die aktuelle Übereinstimmung zu ersetzen, oder aber alle Übereinstimmungen zu ersetzen.

Über die Dropdown-Listen am unteren Ende des Feldes können Sie bestimmen, ob die Suche im aktuellen Text läuft, in allen Textdateien, allen style-Dateien oder generell allen Dateien. Außerdem können Sie auswählen, ob Sie normal (Zeichenkette) oder mittels eines regulären Ausdrucks suchen wollen.

Sie können alle Fundstellen für einen Suchausdruck über *Suchen → Alle(s) zählen* zählen. Die Zählung umfasst dabei alle Dateien und Regionen, die Sie in der Dropdown-Liste ausgewählt haben.

Sie können auch mit *Suchen → Zu Zeile gehen* im momentan geöffneten Editor zu einer bestimmen Zeile wechseln.

> **Bemerkung**
>
> Nutzen Sie alle Möglichkeiten von „Suchen und Ersetzen“, indem Sie reguläre Ausdrücke verwenden. Mehr dazu unter *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214).

5.3.1 Gespeicherte Suchen

Suchbegriffe (auch reguläre Ausdrücke), nach denen Sie häufig suchen, können Sie speichern und dann bei Bedarf immer wieder nutzen. Um eine Suche zu speichern, rechtsklicken Sie einfach auf das Suchfeld und wählen Sie *Aktuelle Suche speichern*.

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 Funktionsmodus

Der Funktionsmodus erlaubt Ihnen, beliebig mächtige Python-Funktionen zu schreiben, die bei jedem Suchen/Ersetzen ausgeführt werden. Sie können so ziemlich jede gewünschte Textmanipulation im Funktionsmodus durchführen. Für weitere Informationen siehe *Function mode for Search & replace in the Editor* (die Titelseite 99).

5.3.3 Suchen (HTML-tags ignorieren)

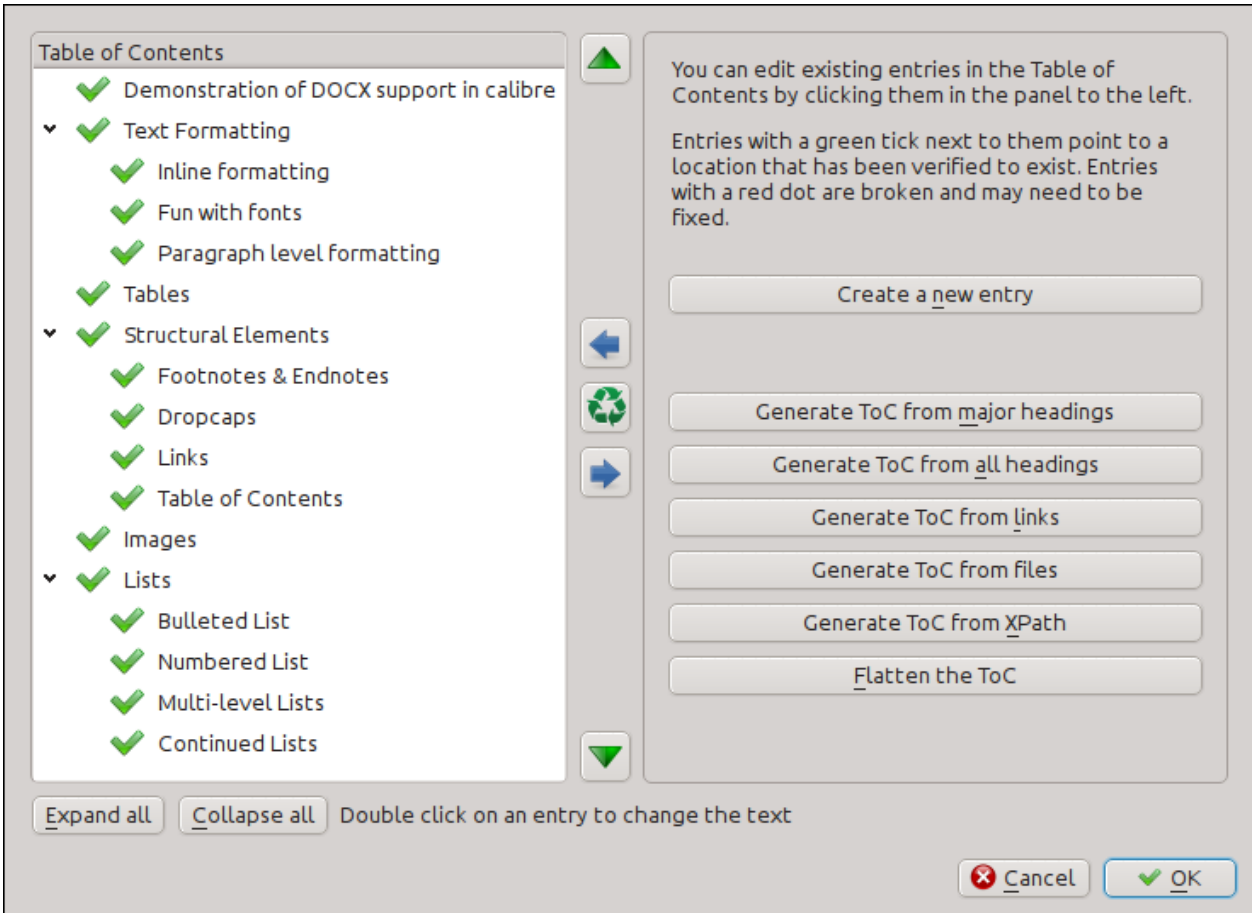
Es existiert auch ein spezielles Werkzeug zur Textsuche, das HTML-tags ignoriert. Enthält das Buch zum Beispiel das HTML „Betonung eines <i>Wortes<i>“, kann mit dem Text „eines Wortes“ gesucht werden und die Stelle wird trotz des Tags „<i>“ gefunden werden. Das Werkzeug steht unter dem Menüpunkt ‚Suchen -> Suchen (HTML Markup ignorieren)‘ zur Verfügung.

5.4 Automatisierte Werkzeuge

‚Buch bearbeiten‘ hat verschiedene Werkzeuge, die Ihnen bei alltäglichen Aufgaben helfen. Sie sind über das Menü *Werkzeuge* zu erreichen.

5.4.1 Bearbeiten des Inhaltsverzeichnisses

Es gibt ein eigenes Werkzeug für eine leichtere Bearbeitung des Inhaltsverzeichnisses. Rufen Sie es über *Werkzeuge* → *Inhaltsverzeichnis* → *Inhaltsverzeichnis bearbeiten* auf.

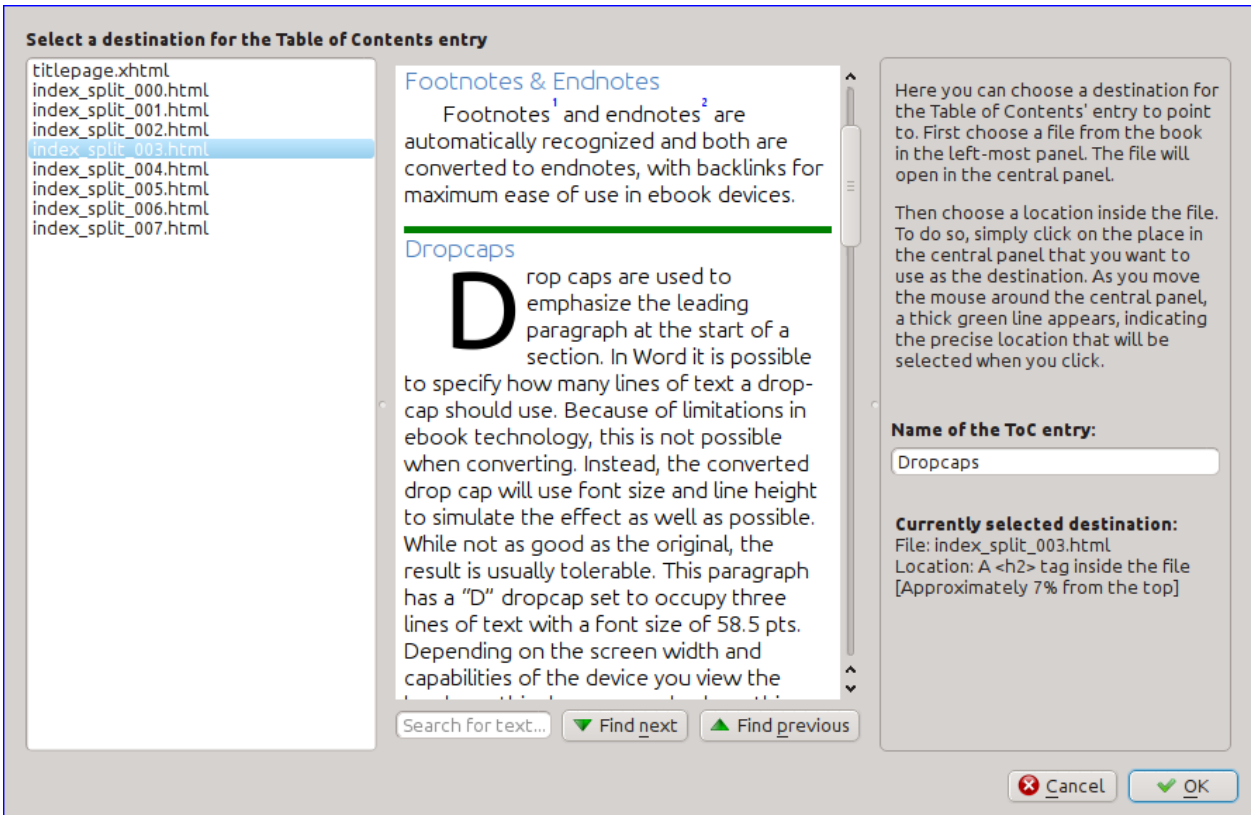


Das Werkzeug „Inhaltsverzeichnis bearbeiten“ zeigt Ihnen das aktuelle Inhaltsverzeichnis (soweit vorhanden) auf der linken Seite. Den Text eines Eintrags können Sie nach einem Doppelklick ändern. Sie können Einträge auch durch Ziehen und Ablegen oder über die Schaltflächen auf der rechten Seite verschieben.

Für Bücher ohne ein bestehendes Inhaltsverzeichnis bietet Ihnen dieses Werkzeug verschiedene Möglichkeiten eines aus dem Text anzulegen. Sie können dafür die Überschriften, Links im Dokument verwenden oder eigene Dateien.

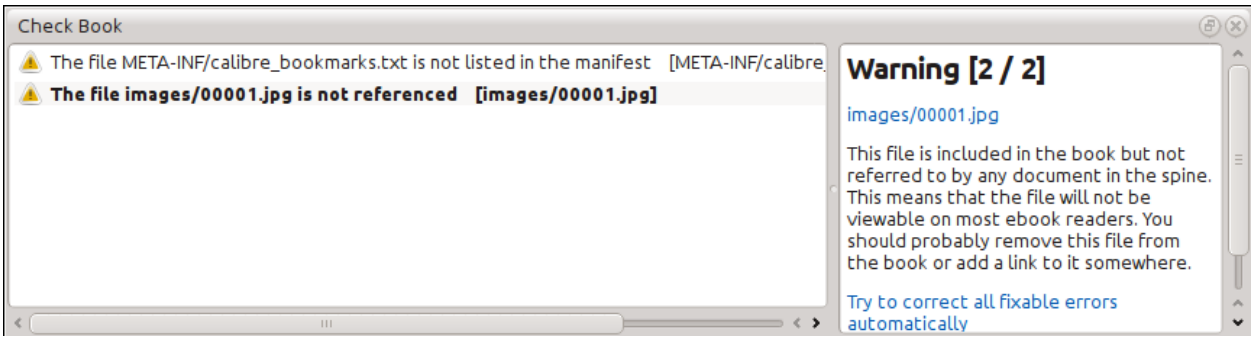
Sie können einzelne Beiträge bearbeiten, indem Sie diese anklicken und dann *Den Ort ändern, auf den dieser Eintrag zeigt*. Danach öffnet sich eine Minivorschau des Buches. Bewegen Sie den Mauszeiger einfach über den Buchansichtsbereich

und klicken Sie dorthin, wohin der Eintrag verweisen soll. Eine dicke grüne Linie zeigt die Position an. Bestätigen Sie danach mit OK.



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*.



Falls Fehler aufgetreten sind, werden sie übersichtlich aufgelistet. Wenn Sie auf einen der Einträge klicken, bekommen Sie Hinweise zum Fehler und können ihn unter Umständen automatisch reparieren lassen. Mit einem Doppelklick gelangen Sie in einem Editor zu der Stelle, an der der Fehler aufgetreten ist um ihn selbst zu korrigieren.

Einige der durchgeführten Überprüfungen sind:

- Ungültige HTML-Markups. Alle HTML-Markups, die nicht als wohlgeformtes XML parsen, werden angezeigt. Damit Ihre Markups in jedem Zusammenhang richtig funktionieren, sollten Sie diese Stellen korrigieren. Calibre kann diese Fehler auch automatisch korrigieren, was aber manchmal zu unvorhergesehenen Ergebnissen führt. Benutzen Sie diese Funktion also mit Vorsicht. Vor der automatischen Korrektur wird aber wie immer eine Sicherung erstellt, sodass Sie alle Änderungen rückgängig machen können. Die Autokorrektur nutzt für das Parsen des

Markups den HTML5-Algorithmus, der sehr fehlertolerant ist und wandelt diesen dann in wohlgeformtes XML um.

- Fehlgeformte oder unbekannte CSS-Styles. Jedes CSS, das ungültig ist oder Eigenschaften besitzt, die nicht im CSS 2.1-Standard (plus ein paar aus CSS 3) definiert wurden, wird gemeldet. CSS wird in allen Stylesheets, Inline-style-Attributen und <style>-Tags in HTML-Dateien überprüft.
- Tote Links. Alle Links, die zu Dateien führen, die es im Buch nicht gibt, werden aufgelistet.
- Verwaiste Dateien. Dateien, auf die es weder im Buchrücken noch in einer anderen Datei einen Hinweis gibt, werden aufgelistet.
- Verschiedene übliche Probleme in OPF-Dateien, wie z.B. doppelte Buchrücken oder manifeste Bestandteile, kaputte idrefs oder Meta-Titelbild-Marker, fehlende Bereiche und so weiter.
- Verschiedene Kompatibilitätsprüfungen für bekannte Probleme, die verhindern, dass das Buch auf einem eBook-Reader angezeigt wird.

5.4.3 Hinzufügen eines Titelbildes

Sie können ein Titelbild über *Werkzeuge → Titelbild hinzufügen* hinzufügen. Wählen Sie ein bestehendes Bild aus oder importieren Sie ein neues in das Buch um es zum Titelbild zu machen. Bei der Bearbeitung von EPUB-Dateien, wird der HTML-Wrapper automatisch erzeugt. Gibt es schon ein Titelbild für das Buch, wird dieses ersetzt. Das Werkzeug markiert die Titelbild-Dateien auch automatisch als Titelbild im OPF.

5.4.4 Einbetten referenzierter Schriftarten

Über *Werkzeuge → Einbetten von referenzierten Schriftarten* finden Sie alle Schriftarten, die im Buch referenziert sind und ob sie bereits eingebunden wurden. Das Werkzeug durchsucht Ihren Computer und bindet die Schriftarten im Buch ein, wenn sie vorhanden sind. Stellen Sie bitte sicher, dass Sie die nötigen Nutzungsrechte für kommerziell genutzte Schriftarten haben, bevor Sie dieses Werkzeug nutzen.

5.4.5 Reduzieren eingebetteter Schriftarten

Über *Werkzeuge → Eingebettete Schriften reduzieren* können Sie alle Schriftarten im Buch auf die tatsächlich verwendeten Zeichen reduzieren. In der Regel werden die Schriftart-Dateien dadurch um ca. 50 % kleiner. Beachten Sie dabei aber bitte, dass jeglicher neuer Text, der eingefügt wird nur die auch schon bisher in dieser Schriftart vorhandenen Zeichen enthalten kann, weil die Schriftart sonst nicht funktioniert. Die Reduktion sollte also immer erst als letzter Schritt erfolgen.

5.4.6 Typografisieren von Satzzeichen

Gewöhnliche Gedankenstriche, Auslassungspunkte, Anführungszeichen, mehrere Bindestriche in Folge usw. in ihre typografisch korrekten Entsprechungen umwandeln. Beachten Sie, dass der Algorithmus manchmal zu falschen Ergebnissen führen kann, insbesondere dann, wenn einfache Anführungszeichen am Beginn von Kontraktionen verwendet werden. Zugriff über *Werkzeuge → Satzzeichen typografisieren*.

5.4.7 Umwandeln von CSS-Eigenschaften

Erstellen Sie Regeln zum Umwandeln des Stylings des Buchs. Erstellen Sie beispielsweise eine Regel, um allen roten Text in grünen zu konvertieren, oder, um die Schriftgröße des gesamten Textes des Buchs zu verdoppeln, oder, um den Text einer bestimmten Schriftfamilie kursiv zu stellen usw.

Das Erstellen der Regeln ist einfach. Das Format der Regeln folgt der natürlichen Sprache und sieht so aus:

- Falls die Eigenschaft *color* (dt. Farbe) *red* (dt. rot) ist, *ändere* diese in *green* (dt. grün)
- Falls die Eigenschaft *font-size* (dt. Schriftgröße) *einem beliebigen Wert* entspricht, *multipliziere* den Wert mit *2*

Aufzurufen über *Werkzeuge → Styles umwandeln*.

5.4.8 Entfernen unbenutzter CSS-Regeln

Alle nicht verwendeten CSS-Regeln aus Stylesheets und <style>-Tags entfernen. Manche aus Produktionsvorlagen erstellte Bücher können zahlreiche zusätzliche CSS-Regeln enthalten, die keinem tatsächlichen Inhalt entsprechen. Diese zusätzlichen Regeln können Lesegeräte verlangsamen, die sämtliche Regeln analysieren müssen. Zugriff über *Werkzeuge → Nicht verwendete CSS-Regeln entfernen*.

5.4.9 Reparieren von HTML

Dieses Werkzeug wandelt einfach nur HTML, das nicht als XML geparst werden kann in wohlgeformtes XML um. In eBooks findet man nichtwohlgeformtes XML häufiger, weshalb dieses Werkzeug das Verbessern automatisiert. Dafür parst das Werkzeug das HTML mit dem HTML 5-Algorithmus (der in allen modernen Browsern verwendet wird) und wandelt das Ergebnis in XML um. Beachten Sie, dass die automatische Verbesserung manchmal zu unerwarteten Ergebnissen führt. Sie können über das bereits besprochene Werkzeug ‚Buch bearbeiten' HTML-Fehler finden und manuell beheben. Sie finden das Werkzeug unter *Werkzeuge → HTML reparieren - alle Dateien*.

5.4.10 Verschönern von Dateien

Dieses Werkzeug wird verwendet um alle HTML- und CSS-Dateien automatisch so zu formatieren, dass sie ‚hübsch' aussehen. Die Einrückung des Code wird automatisch angepasst, dass zusammengehörende Teile auf einer Linie sind, an passender Stelle Leerzeilen eingefügt werden usw. Beachten Sie dass dadurch auch ungültiger HTML/CSS-Code korrigiert wird. Deshalb sollten Sie das ‚Buch überprüfen'-Werkzeug verwenden um alle Fehler in Ihrem Sinne zu korrigieren, bevor Sie ‚Alle Dateien verschönern' verwenden. Zugriff auf dieses Werkzeug haben Sie über *Werkzeuge → Alle Dateien verschönern*.

Bemerkung

In HTML kann jeder Text über die CSS Whitespace-Direktive einen deutlichen Whitespace haben. Deshalb kann die Verschönerung die Umsetzung des HTML-Code möglicherweise verändern. Um das so weit wie möglich zu verhindern, verschönert der Algorithmus nur Block Level Marker, die andere Block Level Marker enthalten. Damit wird zum Beispiel der Whitespace von Text in einem <p> Marker nicht verändert. Aber ein <body> Marker, der nur andere <p> und Marker enthält, wird verschönert. Das bedeutet manchmal dass eine bestimmte Datei nicht von der Verschönerung verändert wird, da sie keine passenden Block Level Marker enthält. In diesen Fällen können Sie verschiedene Werkzeuge zur Verschönerung verwenden, die weniger vorsichtig vorgehen, z.B.: HTML Tidy[35].

5.4.11 Einfügen eines Inline-Inhaltsverzeichnisses

In eBooks liegt das Inhaltsverzeichnis normalerweise getrennt vom Haupttext vor und wird über eine eigene Schaltfläche oder ein Menü des eBook-Readers aufgerufen. Sie können von Calibre auch ein *eingebundenes* Inhaltsverzeichnis erstellen lassen, das Teil des Buchtextes ist. Es wird automatisch auf der Grundlage des aktuellen Inhaltsverzeichnisses erzeugt.

Wenn Sie dieses Werkzeug mehrmals verwenden, wird jedes Mal das zuvor erstellte Inline-Inhaltsverzeichnis ersetzt. Sie können hier auf das Werkzeug zugreifen: *Werkzeuge → Inhaltsverzeichnis → Inline-Inhaltsverzeichnis einfügen*.

5.4.12 Festlegen der Semantik

Dieses Werkzeug dient dazu eine *Semantik* in EPUB-Dateien festzulegen. Die Semantik wird einfach durch Links in der OPF-Datei gebildet, die bestimmte Orte im Buch mit einer besonderen Bedeutung kennzeichnen. Sie können sie verwenden, um das Vorwort, die Widmung, das Titelbild, das Inhaltsverzeichnis usw. zu kennzeichnen. Wählen Sie einfach die Art der Semantik-Informationen aus, die Sie festlegen möchten und danach den Ort im Buch, worauf der Link verweisen soll. Sie können auf dieses Werkzeug über *Werkzeuge → Semantik festlegen* zugreifen.

[35] https://infohound.net/tidy/

5.4.13 Filtern von style-Informationen

Sie können dieses Werkzeug verwenden, um festgelegte CSS-style-Attribute aus dem kompletten Buch zu entfernen. Sie können bestimmen, welche Eigenschaften entfernt werden sollen, zum Beispiel `color, background-color, line-height` (dt. Farbe, Hintergrundfarbe, Zeilenhöhe) und das Werkzeug wird sie überall, wo sie vorkommen, löschen — Stylesheets, `<style>`-Tags und Inline-`style`-Attribute. Nach dem Entfernen der style-Informationen wird eine Zusammenfassung aller Änderungen angezeigt, damit Sie diese nachvollziehen können. Das Werkzeug finden Sie unter *Werkzeuge → style-Attribute filtern*.

5.4.14 Buchstrukturen upgraden

Dieses Tool kann verwendet werden, um Buchstrukturen zu upgraden, wenn es möglich ist. Es ermöglicht beispielsweise das Upgraden von EPUB 2 Bücher auf EPUB 3. Auf das Tool kann über *Buchstrukturen upgraden* zugegriffen werden.

5.5 Sicherungspunkte

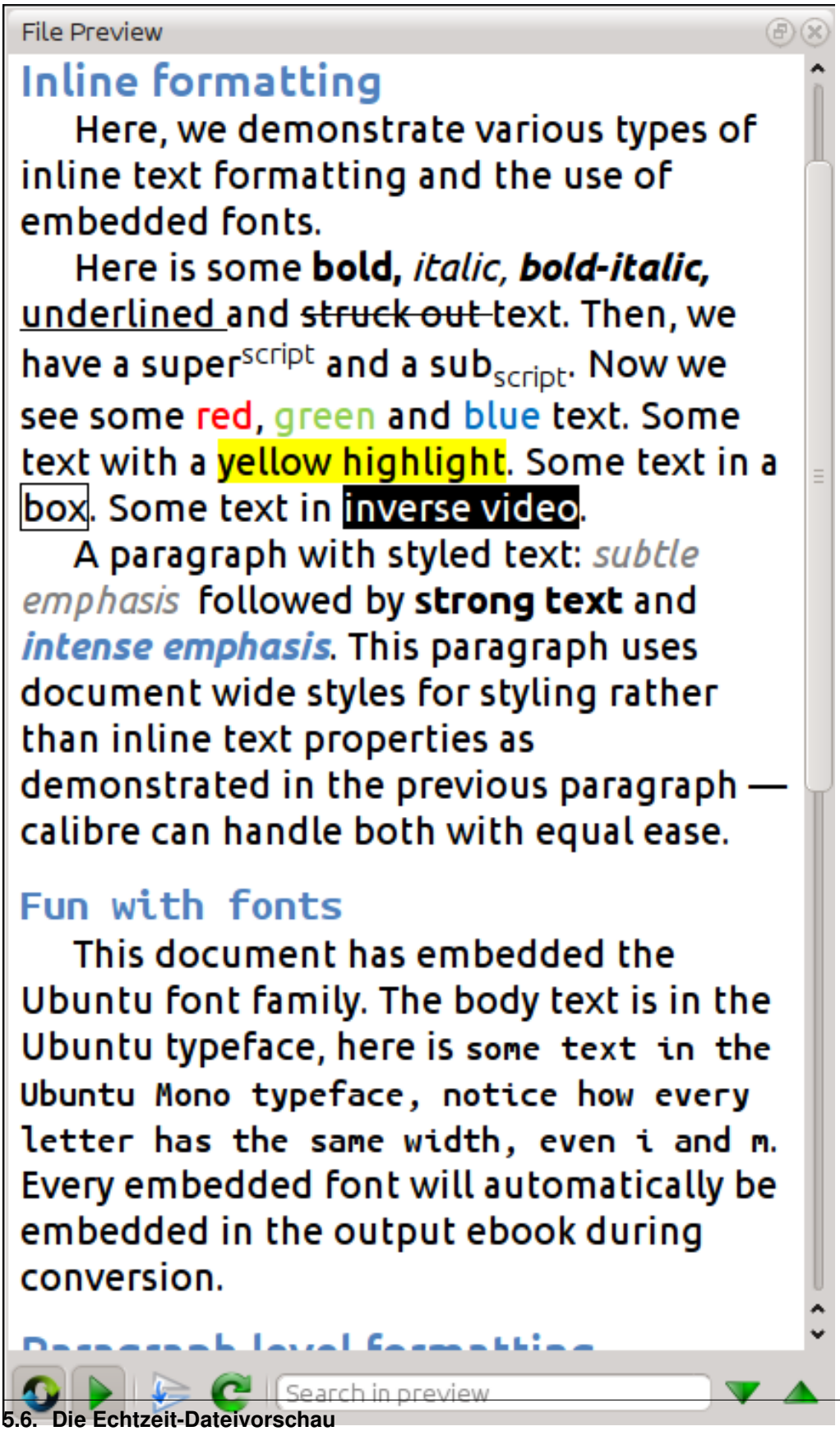
Sicherungspunkte ist eine Möglichkeit den momentanen Stand des Buches als „besonders“ zu kennzeichnen. Sie können das Buch dann ändern und jederzeit zu diesem besonderen Stand zurückkehren, wenn Ihnen die Änderungen nicht gefallen. Sicherungspunkte werden automatisch erstellt, wenn sie eines der automatischen Werkzeuge verwenden, von denen in den vorhergehenden Abschnitten die Rede war.

Sie können einen Sicherungspunkt über *Bearbeiten → Sicherungspunkt erstellen* anlegen. Zurück zu einem vorherigen Sicherungspunkt kommen Sie über *Bearbeiten → Rückgängig zu ….*

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.

Sie finden unter *Anzeigen → Sicherungspunkte* eine Liste der verfügbaren Sicherungspunkte. Die aktuelle Version des Buches lässt sich über das *Vergleichen von eBooks* (die Titelseite 121) Werkzeug mit anderen vergleichen, indem Sie den entsprechenden Sicherungspunkt auswählen und auf die *Vergleichen* Schaltfläche klicken. Die Schaltfläche *Rückgängig zu …* stellt den Zustand des Buchs zum ausgewählten Sicherungspunkt wieder her und verwirft alle Änderungen, die seit Anlegen dieses Sicherungspunkts gemacht wurden.

5.6 Die Echtzeit-Dateivorschau



Die *Dateivorschau* gibt Ihnen einen Überblick über die einzelnen Dateien. Der Bereich zeigt Ihnen Ihre Änderungen in Echtzeit (mit einer kurzen Verzögerung). Wenn Sie HTML- oder CSS-Dateien bearbeiten, wird der Vorschaubereich automatisch aktualisiert, um die Änderungen zu zeigen. Wenn Sie den Cursor im Editor bewegen, zeigt der Vorschaubereich die jeweilige Position im Buch an. Beim Klick in den Vorschaubereich springt der Cursor im Editor auf die angeklickte Stelle. Wenn Sie einen Verweis auf eine andere Datei im Buch anklicken, wird diese Datei automatisch im Editor und im Vorschaubereich geöffnet.

Sie können das automatische Synchronisieren der Position und Echtzeitvorschau der Änderungen mit den Schaltflächen unter dem Vorschaubereich deaktivieren. Die Echtzeitaktualisierung des Vorschaubereichs wird nur dann ausgeführt, wenn Sie gerade nichts in den Editor eingeben, damit Sie nicht abgelenkt werden oder Zeit verlieren, weil Sie auf die Dartstellung der Vorschau warten.

Der Vorschaubereich zeigt Ihnen, wie der Text in der Anzeige aussehen wird. Sie sollten Ihr Buch aber dennoch unbedingt auf einem Gerät testen, denn der Vorschaubereich kann teils mehr, teils weniger als ein E-Reader. Er toleriert Fehler und schlampige Markups viel besser als die meisten Geräte. Außerdem zeigt er keine Seitenränder, Seitenumbrüche und eingebettete Schriftarten, die das Aliasing von Schriftartnamen verwenden. Nutzen Sie den Vorschaubereich für die Arbeit an einem Buch und wechseln Sie zur Kontrolle zu einem Reader oder einem Software-Emulator.

Bemerkung

Der Vorschaubereich unterstützt keine eingebetteten Schriftarten, wenn der Name der Schriftart, der in der Schriftartdatei abgelegt ist, nicht dem Namen in der CSS @font-face-Regel entspricht. Sie können das Werkzeug „Buch überprüfen“ verwenden, um solche problematischen Schriftarten schnell zu finden und das Problem zu beheben.

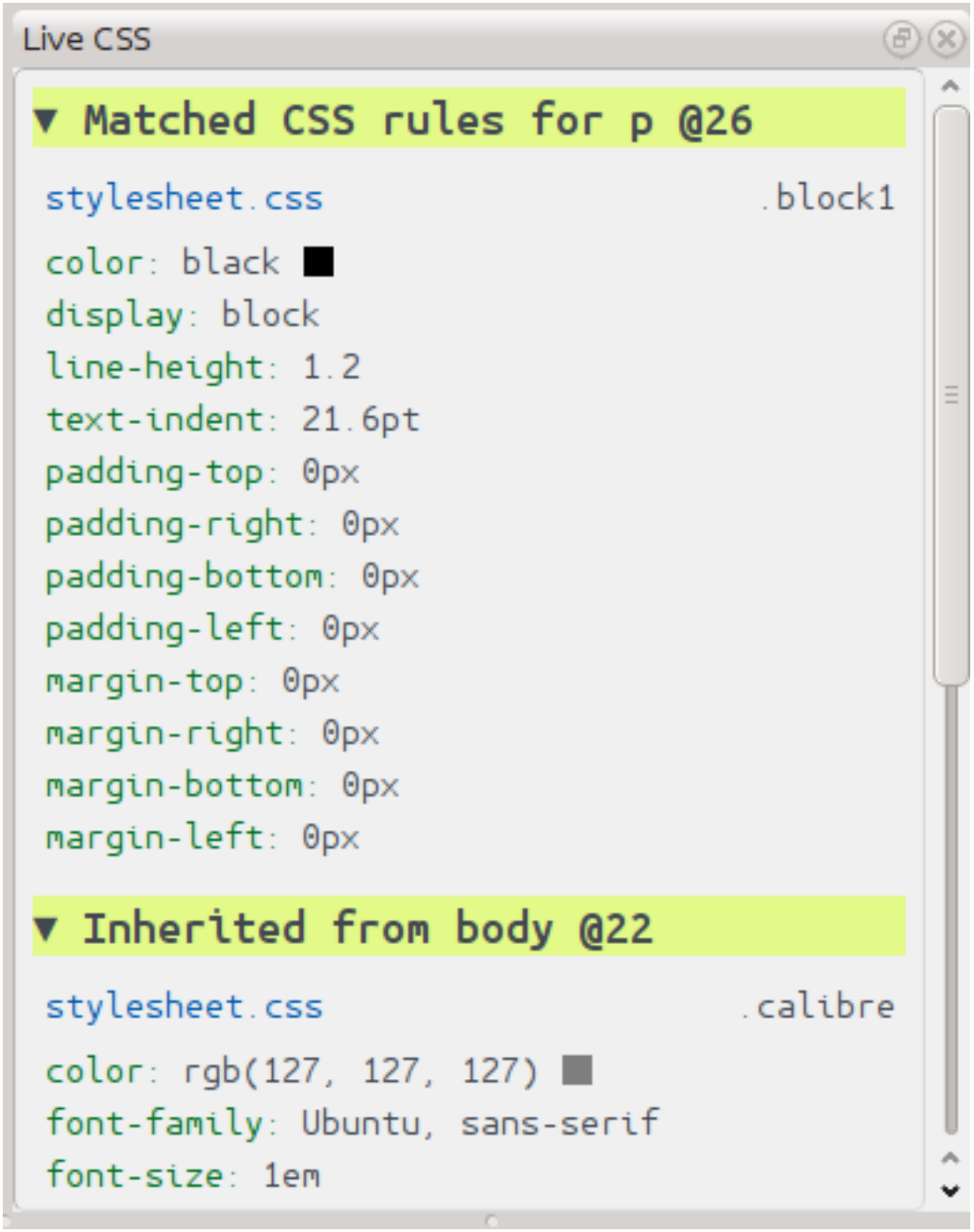
5.6.1 Aufteilen von HTML-Dateien

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.

Beim Teilen einer Datei werden automatisch alle Links und Verweise aktualisiert, die auf die untere Hälfte der Datei verwiesen haben. Außerdem wird die gerade geteilte Datei im Editor geöffnet.

Sie können auch eine einzelne HTML-Datei an mehreren Stellen automatisch aufteilen lassen, indem Sie innerhalb der Datei mit der rechten Maustaste in den Editor klicken und *An unterschiedlichen Stellen teilen* auswählen. Damit können Sie eine große Datei ganz einfach bei allen Markern für Überschriften oder allen Markern, die zu einer bestimmte Klasse gehören, aufteilen.

5.7 Der Echtzeit-CSS-Bereich



Der :guilabel:`Echtzeit-CSS-Bereich zeigt Ihnen alle style-Regeln an, die auf das von Ihnen gerade bearbeitete Tag angewendet werden. Der Name des Tags wird mit seiner Zeilennummer angezeigt, gefolgt von einer Liste der passenden style-Regeln.

Damit kann man schnell sehen, welche style-Regel für einen HTML-Tag gilt. Die Ansicht hat auch klickbare Verweise (in Blau), mit denen Sie direkt zur Position gelangen, an der der Stil definiert wird, um diese zu ändern. style-Regeln, die direkt zum Tag gehören und Regeln, die von übergeordneten Tags geerbt werden, werden ebenfalls angezeigt.

Der Bereich zeigt Ihnen auch, was die zuletzt berechneten styles für den Tag sind. Eigenschaften in der Liste, die durch Regeln mit höherer Priorität ersetzt wurden, werden durchgestrichen angezeigt.

Sie können den Echtzeit-CSS-Bereich über *Anzeigen → Echtzeit-CSS* aktivieren.

5.8 Verschiedene Werkzeuge

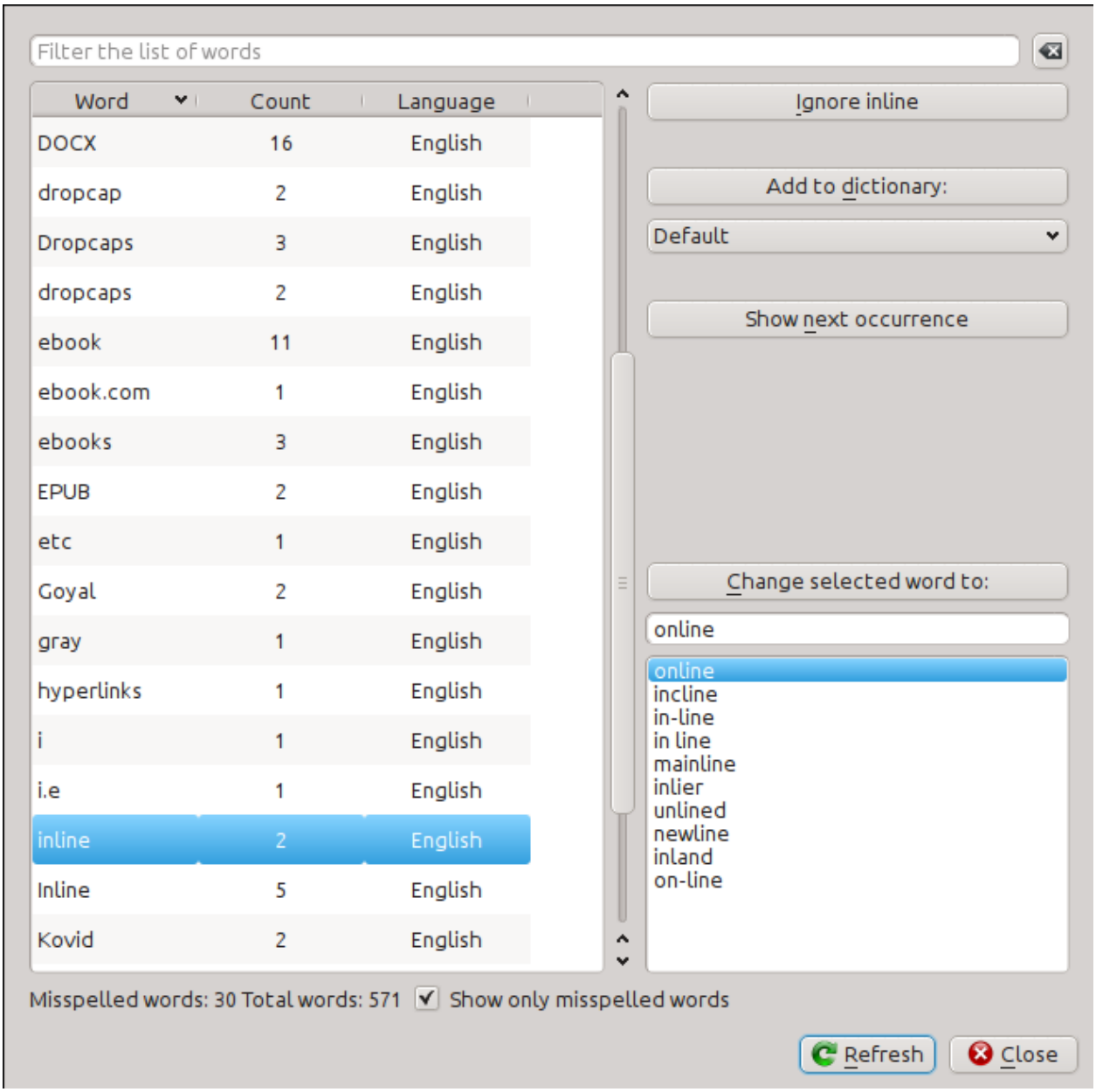
Es existieren noch einige weitere Werkzeuge, die für die Bearbeitung eines Buches hilfreich sein können.

5.8.1 Die Inhaltsverzeichnis-Ansicht

Die Inhaltsverzeichnis-Ansicht zeigt Ihnen das aktuelle Inhaltsverzeichnis des Buchs. Ein Doppelklick auf einen der Einträge öffnet den durch diesen Eintrag verlinkten Ort in einem Editor. Mit einem Rechtsklick kann das Inhaltsverzeichnis bearbeitet, aktualisiert oder seine Einträge aus-/eingeklappt werden. Diese Ansicht ist über *Anzeigen → Inhaltsverzeichnis* zugänglich.

5.8.2 Überprüfen der Rechtschreibung von Wörtern im Buch

Eine Rechtschreibprüfung können Sie über die Option *Werkzeuge → Überprüfung der Rechtschreibung* durchführen.



Wörter werden zusammen mit deren Häufigkeit im Buch und der Sprache des Wortes angezeigt. Die Sprache wird aus den Metadaten des Buches und den `lang`-Attributen in den HTML-Dateien ermittelt. Damit arbeitet die Rechtschreibprüfung problemlos auch mit solchen Büchern arbeiten zusammen, die Texte in mehreren Sprachen enthalten. Zum Beispiel wird im folgenden HTML-Auszug das Wort „color“ nach amerikanischer und das Wort „colour“ nach britischer Rechtschreibung überprüft:

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

Bemerkung

Sie können ein Wort doppelklicken um das nächste Vorkommen dieses Wortes im Editor hervorzuheben. Das ist nützlich, um ein Wort manuell zu bearbeiten, oder um es in seinem Kontext zu betrachten.

Um ein Wort zu ändern, einfach mit einem Doppelklick auf eine der vorgeschlagenen Alternativen auf der rechten Seite klicken oder einen eigenen Vorschlag eingeben und auf die Schaltfläche *Ausgewähltes Wort ändern zu* klicken. Dies ersetzt jedes auftretende Wort im gesamten Buch. Mit einem Rechtsklick auf ein Wort in der Hauptwortliste kann man auch bequem das Wort vom Rechtsklickmenü aus ändern.

Mit einem Klick auf die Schaltfläche *Ignore* kann die Rechtschreibkorrektur ein Wort ignorieren. Außerdem kann mithilfe der Schaltfläche *Add to dictionary* ein Wort zum Benutzerwörterbuch hinzugefügt werden. Da die Rechtschreibkorrektur auf mehrere Benutzerwörterbücher zurückgreifen kann, können Sie das Wörterbuch, zu dem das Wort hinzugefügt werden soll, ausgewählt werden.

Sie können mit der Rechtschreibprüfung auch alle Wörter Ihres Buchs anzeigen lassen, nicht nur die falsch geschriebenen. Das ist nützlich, wenn Sie sehen möchten, welche Wörter am häufigsten verwendet werden oder wenn Sie bestimmte Wörter einfach suchen und ersetzen möchten.

Bemerkung

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.

Bemerkung

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 -->
```

Hinzufügen von neuen Wörterbüchern

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 .oxt format). You can download these dictionaries from The LibreOffice Extensions repository[36].

5.8.3 Einfügen von Sonderzeichen

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.

Da es viele Zeichen gibt, können Sie ihre eigene :guilabel:‘Favoriten‘ Zeichen definieren, die dann an erster Stelle angezeigt werden. Dazu klicken Sie mit der rechten Maustaste auf ein Zeichen, das Sie zu den Favoriten hinzufügen möchten. Durch Klicken mit der rechten Maustaste auf ein Zeichen in den Favoriten, können Sie es wieder aus den Favoriten entfernen. Zum Neu-ordnen der Zeichen in den Favoriten klicken Sie auf die Schaltfläche *Favoriten Neu-ordnen* und ziehen Sie die Zeichen an die gewünschte Stelle.

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.

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

Schliesslich können spezielle Zeichen auch durch HTML-Entitäten eingegeben werden.   wird zum Beispiel zu einem geschützen Leerschlag. Zum Ersetzen muss das Semikolon zwingend eingegeben werden.

5.8.4 Die Codeinspektor-Ansicht

Diese Ansicht zeigt den HTML-Code und das darauf angewandte CSS. Mit einem Rechtsklick auf eine Stelle im Vorschaubereich und der Auswahl *Element Untersuchen* können Sie den Inspektor öffnen. So können Sie den HTML-Code des Elements sehen und – viel wichtiger – die CSS Regeln, die darauf angewandt werden. Sie können sogar dynamisch die Regeln verändern und sofort die jeweilige Änderung sehen. Beachten Sie, dass das Ändern des Styles nicht den Buchinhalt verändert. Es ist nur für schnelles Ausprobieren. Die Möglichkeit etwas live im Inspektor zu bearbeiten ist in Entwicklung

5.8.5 Überprüfen externer Verweise

Dieses Hilfsmittel wird zur Überprüfung aller Verknüpfungen mit externen Webseiten verwendet. Das Werkzeug wird versuchen, sämtliche extern verknüpften Webseiten zu erreichen und sämtliche fehlerhaften Verknüpfungen zur Korrektur in einem praktischen Format anzeigen.

5.8.6 Externe Ressourcen werden heruntergeladen

Sie können dieses Werkzeug verwenden, um automatisch alle Bilder, Stylesheets etc. herunterzuladen, welche nicht mit dem Buch gebündelt sind (d. h. wenn diese per URL auf einen Ort im Internet verweisen). Das Werkzeug wird alle dieser Ressourcen finden und sie automatisch herunterladen, sie dem Buch hinzufügen und alle Referenzen auf sie durch Referenzen auf die heruntergeladenen Dateien ersetzen.

5.8.7 Ordnen von Dateien in Ordner nach Typ

Wenn Sie EPUB-Dateien bearbeiten, die Sie von irgendwo bekommen haben, werden Sie oft sehen, dass die Dateien willkürlich in verschiedene Unterordner sortiert sind. Dieses Tool ermöglicht Ihnen die Dateien automatisch aufgrund ihrer Typen in Unterordner zu sortieren. Zugriff darauf via *Werkzeuge → In Ordner sortieren*. Beachten Sie, dass dieses Werkzeug nur ändert, wie die Dateien in EPUB sortiert sind, es verändert nicht, wie diese im Dateibrowser angezeigt werden.

5.8.8 Importieren von Dateien in anderen eBook-Formaten als EPUB

Der Editor hat die Fähigkeit, Dateien in einigen anderen eBook Formaten direkt als neue EPUB zu importieren, ohne eine komplette Konvertierung zu durchlaufen. Das ist besonders nützlich, wenn sie EPUB Dateien direkt aus Ihren, von Hand bearbeiteten HTML Dateien erstellen möchten. Dazu nutzen Sie die Funktion *Datei → HTML- oder DOCX-Datei als neues Buch importieren*.

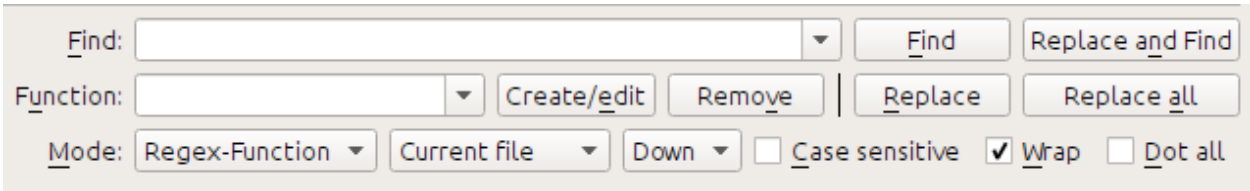
Function mode for Search & replace in the Editor

The *Search & replace* tool in the editor support a *function mode*. In this mode, you can combine regular expressions (see *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214)) with arbitrarily powerful Python functions to do all sorts of advanced text processing.

In the standard *regexp* 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.

Techniken und Syntax des Funktions Modus werden mithilfe von Beispielen beschrieben, welche zeigen wie zunehmend schwierigere Aufgaben erledigt werden.

[37] https://docs.python.org



Automatisches Anpassen der Groß/Kleinschreibung der Dokumentüberschriften

Hier verwenden wir eine der eingebauten Funktionen des Editors um automatisch die Groß/Kleinschreibung aller Texte innerhalb eines Überschriften Tags in Titel Schreibweise zu ändern:

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

For the function, simply choose the *Title-case text (ignore tags)* builtin function. The will change titles that look like: `<h1>some titLE</h1>` to `<h1>Some Title</h1>`. It will work even if there are other HTML tags inside the heading tags.

Die erste benutzerdefinierte Funktion - Trennstriche typografisieren

Die echten Macht der Funktions Modus kommt von der Möglichkeit eigene Funktionen zu schreiben, die Text willkürlich anpassen. Das Satzzeichen typografisieren Werkzeug im Editor ignoriert einzelne Trennstriche, mit dieser Funktion können diese durch Gedankenstriche ersetzen.

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 in available here[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.

Verwendet man diese Funktion um den folgenden Regulären Ausdruck zu suchen:

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

Wird sie alle Trennstriche mit Gedankenstrichen ersetzen. Allerdings nur im tatsächlichen Text und nicht innerhalb von HTML Tags.

Die Macht des Funktions Modus - Mit einem Rechtschreibbuch schlecht getrennte Wörter korrigieren

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
```

(Fortsetzung auf der nächsten Seite)

[38] https://docs.python.org/library/re.html#match-objects

(Fortsetzung der vorherigen Seite)

```
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 &amp;
    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
```

Diese Funktion wird mit dem selben Regulären Ausdruck verwendet wie die vorherige und zwar:

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

Damit werden wie durch Magie alle inkorrekt getrennten Wörter im Buchtext korrigiert. Diese Funktion bedients sich hauptsächlich einem der zusätzlichen Argumente der replace Funktion und zwar `dictionaries`. Dieses Argument verweißt auf die Wörterbücher die vom Editor selbst verwendet werden um die Rechtschreibkontrolle des Buches durchzuführen. Was diese Funktion mach ist sich alle Wörter die durch einen Bindestrich getrennt sind zu nehmen, den Bindestrich zu entfernen und zu überprüfen ob das Wort ohne Bindestrich im Wörterbuch steht. Wenn ja werden die ürsprünglichen Wörter durch das zusammengesetzte Wort ohne Bindestrich ersetzt.

Zu Beachten ist, dass diese Technik nur bei einsprachigen Büchern funktioniert, da `dictionaries.recognized()` im Normalfall nur die Hauptsprache des Buches verwendet.

Automatisches Nummerieren von Abschnitten

Now we will see something a little different. Suppose your HTML file has many sections, each with a heading in an `<h2>` tag that looks like `<h2>Some text</h2>`. You can create a custom function that will automatically number these headings with consecutive section numbers, so that they look like `<h2>1. Some text</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'
```

Verwendung mit dem Suchausdruck:

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

Positioniere die Schreibmarke am Anfang der Datei und klicke *Alles ersetzen*.

This function uses another of the useful extra arguments to `replace()`: the `number` argument. When doing a *Replace*

All number is automatically incremented for every successive match.

Another new feature is the use of `replace.file_order` – setting that to `'spine'` means that if this search is run on multiple HTML files, the files are processed in the order in which they appear in the book. See *Wenn auf mehrere HTML-Dateien anzuwenden, wähle die Dateireihenfolge* (die Titelseite 105) for details.

Inhaltsverzeichnis automatisch erstellen

Finally, lets try something a little more ambitious. Suppose your book has headings in `h1` and `h2` tags that look like `<h1 id="someid">Some Text</h1>`. We will auto-generate an HTML Table of Contents based on these headings. Create the custom function below:

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

Und Verwendung mit dem Suchausdruck:

```
<(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.

Another new feature is the use of `call_after_last_match` – setting that to `True` on the `replace()` function means that the editor will call `replace()` one extra time after all matches have been found. For this extra call, the match object will be `None`.

This was just a demonstration to show you the power of function mode, if you really needed to generate a Table of Contents from headings in your book, you would be better off using the dedicated Table of Contents tool in *Tools → Table of Contents*.

Die API für den Funktionsmodus

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
```

When a find/replace is run, for every match that is found, the `replace()` function will be called, it must return the replacement string for that match. If no replacements are to be done, it should return `match.group()` which is the original string. The various arguments to the `replace()` function are documented below.

Das `match`-Argument

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.

Das `number`-Argument

The `number` argument is the number of the current match. When you run *Replace All*, every successive match will cause `replace()` to be called with an increasing number. The first match has number 1.

Das Argument `file_name`

This is the filename of the file in which the current match was found. When searching inside marked text, the `file_name` is empty. The `file_name` is in canonical form, a path relative to the root of the book, using `/` as the path separator.

Das Argument `metadata`

This represents the metadata of the current book, such as title, authors, language, etc. It is an object of class *`calibre.ebooks.metadata.book.base.Metadata`* (die Titelseite 210). Useful attributes include, `title`, `authors` (a list of authors) and `language` (the language code).

[39] https://docs.python.org/library/re.html#match-objects

Das 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.

Das 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.

Das Argument `functions`

The `functions` argument gives you access to all other user defined functions. This is useful for code re-use. You can define utility functions in one place and re-use them in all your other functions. For example, suppose you create a function name `My Function` like this:

```
def utility():
    # do something

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

Then, in another function, you can access the `utility()` function like this:

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

You can also use the functions object to store persistent data, that can be re-used by other functions. For example, you could have one function that when run with *Replace All* collects some data and another function that uses it when it is run afterwards. Consider the following two functions:

```
# 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']
    ...
```

Fehlersuche in deinen Funktionen

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.

Wenn auf mehrere HTML-Dateien anzuwenden, wähle die Dateireihenfolge

When you run a *Replace all* on multiple HTML files, the order in which the files are processes depends on what files you have open for editing. You can force the search to process files in the order in which the 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` accepts two values, `spine` and `spine-reverse` which cause the search to process multiple files in the order they appear in the book, either forwards or backwards, respectively.

Die Funktion nach dem letzten Fund ein zusätzliches Mal aufrufen

Sometimes, as in the auto generate table of contents example above, it is useful to have your function called an extra time after the last match is found. You can do this by setting the `call_after_last_match` attribute on your function, like this:

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

replace.call_after_last_match = True
```

Die Ausgabe der Funktion an den markierten Text anhängen

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
```

Beim Durchsuchen von markiertem Text das Ergebnisfenster unterdrücken

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
```

More examples

More useful examples, contributed by calibre users, can be found in the calibre E-book editor forum[40].

Snippets

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. For example, suppose you often find yourself inserting link tags when editing HTML files, then you can simply type `<a` in the editor and press `Control+J`. The editor will expand it to:

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

Des Weiteren wird das Wort `filename` ausgewählt und der Cursor darüber plaziert. Auf diese Weise ist es einfach den echten Dateinamen mit Hilfe der nützlichen *Automatisches Vervollständigen* (die Titelseite 111) Funktion einzugeben. Nach dem der Name einmal eingegeben wurde springt der Cursor durch die `Control+J` Taste zwischen die `<a>` Tags damit der Text für den Link einfach eingeben werden kann.

Das Snippet System des Editors ist sehr fortschrittlich, es beinhaltet einige vorinstallierte Snippets und es ist möglich eigene Snippets zu erstellen, welche zu dem eigenen Bearbeitungsstil passen.

Die nachfolgende Beschreibung der vorgefertigten Snippets sollten dabei helfen, die Möglichkeiten des Snippet-Systems aufzuzeigen.

Bemerkung

You can also use snippets in the text entry fields in the *Search & replace* panel, however, placeholders (using `Control+J` to jump around) will not work.

Die vorgefertigten Snippets

Die vorgefertigten Snippets werden unten beschrieben. Sie können sie überschreiben, indem Sie Ihr eigenes Snippet mit dem gleichen Auslösetext erstellen.

Fülltext einfügen [Lorem]

The first built-in snippet, and the simplest is used to insert filler text into a document. The filler text is taken from De finibus bonorum et malorum[41] a philosophical work by Cicero (translated to English). To use it simply type `Lorem` in an HTML file and press `Control+J`. It will be replaced by a couple of paragraphs of filler.

The definition of this snippet is very simple, the trigger text is defined as `Lorem` and the template is defined simply as the literal text to be inserted. You can easily customize it to use your favorite form of filler text.

[40] https://www.mobileread.com/forums/showthread.php?t=237181
[41] https://en.wikipedia.org/wiki/De_finibus_bonorum_et_malorum

Einen selbstschließenden HTML Tag [<>] einfügen

Now let's look at a simple example of the powerful concept of *placeholders*. Say you want to insert the self-closing tag `<hr/>`. Just type `<>`, and press `Control+J`, the editor will expand the snippet to:

```
<|/>
```

Here, the `|` symbol represents the current cursor position. You can then type `hr` and press `Control+J` to move the cursor to after the end of the tag. This snippet is defined as:

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

Placeholders are simply the dollar ($) sign followed by a number. When the snippet is expanded by pressing `Control+J` the cursor is positioned at the first placeholder (the placeholder with the lowest number). When you press `Control+J` again the cursor jumps to the next placeholder (the placeholder with the next higher number).

Einen HTML Link-Tag [<a] einfügen

HTML link tags all share a common structure. They have an `href` attribute and some text between the opening and closing tags. A snippet to make typing them more efficient will introduce us to some more features of placeholders. To use this snippet, simply type `<a` and press `Control+J`. The editor will expand this to:

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

Not only that, the word `filename` will be selected, with the cursor placed over it, so that you can easily type in the real filename, using the editor's nifty *Automatisches Vervollständigen* (die Titelseite 111) feature. And once you are done typing the filename, press `Control+J` again and the cursor will jump to the position in between the `<a>` tags so you can easily type in the text for the link. After you are done typing the text, press `Control+J` again to jump to the point after the closing tag. This snippet is defined as:

```
Trigger: <a
Template: <a href="${1:filename}">${2*}</a>$3
```

There are a couple of new features here. First the `$1` placeholder has become more complex. It now includes some *default text* (the word `filename`). If a placeholder contains default text, the default text is substituted for the placeholder when the snippet is expanded. Also when you jump to a placeholder with default text using `Control+J`, the default text is selected. In this way, you can use default text to act as a reminder to you to fill in important parts of the template. You can specify default text for a placeholder by using the syntax: `${<number>:default text}`.

The other new feature is that the second placeholder has an asterisk after it (`${2*}`). This means that any text that was selected before expanding the template is substituted for the placeholder. To see this in action, select some text in the editor, press `Control+J`, type `<a` and press `Control+J` again, the template will be expanded to:

```
<a href="filename">whatever text you selected</a>
```

Einen HTML Bild-Tag einfügen [<i]

This is very similar to inserting an HTML link, as we saw above. It allows you to quickly input an `<img src="filename" alt="description" />` tag and jump between the `src` and `alt` attributes:

```
Trigger: <i
Template: <img src="${1:filename}" alt="${2*:description}" />$3
```

Insert an arbitrary HTML tag [<<]

This allows you to insert an arbitrary full HTML tag (or wrap previously selected text in the tag). To use it, simply type `<<` and press `Control+J`. The editor will expand it to:

```
<|></>
```

Type the tag name, for example: `span` and press `Control+J`, that will result in:

```
<span>|</span>
```

You will note that the closing tag has been automatically filled with `span`. This is achieved with yet another feature of placeholders, *mirroring*. Mirroring simply means that if you specify the sample placeholder more than once in a template, the second and all later positions will be automatically filled in with whatever you type in the first position, when you press `Control+J`. The definition for this snippet is:

```
Trigger: <<
Template: <$1>${2*}</$1>$3
```

As you can see, the first placeholder ($1) has been specified twice, the second time in the closing tag, which will simply copy whatever you type in the opening tag.

Inserting an arbitrary HTML tag with a class attribute [<c]

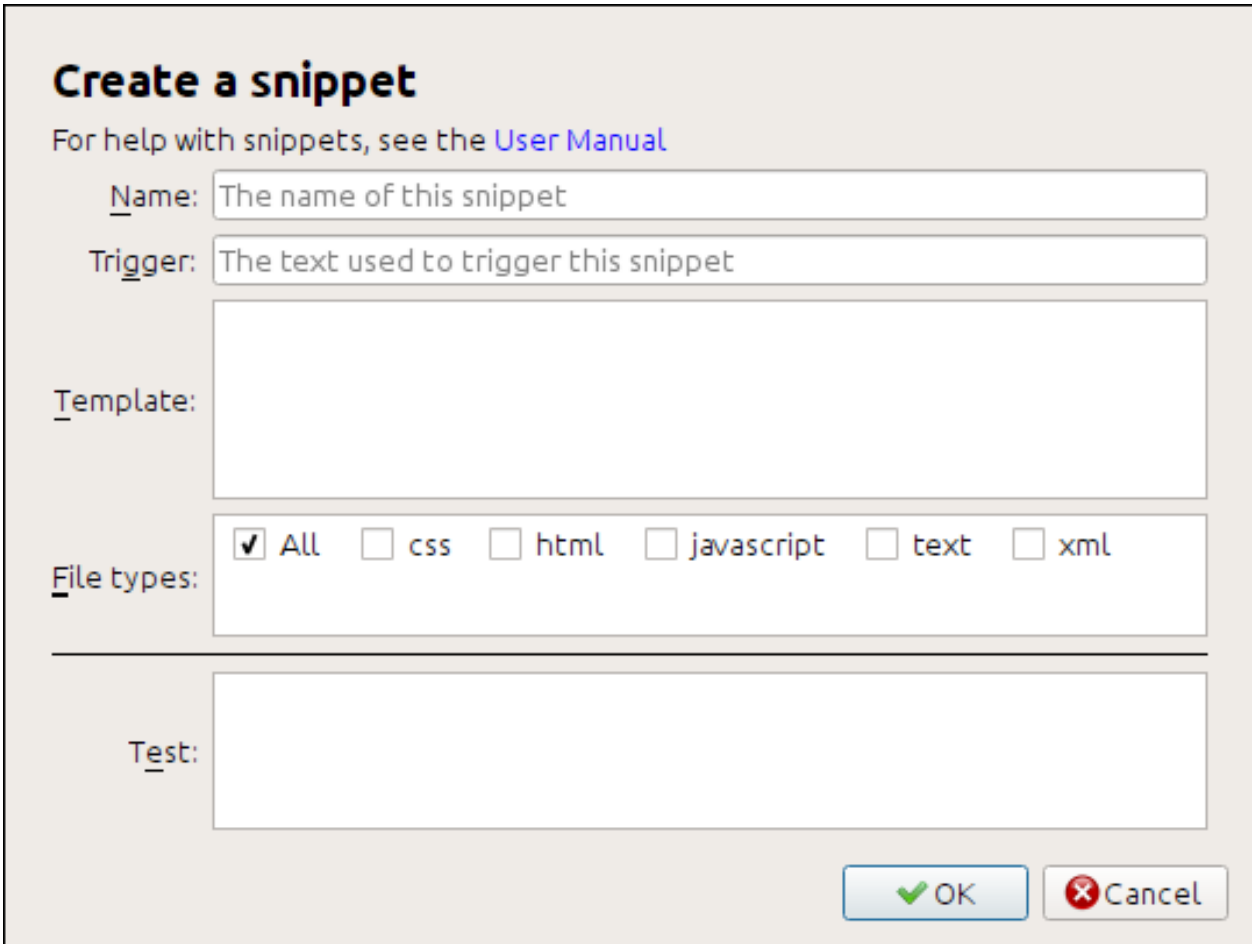
This is very similar to the insert arbitrary tag example above, except that it assumes that you want to specify a class for the tag:

```
Trigger: <c
Template: <$1 class="${2:classname}">${3*}</$1>$4
```

This will allow you to first type the tag name, press `Control+J`, type the class name, press `Control+J` type the contents of the tag and press `Control+J` one last time to jump out of the tag. The closing tag will be auto-filled.

Erstellen eigener Snippets

Snippets really shine because you can create your own to suit your editing style. To create your own snippets go to *Edit → Preferences → Editor settings → Manage snippets* in the editor. This will pop-up an easy to use dialog to help you create your own snippets. Simply click the *Add snippet* button and you will see a dialog that looks like:



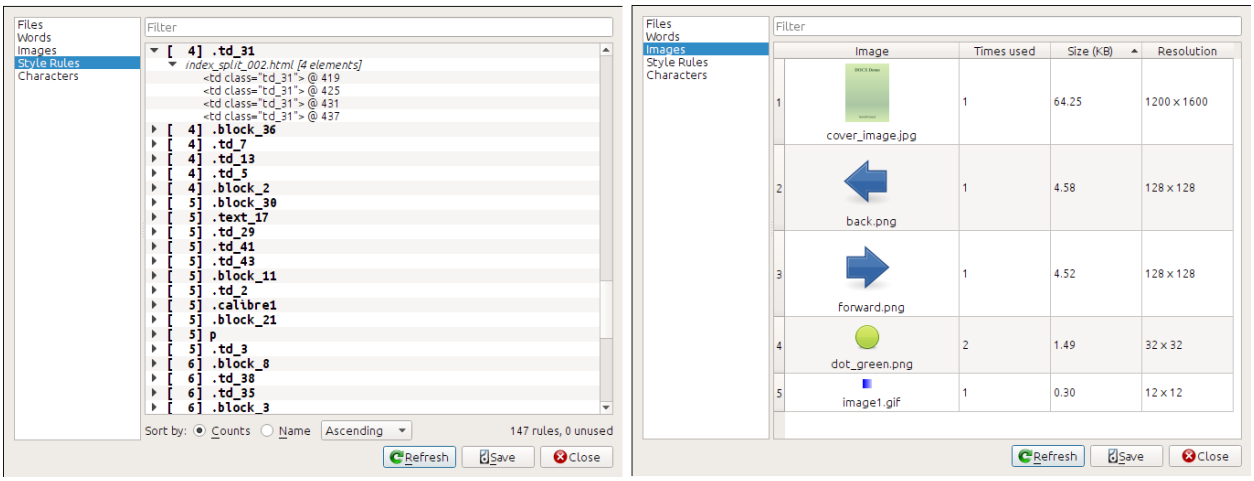
First give your snippet a name, something descriptive, to help identify the snippet in the future. Then specify the *trigger*. A trigger is simply the text that you have to type in the editor before pressing `Control+J` in order to expand the snippet.

Then specify the snippet template. You should start with one of the examples above and modify it to suit your needs. Finally, specify which file types you want the snippet to be active for. This way you can have multiple snippets with the same trigger text that work differently in different file types.

The next step is to test your newly created snippet. Use the *Test* box at the bottom. Type in the trigger text and press `Control+J` to expand the snippet and jump between placeholders.

5.8.9 Das Berichte-Werkzeug

Der Editor enthält ein nettes Werkzeug namens *Berichte* (via *Werkzeuge → Berichte*). Dieses zeigt eine Zusammenfassung von allen Dateien, Bildern, Verweisen, Wörtern, Buchstaben und Stilregeln an, die im Buch genutzt werden. Jede Zeile im Bericht ist verknüpft. Per Doppelklick springt der Editor zu der Stelle im Buch, wo das gewählte Element benutzt bzw. definiert wird (je nachdem). Zum Beispiel, im *Verweise*, können Sie auf *Quelle* doppelklicken und springen zu der Stelle, wo der Link definert wird. Mit einem Doppelklick auf einen Eintrag unter *Ziele* springen sie zu dem Ort, auf den der Link verweist.



5.9 Besondere Funktionen des Code-Editors

Der HTML-Editor in Calibre ist sehr mächtig. Er verfügt über viele Funktionen die das Bearbeiten von HTML (und CSS) leichter machen.

5.9.1 Syntaxhervorhebung

Der HTML-Editor verfügt über fortgeschrittenes Syntaxhervorhebung. Seine Funktionen beinhalten:

- Text zwischen fett, kursiv und Titel tags wird fett/kursiv dargestellt.
- 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`.
- Ungültiges HTML wird hervorgehoben indem es rot unterstrichen wird
- 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.
- In `<style>`-Tags eingebettetes CSS wird hervorgehoben
- Spezielle, schwierig zu unterscheidende Zeichen wie geschützte Leerzeichen, verschiedene Bindestriche, etc. werden hervorgehoben.
- Verweise zu anderen Dateien in `<a>`-Tags, `<img>` und ``<link>`-Tags haben alle hervorgehobene Dateinamen. Falls der referenzierte Dateiname nicht existiert, ist der Dateiname rot unterstrichen.

5.9.2 Kontextspezifische Hilfe

Sie können auf den Namen eines HTML-Tags oder einer CSS-Eigenschaft rechtsklicken, um Hilfe für diesen Tag bzw. diese Eigenschaft zu erhalten.

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 Automatisches Vervollständigen

Eine der mühsamsten Aufgaben beim Bearbeiten von eBooks ist das Erstellen von Verweisen auf andere Dateien des Buches, auf CSS-Stylesheets oder Bilder, da der korrekte Dateiname und der relative Pfad zur Datei bekannt sein müssen. Die Auto-Vervollständigung des Editors vereinfacht diese Aufgabe.

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 Snippets

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 *Snippets* (die Titelseite 106).

KAPITEL 6

Der Calibre Inhalteserver

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.

Inhalt

- *Von anderen Geräten auf den Inhalteserver zugreifen* (die Titelseite 114)
  - *Accessing the server from devices on your home network* (die Titelseite 114)
  - *Von überall aus dem Internet auf den Inhalteserver zugreifen* (die Titelseite 115)
- *The server interface* (die Titelseite 115)
  - *Die Bücherliste* (die Titelseite 115)
  - *The book viewer* (die Titelseite 116)
- *Browserunterstützung* (die Titelseite 116)
- *Enabling offline support* (die Titelseite 116)
- *Managing user accounts from the command-line only* (die Titelseite 116)
- *Integrating the calibre Content server into other servers* (die Titelseite 117)
  - *Using a full virtual host* (die Titelseite 117)
  - *Using a URL prefix* (die Titelseite 117)
- *Creating a service for the calibre server on a modern Linux system* (die Titelseite 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.

Bemerkung

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 Von anderen Geräten auf den Inhalteserver zugreifen

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 workaround 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 Von überall aus dem Internet auf den Inhalteserver zugreifen

Warnung

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.

Bemerkung

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.

6.2.1 Die Bücherliste

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.

[42] https://www.mobileread.com/forums/forumdisplay.php?f=166
[43] https://www.whatismyip.com/
[44] https://www.noip.com/free

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 viewer

You can read any book in your calibre library by simply tapping on it and then tapping the *Read* button. The book viewer 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 Browserunterstützung

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.

Bemerkung

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;
    }
}
```

Oder, für Apache:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so

<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`.

Bei nginx lautet die notwendige Konfiguration:

```
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.

Bemerkung

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.

Bemerkung

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[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
```

Bemerkung

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.

Bemerkung

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/

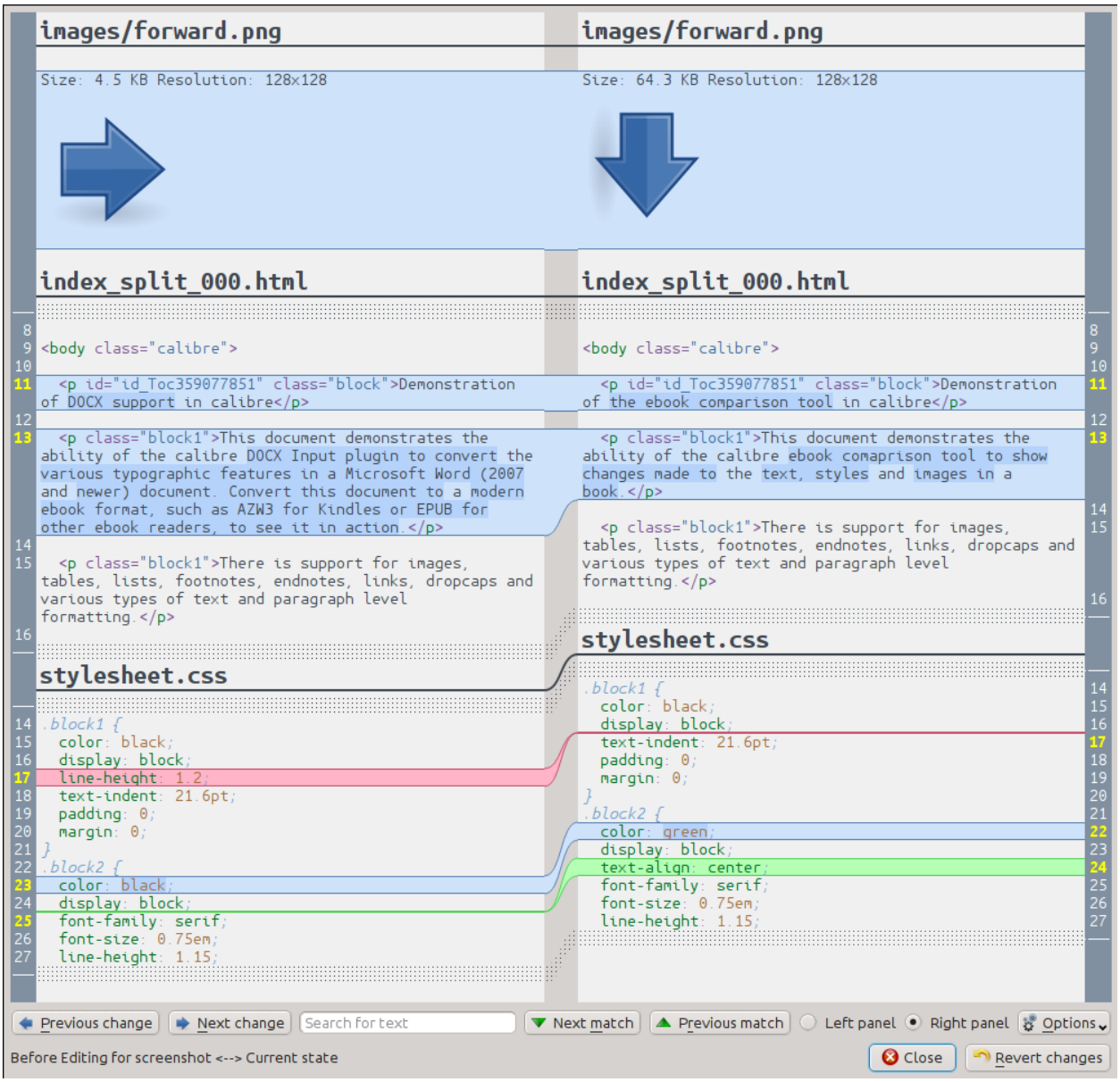
KAPITEL 7

Vergleichen von eBooks

Calibre enthält ein integriertes Werkzeug zum eBook-Vergleich und wird verwendet, um Veränderungen innerhalb des eBooks nach einer Bearbeitung oder Konvertierung darzustellen. Das Werkzeug vergleicht Bücher im EPUB- und AZW3-Format.

Um es zu verwenden, öffnen Sie entweder das eBook im Werkzeug zum *Bearbeiten von eBooks* (die Titelseite 79) und klicken dann auf *Datei → Mit einem anderen Buch vergleichen* oder verwenden Sie den *Über dieses Buch* (die Titelseite 20)-Bereich. Falls Sie eine Konvertierung von EPUB zu EPUB durchführen, wird die ursprüngliche EPUB-Datei als ORIGINAL_EPUB gesichert. Rechtsklicken Sie einfach auf den Eintrag ORIGINAL_EPUB im Buchdetailsbereich und wäheln Sie *Mit EPUB-Format vergleichen*.

Das sich öffnende Vergleichswerkzeug wird ähnlich der Abbildung unten aussehen. Es zeigt Unterschiede im Text, Formatvorlagen und Bildern in den ausgewählten Büchern.



7.1 Verstehen der Vergleichsansicht

Wie im Screenshot oben zu sehen, zeigt der Vergleichsansicht die Unterschiede zwischen den beiden Büchern nebeneinander. Es werden nur die Unterschiede mit einigen Zeilen um den Kontext angezeigt. Dies erleichtert es auf einen Blick zu erkennen, was in einem großen Dokument, wie beispielsweise in einem Buch, geändert wurde.

Hinzugefügter Text wird mit einem grünen Hintergrund, entfernt Text mit rotem Hintergrund und geänderter Text mit blauem Hintergrund angezeigt.

Die Zeilennummern aller geänderten Textstellen werden an den Seiten angezeigt, sodass es einfach ist, zu einer bestimmten Änderung im Editor zu gelangen. Falls Sie das Vergleichswerkzeug aus dem Editor heraus öffnen, können Sie auch auf eine Zeile im rechten Bereich doppelklicken, um im Editor automatisch zu dieser Zeile zu gelangen.

Es ist ein nützliches Verfahren beim Vergleich von Büchern, das Vergleichswerkzeug vor der eigentlichen Berechnung der Unterschiede zu veranlassen, alle Dateien im Text und Stil zu verschönern. Danach ist der besser strukturierte Text für die Unterschiede einfacher zu verfolgen. Klicken Sie hierzu auf die Schaltfläche *Optionen* in der unteren Ecke und

wählen Sie dort *Dateien vor dem Vergleichen verschönern* aus. Beachten Sie, dass verschönern manchmal unerwünschte Effekte haben kann. Beispielsweise kann es bei gültigen Markups dazu führen, dass diese zu ungültigen Markups geändert werden. Die Anzahl der angezeigten Zeilen für die Unterschiede im Kontext können Sie ebenfalls über die Schaltfläche *Optionen* ändern.

Sie können über die Suchleiste unten nach jedem Text innerhalb der Unterschiede suchen. Sie müssen für die Suche angeben, in welchem Fensterbereich gesucht werden soll: *Linker Bereich* oder *Rechter Bereich*.

7.2 Starten des Vergleichswerkzeugs

Das Vergleichswerkzeug ist besonders hilfreich, wenn Sie zwei Versionen des gleichen Buches haben und der Unterschied zwischen diesen gefunden werden soll. Zu diesem Zweck gibt es mehrere Möglichkeiten, um das Werkzeug zu starten.

7.2.1 Vergleichen zweier eBook-Dateien

Öffnen Sie die erste Datei mit dem *Bearbeiten von eBooks* (die Titelseite 79). Klicken Sie auf *Datei → Mit einem anderen Buch vergleichen* und wählen dann die zweite Datei aus (Diese muss im gleichen Format wie die erste vorliegen). Die Vergleichsansicht öffnet die zur Zeit bearbeitende Datei des Editors in der rechten Fensterseite und die zweite Datei in der linken Fensterseite.

7.2.2 Vergleichen des ORIGINAL_FMT mit FMT

Wenn Sie in Calibre eine Konvertierung ins gleiche Format FMT vornehmen, wird die ursprüngliche Datei als ORIGINAL_FMT gespeichert. FMT steht hierbei stellvertretend als Platzhalter für das Format, wie z. B. für EPUB. Die Veränderung bei einer Konvertierung im Buch können Sie durch Rechtsklicken auf den Eintrag ORIGINAL_FMT im *Über dieses Buch* (die Titelseite 20)-Bereich des Calibre-Hauptfensters und anschließender Auswahl von *Mit dem FMT Format vergleichen* sehen. In der Vergleichsansicht wird das ORIGINAL_FMT auf der linken und FMT auf der rechten Seite geöffnet.

7.2.3 Vergleichen eines Sicherungspunkts mit dem aktuellen Stand des Buches während des Bearbeitens

Der *Bearbeiten von eBooks* (die Titelseite 79) hat ein sehr nützliches Feature, genannt *Sicherungspunkte* (die Titelseite 91). Hiermit können Sie den aktuellen Zustand des Buches als einen benannten *Sicherungspunkt* speichern. Durch Auswahl des *Sicherungspunkts* können Sie wieder zum Zustand vor der Änderung zurückkehren, falls Ihnen die Änderung nach Erstellung des Sicherungspunkts nicht gefallen. Sicherungspunkte werden auch automatisch bei der Durchführung verschiedene automatisierte Aktionen im Editor erstellt. Sie können die Liste der Sicherungspunkte über *Anzeigen → Sicherungspunkte* anzeigen lassen und dann die Schaltfläche *Vergleichen* verwenden, um das Buch beim ausgewählten Sicherungspunkt mit dem aktuellen Zustand zu vergleichen. Das Vergleichswerkzeug zeigt den ausgewählten Sicherungspunkt auf der linken Fensterseite und den aktuellen Stand des Dokuments im Editor auf der rechten Fensterseite.

KAPITEL 8

Bearbeiten von eBook Metadaten

Inhalt

- *Bearbeiten der Metadaten einzelner Bücher* (die Titelseite 125)
  - *Herunterladen von Metadaten* (die Titelseite 126)
  - *Verwalten von Buchformaten* (die Titelseite 126)
  - *Alles über Titelbilder* (die Titelseite 126)
- *Bearbeiten der Metadaten mehrerer Bücher gleichzeitig* (die Titelseite 126)
  - *Suchen und Ersetzen* (die Titelseite 126)
  - *Gleichzeitiges Herunterladen von Metadaten* (die Titelseite 127)
- *Adding extra data files to a book* (die Titelseite 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 Bearbeiten der Metadaten einzelner Bücher

Klicken Sie entweder auf das zu bearbeitende Buch und danach auf die Schaltfläche *Metadaten bearbeiten* oder drücken Sie die Taste `E`. Es öffnet sich ein Dialog, in dem alle Aspekte der Metadaten bearbeitet werden können. Er hat verschiedene Funktionen, der die Bearbeitung schneller und effizienter macht. Eine Liste der häufig verwendeten Tipps:

- Klicken Sie auf die Schaltfläche links neben Titel, um Titel und Autoren automatisch zu tauschen.
- Sie können auf den blauen Pfeil neben dem Feld „Autorensortierung“ klicken, dann füllt es Calibre mit den gespeicherten Autorensortierwert für den jeweiligen Autor aus. Verwenden Sie den *Autoren verwalten* Dialog, um die Autorensortierwerte zu betrachten und zu ändern. Dieser Dialog können Sie öffnen, indem Sie lange auf den

blauen Pfeil klicken, bis sich ein Menü mit dem Eintrag „Autoren verwalten“ öffnet. Ebenso kann der Dialog über einen Klick auf die Schaltfläche neben „Autor(en)“ aufgerufen werden.

- 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.
- Das Feld „Sortierung nach Autor“ wird rot, wenn der Wert anders ist, als von Calibre angenommen.

8.1.1 Herunterladen von Metadaten

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.

Füllen Sie die Felder für Titel und Autor aus und klicken Sie auf die *Metadaten herunterladen* Schaltfläche. Calibre zeigt Ihnen dann eine Liste von Büchern an, die am ehesten mit dem Titel und Autor übereinstimmen. Wenn Sie das ID-Feld zuerst mit der ISBN ausfüllen, wird sie dem Titel und Autor bevorzugt. Wenn es keine Treffer gab, versuchen Sie die Suche ein wenig allgemeiner zu halten und geben Sie nur einige Schlagwörter im Titel und den Nachnamen beim Autor an.

8.1.2 Verwalten von Buchformaten

In Calibre können einem Buch verschiedene *Formate* zugeordnet werden. Zum Beispiel können Sie die gesammelten Werke Shakepeares im eBook-Format haben und es später ins MOBI-Format konvertieren um es auf dem Kindle zu lesen. Calibre fügt dem Buch automatisch das neue Format zu. Im Bereich „Verfügbare Format“ im Dialog „Medadaten bearbeiten“ können Sie diese Formate verwalten. Sie können neue Formate hinzufügen, bestehende löschen oder mit Calibre die Metadaten und das Titelbild für ein Buchformat anpassen.

8.1.3 Alles über Titelbilder

Sie können mit Calibre Titelbilder herunterladen, vorausgesetzt das Buch hat eine ISBN. Alternativ können Sie eine Datei von Ihrem Computer als Titelbild verwenden. Calibre kann auch ein Standardtitelbild mit den einfachen Metadaten erzeugen. Außerdem können Sie Bilder durch Ziehen und Ablegen im Titelbildfenster oder über Rechtsklick, kopieren/einfügen ändern.

Zusätzlich gibt es eine Schaltfläche, um automatisch hässliche Ränder von Titelbildern zu entfernen.

8.2 Bearbeiten der Metadaten mehrerer Bücher gleichzeitig

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 Suchen und Ersetzen

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 *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214).

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*.

Wenn das Suchfeld in dem Sie suchen, mehrere Felder enthält, dann wird jeder Tag seperat behandelt. Wenn Ihr Tags zum Beispiel sind *Horror, Scary*, der Suchausdruck *r*, wird nicht der gesamte Ausdruck gesucht, sondern zuerst *Horror* und dann *Scary*.

Um bei der Suche die Unterscheidung zwischen Groß- und Kleinbuchstaben zu ignorieren, deaktivieren Sie *Groß-/Kleinschreibung beachten*.

You can have calibre change the case of the result (information after the replace has happened) by choosing one of the functions from the *Apply function after replace* box. The operations available are:

- *Kleinbuchstaben* – Ändern aller Zeichen des Feldes in Kleinbuchstaben
- *Großbuchstaben* – Ändern aller Zeichen des Feldes in Großbuchstaben
- *Überschriftenschreibweise* – Große Anfangsbuchstaben bei allen Wörtern außer Präpositionen, Artikeln und Konjunktionen. Funktioniert nur bei englischen Wörtern.

The *Your test* box is provided for you to enter text to check that search/replace is doing what you want. In the majority of cases the book test boxes will be sufficient, but it is possible that there is a case you want to check that isn't shown in these boxes. Enter that case into *Your test*.

Regular expression mode has some differences from character mode, beyond (of course) using regular expressions. The first is that functions are applied to the parts of the string matched by the search string, not the entire field. The second is that functions apply to the replacement string, not to the entire field.

The third and most important is that the replace string can make reference to parts of the search string by using backreferences. A backreference is `\\n` where n is an integer that refers to the n'th parenthesized group in the search expression. For example, given the same example as above, *a bad cat*, a search expression *a (…) (…)*, and a replace expression *a \2 \1*, the result will be *a cat bad*. Please see the *Alles über die Verwendung von regulären Ausdrücken in Calibre* (die Titelseite 214) for more information on backreferences.

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 *\1*, and the desired case change function should be selected.

Finally, in regular expression mode you can copy values from one field to another. Simply make the source and destination field different. The copy can replace the destination field, prepend to the field (add to the front), or append to the field (add at the end). The ‚use comma' checkbox tells calibre to (or not to) add a comma between the text and the destination field in prepend and append modes. If the destination is multiple (e.g., tags), then you cannot uncheck this box.

Suchen und Ersetzen wird durchgeführt, nachdem Änderungen an Metadaten in anderen Reitern angewendet wurden. Das kann zu Verwirrung führen, weil die Test-Felder die Informationen ohne Berücksichtigung anderer Änderungen anzeigt, da diese Funktion erst nach den anderen Änderungen angewandt wird. Wenn Sie unsicher sind, was passiert, vermischen Sie Suchen/Ersetzen nicht mit anderen Änderungen.

8.2.2 Gleichzeitiges Herunterladen von Metadaten

Um Metadaten für mehrere Bücher gleichzeitig herunterzuladen Mausklick rechts auf *Metadaten bearbeiten* und *Metadaten herunterladen* wählen. Metadaten und Titelbilder können getrennt oder zusammen heruntergeladen werden.

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.

KAPITEL 9

Häufig gestellte Fragen (FAQ)

Inhalt

- *eBook-Formatkonvertierung* (die Titelseite 129)
- *Geräteintegration* (die Titelseite 132)
- *Bibliotheken-Verwaltung* (die Titelseite 139)
- *Verschiedenes* (die Titelseite 144)

9.1 eBook-Formatkonvertierung

Inhalt

- *Welche Formate unterstützt Calibre bei der Konvertierung?* (die Titelseite 130)
- *Was ist das beste Quellformat zum Konvertieren?* (die Titelseite 130)
- *Ich habe eine PDF-Datei konvertiert, aber bei der Ausgabedatei gab es einige Probleme.* (die Titelseite 130)
- *Wie konvertiere ich meine Datei mit nicht-englischen Zeichen oder Anführungszeichen?* (die Titelseite 130)
- *Was hat es mit dem Inhaltsverzeichnis in MOBI-Dateien auf sich?* (die Titelseite 131)
- *Wie konvertiere ich eine Sammlung von HTML-Dateien in einer bestimmten Reihenfolge?* (die Titelseite 131)
- *Das mit Calibre erzeugte EPUB ist nicht gültig?* (die Titelseite 132)
- *Wie verwende ich einige der fortgeschrittenen Funktionen der Konvertierungswerkzeuge?* (die Titelseite 132)

9.1.1 Welche Formate unterstützt Calibre bei der Konvertierung?

Calibre unterstützt bei der Konvertierung viele Quell- und Zielformate. Es kann jedes Quellformat aus der folgenden Liste in die genannten Zielformate konvertieren.

Input Formats: AZW, AZW3, AZW4, CBZ, CBR, CB7, CBC, CHM, DJVU, DOCX, EPUB, FB2, FBZ, HTML, HTMLZ, KEPUB, LIT, LRF, MOBI, ODT, PDF, PRC, PDB, PML, RB, RTF, SNB, TCR, TXT, TXTZ

Output Formats: AZW3, EPUB, DOCX, FB2, HTMLZ, KEPUB, OEB, LIT, LRF, MOBI, PDB, PMLZ, RB, PDF, RTF, SNB, TCR, TXT, TXTZ, ZIP

Bemerkung

PRC ist ein generisches Format, Calibre unterstützt PRC Dateien mit TextRead und MOBIBook headern. PDB ist auch ein generisches Format. Calibre unterstützt eReader, Plucker (nur Eingabe), PML und zTxt PDB Dateien. DJVU Unterstützung ist nur zur Konvertierung von DJVU Dateien, die eingebetteten Text enthalten. Diese werden üblicherweise mit ORC Software erzeugt. MOBI Bücher können einer von zwei Typen sein MOBI6 und KF8. Calibre unterstützt beide vollständig. MOBI Dateien haben oft .azw oder .azw3 Dateiendungen. DOCX Dateien aus Microsoft Word 2007 und neuer werden unterstützt.

9.1.2 Was ist das beste Quellformat zum Konvertieren?

In order of decreasing preference: LIT, MOBI, AZW, EPUB, KEPUB, AZW3, FB2, FBZ, DOCX, HTML, PRC, ODT, RTF, PDB, TXT, PDF

9.1.3 Ich habe eine PDF-Datei konvertiert, aber bei der Ausgabedatei gab es einige Probleme.

PDF ist ein schreckliches Ausgangsformat für die Konvertierung. Eine Liste verschiedener Probleme, die beim Konvertieren von PDF-Dateien auftreten können, finden Sie unter *PDF-Dokumente konvertieren* (die Titelseite 75).

9.1.4 Wie konvertiere ich meine Datei mit nicht-englischen Zeichen oder Anführungszeichen?

Dieses Problem hat zwei Aspekte:

1. Die Enkodierung der Eingabedatei erfahren: Calibre versucht zu erraten, welche Zeichenenkodierung deine Eingabedateien benutzen, was aber oft unmöglich ist, weshalb Sie festlegen müssen, welche Enkodierung benutzt werden soll. Das kann in der Benutzeroberfläche über das *Eingabe Zeichenkodierung* Feld im *Ansicht → Text* Abschnitt des Konvertierungsdialogs gemacht werden. Die Kommandozeilenwerkzeuge haben eine Option `ebook-convert-txt-input --input-encoding`.
2. Um calibre HTML-Dateien hinzuzufügen, müssen Sie calibre gegebenenfalls mitteilen, wie die Dateien kodiert sind. Hierzu gehen Sie zu *Einstellungen → Erweitert → Erweiterungen → Dateityp* und legen in der *HTML to ZIP* Erweiterung fest, wie Ihre HTML-Datei kodiert ist. Wenn Sie nun eine HTML-Datei zu calibre hinzufügen, wird diese korrekt verarbeitet. HTML-Dateien aus unterschiedlichen Quellen sind häufig auch unterschiedlich kodiert, weshalb Sie diese Einstellungen möglicherweise mehrmals ändern werden müssen. Eine häufige Kodierung für viele Dateien aus dem Internet ist `cp1252`, weshalb wir vorschlagen, dass Sie diese zuerst probieren. Bitte beachten Sie beim Konvertieren von HTML-Dateien, dass Sie die oben genannten Eingabe-Kodierungs-Einstellungen unausgefüllt lassen müssen, da die *HTML to ZIP* Erweiterung HTML-Dateien automatisch in die Standardkodierung (UTF-8) umwandelt.

9.1.5 Was hat es mit dem Inhaltsverzeichnis in MOBI-Dateien auf sich?

Zuerst muss man verstehen, dass die meisten eBooks zwei Inhaltsverzeichnisse haben. Eins ist das klassische Inhaltsverzeichnis, wie das IHV (Inhaltsverzeichnis), das du in Papier Büchern findest. Dieses Inhaltsverzeichnis ist Teil des Dokumentenflusses und kann formatiert werden wie du möchtest. Dieses IHV (Inhaltsverzeichnis) heißt Inhalts-IHV.

Als nächstes gibt es das *Metadaten-Inhaltsverzeichnis*. Ein Metadaten-IHV ist kein Teil des Buchtextes und wird in der Regel über eine bestimmte Schaltfläche im Reader aufgerufen. Im calibre Ebook Reader zum Beispiel verwendet man den Button Zeige Inhaltsverzeichnis um dieses IHV einzusehen. Dieses IHV kann von calibre nicht formatiert werden. Wie es dargestellt wird, hängt vom Betrachtungsprogramm ab.

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 Go to Table of Contents on your Kindle, it is to this extra content ToC that the Kindle takes you.

Es mag Ihnen jetzt so vorkommen, als hätte das MOBI-Buch zwei identische Inhaltsverzeichnisse. Bedenken Sie, dass eines davon ein *Inhalts-ToC* und das andere ein Metadateninhaltsverzeichnis ist, und beide die gleichen Einträge haben und gleich aussehen können. Das eine kann direkt über das Kindle-Menü aufgerufen werden, das andere nicht.

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.

Wenn Sie ein von Hand bearbeitetes Inhaltsverzeichnis im Eingabedokument haben, können Sie die Inhaltsverzeichnis-erkennen-Optionen in Calibre verwenden, um daraus automatisch das Metadateninhaltsverzeichnis zu erstellen. Lesen Sie für Details den Abschnitt zum Konvertieren im Nutzerhandbuch. Dort erfahren Sie auch, wie Sie diese Funktionen nutzen.

Letztendlich ermutige ich dich das Inhalts-IHV loszuwerden und nur das Metadaten-Inhaltsverzeichnis in deinen eBooks zu behalten. Metadaten-Inhaltsverzeichnisse ermöglichen den Leuten, die deine eBooks lesen, eine deutlich überlegene Navagationserfahrung (außer auf dem Kindle, wo sie prinzipiell das gleiche sind wie ein Inhalts-IHV).

Bemerkung

Das neuere AZW3-Format kann richtig mit dem Metadateninhaltsverzeichnis umgehen. Leider hat die Kindle-Firmware die Tendenz, nicht richtig zu funktionieren, wenn Sie die Erzeugung des Inline-Inhaltsverzeichnisses am Dateiende deaktivieren. Es ist also zu empfehlen, das erzeugte Inhaltsverzeichnis einfach in Ruhe zu lassen. Wenn Sie doch eine AZW3-Datei mit einem Metadateninhaltsverzeichnis, aber ohne Inline-Inhaltsverzeichnis am Dateiende erzeugen, werden einige Dinge auf dem Kindle nicht funktionieren, zum Beispiel PageFlip.

9.1.6 Wie konvertiere ich eine Sammlung von HTML-Dateien in einer bestimmten Reihenfolge?

Um eine Sammlung von HTML-Dateien in einer bestimmten Reihenfolge zu konvertieren müssen Sie eine Inhaltsverzeichnis-Datei erstellen, d.h. eine weitere HTML-Datei, die Links zu allen anderen Dateien in der gewünschten Reihenfolge enthält. Eine solche Datei sieht folgendermaßen aus:

```
<html>
   <body>
     <h1>Table of Contents</h1>
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    <p style="text-indent:0pt">
       <a href="file1.html">First File</a><br/>
       <a href="file2.html">Second File</a><br/>
       .
       .
       .
    </p>
  </body>
</html>
```

Dann fügen Sie einfach diese HTML_Datei zum GUI hinzu und verwenden die *Bücher konvertieren* -Schaltfläche, um Ihr Buch zu erstellen. Sie können im Tab „Inhaltsverzeichnis“ des Konvertierungsfensters festlegen, wie das Inhaltsverzeichnis erzeugt werden soll.

Bemerkung

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 Das mit Calibre erzeugte EPUB ist nicht gültig?

Calibre garantiert nicht, dass ein erstelltes EPUB valide ist. Das Einzige, was garantiert werden kann ist, das bei der Eingabe von validem XHTML 1.1 + CSS 2.1 ein valides EPUB ausgegeben werden kann. Calibre bemüht sich sehr darum, dass die erstellten EPUBs wie beabsichtigt auf vielen verschiedenen Geräten funktionieren. Dieses Ziel ist für die Mehrheit der Nutzer wichtiger, als das Erstellen von validen EPUBs. Wenn Sie ein Werkzeug benötigen, das immer valide EPUBs erzeugt, dann ist Calibre nichts für Sie. Wenn Sie ein von Calibre erstelltes EPUB zu einem online Store senden möchten, der eine EPUB Validitätskontrolle durchführt, müssen Sie selber sicherstellen, dass das die EPUB Dateivalide ist. Calibre wird das nicht für Sie erledigen. –Kurz gesagt: Sie müssen Calibre mit gültigen XHTML + CSS als Eingabedokumente versorgen.

9.1.8 Wie verwende ich einige der fortgeschrittenen Funktionen der Konvertierungswerkzeuge?

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[46].

9.2 Geräteintegration

Inhalt

- *Welche Geräte unterstützt Calibre?* (die Titelseite 133)
- *Was kann ich dazu beitragen, damit mein Gerät in Calibre unterstützt wird?* (die Titelseite 133)

[46] https://calibre-ebook.com/downloads/html-demo.zip

- *Mein Gerät wird von Calibre nicht erkannt?* (die Titelseite 134)
- *Mein Gerät entspricht nicht dem Standard oder ist ungewöhnlich. Was kann ich machen, um es zu verbinden?* (die Titelseite 134)
- *Wie verwende ich Calibre mit meinem iPad/iPhone/iPod touch?* (die Titelseite 134)
- *Wie verwende ich Calibre mit meinem Android Telefon/Tablet oder Kindle Fire?* (die Titelseite 135)
- *Kann ich über den Internetbrowser mit meinem Kindle oder einem anderen Gerät auf meine Bücher in Calibre zugreifen?* (die Titelseite 136)
- *Ich kann keine E-Mails mit Calibre versenden?* (die Titelseite 136)
- *My device is getting mounted read-only in Linux, so calibre cannot connect to it?* (die Titelseite 137)
- *Warum unterstützt Calibre keine Sammlungen für Kindle oder Nook?* (die Titelseite 137)
- *Ich bekomme eine Fehlermeldung in Calibre bei der Verwendung meines Kobo Touch/Glo/usw.* (die Titelseite 138)
- *Covers for books I send to my e-ink Kindle show up momentarily and then are replaced by a generic cover?* (die Titelseite 138)
- *Covers for books sent to my Kindle ColorSoft and newer do not show up in the book list?* (die Titelseite 139)
- *Die Titelbilder meiner MOBI-Dateien werden nicht mehr in meinem Kindle für PC, Android, iPad usw. angezeigt.* (die Titelseite 139)
- *Warum tauchen einige Bücher, die ich mit Calibre an meinen Kindle gesendet habe, nicht auf?* (die Titelseite 139)

9.2.1 Welche Geräte unterstützt 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 Was kann ich dazu beitragen, damit mein Gerät in Calibre unterstützt wird?

Wenn Ihr Gerät im Betriebssystem als USB-Speicher angezeigt wird, dann ist die Unterstützung für Calibre sehr leicht umzusetzen. Wir brauchen nur ein paar Informationen von Ihnen:

- Vollständige Liste an eBook Formaten, die dein Gerät unterstüzt.
- Is there a special folder on the device in which all e-book files should be placed? Also does the device detect files placed in sub-folders?
- Außerdem brauchen wir Informationen zu Ihrem Gerät, die Calibre automatisch sammeln kann. Wenn Ihr Gerät SD-Karten unterstützt, führen Sie diese zuerst ein. Danach verbinden Sie Ihr Gerät mit dem Computer. In Calibre gehen Sie zu *Einstellungen* → *Verschiedenes* und klicken Sie auf die Schaltfläche „Fehlerdiagnose bei der Geräteerkennung“. Es wird ein Fehlerbericht generiert. Kopieren Sie ihn in eine Datei und wiederholen Sie den Vorgang, dieses Mal ohne das angeschlossene Gerät.
- Senden Sie die beiden Ausgaben zusammen mit den anderen Informationen an uns und wir werden einen Treiber für Ihr Gerät schreiben.

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/bugs

9.2.3 Mein Gerät wird von Calibre nicht erkannt?

Folgen Sie diesen Schritten, um das Problem zu finden:

- Stellen Sie sicher, dass Sie nur ein Gerät zur gleichen Zeit mit dem Computer verbinden. Sie sollten zur gleichen Zeit kein von Calibre unterstütztes Gerät wie ein iPhone/iPad usw. angeschlossen haben.
- Wenn du ein Apple i-Gerät (iPad, iPod Touch, iPhone) verbindest, erlaubt Apple Drittanbietersoftware nicht mehr sich mit seinen Geräten mit einem USB Kabel zu verbinden. Nutze stattdessen eine drahtlose Verbindung über den Calibre Inhalteserver.
- If you are connecting a 2024 Kindle or newer or an Android device, and are on macOS or Linux, read the note under *Verwenden eines USB-Kabels* (die Titelseite 135).
- On macOS if you get permission errors when connecting a device to calibre, you can fix that by looking under *System Preferences > Security and Privacy > Privacy > Files and Folders*.
- Make sure you are running the latest version of calibre (currently 8.7.0). The latest version can always be downloaded from the calibre website[48]. You can tell what version of calibre you are currently running by looking at the bottom line of the main calibre window.
- Stelle sicher, dass dein Betriebssystem das Gerät findet. Das bedeutet, das Gerät sollte im Windows Explorer (unter Windows) oder im Finder (unter macOS) auftauchen.
- Gehen Sie in Calibre zu *Einstellungen → Ignorierte Geräte* und stellen Sie sicher, dass Ihr Gerät nicht ignoriert wird.
- Wenn keiner der vorherigen Schritte funktioniert, gehen Sie unter *Einstellungen → Verschiedenes* und klicken Sie auf *Fehlerdiagnose bei der Geräteerkennung*. Stellen Sie die Informationen von dort zusammen mit Ihren Gerätedetails als Ticket im calibre bug tracker[49] ein.

9.2.4 Mein Gerät entspricht nicht dem Standard oder ist ungewöhnlich. Was kann ich machen, um es zu verbinden?

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 Wie verwende ich Calibre mit meinem iPad/iPhone/iPod touch?

An easy way to browse your calibre collection from your Apple device is by using *Der Calibre Inhalteserver* (die Titelseite 113), which makes your collection available over the net. First perform the following steps in calibre

- Stellen Sie in Calibre EPUB als bevorzugtes Zielformat ein (das Zielformat kann unter *Einstellungen → Benutzeroberfläche → Verhalten* festgelegt werden)
- 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.
- 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*.

[48] https://calibre-ebook.com/download
[49] https://bugs.launchpad.net/calibre

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.

Using the Content server

Starten Sie den Safaribrowser und tippen Sie die IP-Adresse und den Port des Computers auf dem der Calibre-Server läuft folgendermaßen ein:

```
http://192.168.1.2:8080/
```

Replace `192.168.1.2` with the local IP address of the computer running calibre. See *Der Calibre Inhalteserver* (die Titelseite 113) for details on running the server and finding out the right IP address to use.

You will see a list of books in Safari, tap on any book and you will be given the option to either download it, or read it in the browser itself. If you choose to download it, Safari will ask you if you want to open it with 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 Wie verwende ich Calibre mit meinem Android Telefon/Tablet oder 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].

Verwenden eines USB-Kabels

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.

Bemerkung

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.

Bemerkung

[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

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 detected.

Kabellos verbinden

calibre has a builtin web server, the *calibre Content server* (die Titelseite 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:

- Stellen Sie in Calibre EPUB als *bevorzugtes Zielformat* ein (das Zielformat kann unter *Einstellungen → Benutzeroberfläche → Verhalten* festgelegt werden)
- Convert the books you want to read on your device to EPUB/MOBI format by selecting them and clicking the *Convert* button.
- Starten Sie den *Inhalteserver* in den Calibre-Einstellungen und lassen Sie Calibre weiter laufen.

Jetzt öffnen Sie auf Ihrem Android Gerät den Browser und gehen zu

http://192.168.1.2:8080/

Replace `192.168.1.2` with the local IP address of the computer running calibre. See *Der Calibre Inhalteserver* (die Titelseite 113) for details on running the server and finding out the right IP address to use.

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 Kann ich über den Internetbrowser mit meinem Kindle oder einem anderen Gerät auf meine Bücher in Calibre zugreifen?

calibre has a *Content server* that exports the books in calibre as a web page. See *Der Calibre Inhalteserver* (die Titelseite 113) for details.

Einige Geräte wie der Kindle (1/2/DX) erlauben keinen Zugriff auf den Port 8080 (der Standard-Port des Inhalteservers). Ändern Sie den Port in diesem Fall in den Calibre-Einstellungen auf 80. (In einigen Betriebssystemen kann es sein, dass Sie den Server aufgrund von Sicherheitseinstellungen nicht auf einem Port kleiner 1024 betreiben können. In diesem Fall wiederum ist die einfachste Lösung eine Portweiterleitung Ihres Routers von 80 auf 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 Ich kann keine E-Mails mit Calibre versenden?

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:

- Erstellen Sie ein kostenloses GMX-Konto auf GMX[55].
- Go to *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

- Log into your GMX account on the website and enable SMTP sending (*Settings->POP3 & IMAP->Send and receive emails via external program*)
- calibre will then be able to use GMX to send the mail.
- If you are sending to your Kindle, remember to update the email preferences on your Amazon Kindle page to allow email sent from your GMX email address. Also note that Amazon does not allow email delivery of AZW3 and new style (KF8) MOBI files. Finally, Amazon recently started sending confirmation emails that you have to click on back to your GMX account before the book is actually delivered.

Selbst mit diesen Maßnahmen können Probleme auftreten. Eine häufige Fehlerquelle sind schlecht gestaltete Anti-Viren-Programme, die Calibre davon abhalten eine Verbindung zum Versenden von E-Mails aufzubauen. Versuchen Sie eine Ausnahme für Calibre in Ihrem Anti-Viren-Programm einzurichten.

Bemerkung

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.

Bemerkung

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.

Bemerkung

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 My device is getting mounted read-only in Linux, so calibre cannot connect to it?

Linux-Kernel laden Geräte im Nur-lesen (read-only) -Modus, wenn ihre Dateisysteme Fehler haben. Sie können das Dateisystem wie folgt reparieren:

```
sudo fsck.vfat -y /dev/sdc
```

Ersetzen Sie /dev/sdc mit dem Pfad zum Geräteknoten Ihres Gerätes. Sie finden den Geräteknoten Ihres Gerätes, der immer unter /dev ist, indem Sie die folgende Ausgabe untersuchen:

```
mount
```

9.2.10 Warum unterstützt Calibre keine Sammlungen für Kindle oder 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].

Bemerkung

Amazon hat die Funktion Sammlungen zu verwalten in neueren Modellen wie dem Kindle Touch und Kindle Fire komplett entfernt. Das macht selbst die oben genannte Erweiterung nutzlos, es sei denn, Sie rooten Ihren Kindle und installieren eine Custom-Firmware.

9.2.11 Ich bekomme eine Fehlermeldung in Calibre bei der Verwendung meines Kobo Touch/Glo/usw.

Der Kobo hat eine sehr fehlerhafte Firmware. Verbindungen schlagen ohne erkennbaren Grund fehl. Bestimmte Kombinationen aus Motherboard, USB-Anschlüssen, -Kabeln und -Hubs neigen dazu, das Problem zu verschärfen. Wenn Sie einen Fehler beim Verbinden Ihres Kobo Touch mit Calibre haben, probieren Sie alle folgenden Schritte aus, die bei *einigen* Calibre-Nutzern das Problem beseitigt haben.

- Verbinden Sie den Kobo direkt mit Ihrem Computer, nicht per USB Hub.
- Versuchen Sie ein anderes USB-Kabel und einen anderen USB-Anschluss Ihres Computers.
- Melden Sie sich beim Kobo ab und gleich danach wieder an. Dadurch wird die Datenbank wieder aufgebaut und die Datenbankfehler bereinigt.
- Versuchen Sie die neuste Firmware auf Ihrem Kobo Touch zu installieren
- Versuchen Sie, ihr Kobo zurückzusetzen (dies behebt das Problem gelegentlich nur für eine Weile bevor es dann wieder auftaucht; in diesem Fall müssen Sie es immer wieder zurücksetzen)
- Versuchen Sie nur ein oder zwei Bücher gleichzeitig auf den Kobo zu übertragen und speichern Sie keine großen Sammlungen auf dem Kobo

9.2.12 Covers for books I send to my e-ink Kindle show up momentarily and then are replaced by a generic cover?

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) Buch mit Calibre an Kindle senden
2) Disconnect the Kindle and wait for Amazon to destroy the cover
3) Kindle mit Calibre wieder verbinden

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.

Bemerkung

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 Covers for books sent to my Kindle ColorSoft and newer do not show up in the book list?

Amazon deliberately broke this functionality in their ColorSoft and newer devices in order to discourage you from reading non Amazon books on their devices. See this forum thread[58] for details. The only known workaround is to send the books as „Personal documents“ to the Kindle which will fix the covers not showing up but break other features such as Whispersync and the books will show up under „Personal documents“ not „Books“ on the device. To enable this in calibre go to *Preferences → Output options → MOBI output* and enable the check box that says *Enable sharing of book content*. This will cause all future books sent to the Kindle by calibre to be marked as personal documents.

9.2.14 Die Titelbilder meiner MOBI-Dateien werden nicht mehr in meinem Kindle für PC, Android, iPad usw. angezeigt.

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).

Beachten Sie, dass dadurch die erzeugte MOBI-Datei unter „Persönliche Dokumente“ und nicht unter „Bücher“ auf dem Kindle Fire auftaucht. Auch Amazons Whispersync wird nicht funktionieren, wohl aber die Titelbilder. Es ist Ihre Entscheidung, welche Funktionalität Ihnen wichtiger ist. Mein Vorschlag ist: Bitten Sie Amazon darum, diesen Fehler zu beseitigen!

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.

Wenn Sie eine MOBI-Datei mit Calibre an einen Kindle-eReader per USB-Verbindung senden, umgeht Calibre diesen Amazon-Bug und lädt selbst eine Titelbildvorschau auf das Gerät. Das funktioniert aber nur bei Verwendung der USB-Verbindung mit Calibre. Beachten Sie, dass Amazon die MOBI-Datei beim E-Mail-Versand automatisch als persönliches Dokument markiert. Das Titelbild wird funktionieren, allerdings wird das Buch unter persönlichen Dokumenten erscheinen.

9.2.15 Warum tauchen einige Bücher, die ich mit Calibre an meinen Kindle gesendet habe, nicht auf?

Auf den Kindle übertragene Bücher tauchen dort erst auf, nachdem Sie von diesem *indexiert* worden sind. Das kann eine Weile dauern. Wenn das Buch nach einiger Zeit noch immer nicht aufgetaucht ist, ist vermutlich der Indexier-Dienst des Kindle abgestürzt. Manchmal kann ein bestimmtes Buch zu diesem Absturz führen. Leider bietet Amazon hier keine Möglichkeit den Absturz des Kindle auf ein Buch zurückzuführen. Sie können nur auf folgende zwei Maßnahmen zurückgreifen: Entweder Sie setzen den Kindle zurück, oder Sie löschen mit dem Windows Explorer (oder der Dateiverwaltung, die Sie stattdessen verwenden) alle Dateien aus dessen Speicher, senden die Bücher diesem dann erneut, aber eines nach dem anderen, zu, bis Sie das problematische Buch gefunden haben, entfernen dieses dann vom Kindle, konvertieren es in Calibre von MOBI nach MOBI oder nach AZW3 und senden es dann zurück. Das wird das Problem höchstwahrscheinlich beseitigen.

9.3 Bibliotheken-Verwaltung

[58] https://www.mobileread.com/forums/showthread.php?t=364350

Inhalt

- *Wo sind die Buchdateien gespeichert?* (die Titelseite 140)
- *Wie verwaltet Calibre Autorennamen und die Sortierung?* (die Titelseite 140)
- *Why doesn't calibre let me store books in my own folder structure?* (die Titelseite 141)
- *Warum hat Calibre keine Spalte für …?* (die Titelseite 142)
- *Kann ich eine Spalte erstellen, welche die Formate oder ISBN anzeigt?* (die Titelseite 142)
- *Wie verschiebe ich meine Calibre-Daten auf einen anderen Computer?* (die Titelseite 143)
- *Die Liste der Bücher in Calibre ist leer!* (die Titelseite 143)
- *Warum bekomme ich Fehler mit meiner Calibre-Bibliothek auf einem Netzwerkspeicher/NAS?* (die Titelseite 144)

9.3.1 Wo sind die Buchdateien gespeichert?

Wenn Sie Calibre das erste Mal starten, werden Sie gefragt, in welchem Ordner Ihre Bücher gespeichert werden sollen. Immer wenn Sie danach ein Buch zu Calibre hinzufügen, wird es in *diesen* Ordner kopiert. Die Bücher sind darin säuberlich in Unter-Ordnern, nach Autor und Titel getrennt, abgelegt. Beachten Sie: Calibre verwaltet den Inhalt dieses Ordners automatisch. Speichern Sie darin also **niemals** irgendwelche Dateien oder Ordner - diese könnten automatisch gelöscht werden. Wenn Sie eine Datei hinzufügen möchten, die zu einem bestimmten Buch gehört, benutzen Sie dazu den Bereich oben rechts im *Metadaten bearbeiten*-Dialog. Dann packt Calibre diese Datei direkt in den richtigen Ordner und nimmt sie auch mit, wenn Sie Änderungen beim Autor oder Titel machen.

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.

Bücher können zwischen verschiedenen Bibliotheken durch Rechtsklick auf das Buch und Auswahl der Aktion *In Bibliothek kopieren* kopiert oder verschoben werden (sobald Sie mehr als eine Bibliothek erstellt oder eingebunden haben).

9.3.2 Wie verwaltet Calibre Autorennamen und die Sortierung?

Author names are complex, especially across cultures, see this note[59] 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.

Kommen wir zur Sortierung nach Autorennamen:

- Wenn ein neuer Autor zu Calibre hinzugefügt wird, (das passiert immer, wenn ein Buch mit einem neuen Autor hinzugefügt wird), dann generiert Calibre automatisch einen Sortierungswert für das Buch und den Autor.
- 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.
- Der Algorithmus nimmt standardmäßig an, dass der Name des Autors in der Form „Vorname Nachname“ vorliegt und generiert ein „Nachname, Name“ Sortierungswert

[59] https://www.w3.org/International/questions/qa-personal-names.en.php?changelang=en

- 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).
- Beim Neuberechnen der Autorensortierung für Bücher verwendet Calibre die Sortierwerte jedes einzelnen Autors. Stellen Sie deshalb sicher, dass die Sortierwerte der einzelnen Autoren korrekt sind, bevor Sie die Neuberechnung der Autorensortierung durchführen lassen.
- 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*

Beachten Sie, dass Sie den Sortierwert eines einzelnen Autors frei wählen können über *Autoren verwalten*. Das ist praktisch, wenn Sie es mit Namen zu tun haben, die Calibre nicht richtig verarbeitet. Beispiele sind komplexe, mehrteilige Namen wie ‚Miguel de Cervantes Saavedra' oder Ostasiatische Namen wie ‚Sun Tsu'.

Durch diese ganze Flexibilität können Sie mit Calibre Autorennamen nach Ihren Wünschen verwalten. Ein häufiger Wunsch ist die Darstellung von Autoren als ‚Nachname, Vorname'. Um das zu erreichen tun sie folgendes (falls der Hinweis unten nicht auf Sie zutrifft):

- Setzen Sie den Wert `author_sort_copy_method` wie oben beschrieben auf `copy`.
- Starten Sie Calibre neu. Ändern Sie keine Metadaten Ihrer Bücher bevor Sie die übrigen Schritte durchführen.
- Ändern Sie im Dialog ‚Autoren verwalten' alle Autorennamen in ‚Nachname, Vorname'.
- Nachdem Sie alle Autoren geändert haben, drücken Sie die Schaltfläche *Alle Werte der Autorensortierung neu berechnen*.
- Drücken Sie ‚OK', dann wird Calibre die Autoren in allen Ihren Büchern abändern. Das kann eine Weile dauern.

Bemerkung

Bei der Änderung von ‚Vorname Nachname' auf ‚Nachname, Vorname' kommt es oft vor, dass die Sortierwerte schon in der Form ‚Nachname, Vorname' vorliegen. Tun Sie in diesem Fall folgendes:

- Setzen Sie den Wert `author_sort_copy_method` wie oben beschrieben auf `copy`.
- Starten Sie Calibre neu. Ändern Sie keine Metadaten Ihrer Bücher bevor Sie die übrigen Schritte durchführen.
- Öffnen Sie den Dialog ‚Autoren verwalten'. Drücken Sie die Schaltfläche `Alle Einträge von "Autorensortierung" zu "Autor" kopieren`.
- Überfliegen Sie die Autoren, um sicherzugehen, dass Sie zufrieden sind. Sie können immer noch ‚Abbrechen' drücken, um die Änderungen zu verwerfen. Sobald Sie ‚OK' drücken, gibt es kein zurück mehr.
- Drücken Sie ‚OK', dann wird Calibre die Autoren in allen Ihren Büchern abändern. Das kann eine Weile dauern.

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
```

Nun erleichtert dieses Schema das Auffinden beispielsweise aller Science-Fiction-Bücher von Isaac Asimov in der Foundation-Serie sehr. Doch was, wenn Sie all Ihre ungelesenen Science-Fiction-Bücher heraussuchen wollten? Dieses Schema bietet keine einfache Lösung, Sie bräuchten vielmehr ein Verzeichnisschema folgender Art:

```
ReadStatus -> Genre -> Author -> Series
```

In Calibre würden Sie stattdessen Schlagworte einsetzen, um Genre und Lesestand zu markieren und dann eine einfache Suchanfrage wie `tag:scifi and not tag:gelesen` durchführen. Calibre hat sogar eine nette grafische Benutzeroberfläche, sodass Sie seine Such-Sprache gar nicht lernen müssen und einfach auf Schlagworte klicken können um sie in die Suche einzubeziehen oder davon auszuschließen.

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.

Zuletzt noch der Grund für die Zahlen am Namensende jedes Titelverzeichnisses: Er ist Robustheit. Die Zahl ist die Identifikationsnummer des Bucheintrags in Calibres Datenbank. Das Vorhandensein dieser Zahl, erlaubt es Ihnen mehrere Einträge des selben Titels und Autors zu haben. Sie ist außerdem einer der Gründe, dafür, dass Calibre seine Datenbank mit allen Metadaten wie von Zauberhand wiederherstellen kann, falls die Datenbankdatei beschädigt wird. Bedenkt man, dass es Calibres Aufgabe ist, Sie davon abzuhalten Metadaten in Dateinamen zu hinterlegen und den Verzeichnisbaum nach Ihren Büchern zu durchsuchen, ist die höhere Robustheit – erreicht durch die Identifikationsnummern – etwas hässlichere Verzeichnisnamen wohl wert.

Wenn Sie immer noch nicht überzeugt sind, befürchte ich, dass Calibre nichts für Sie ist. Suchen Sie anderswo nach Lösungen für Ihre Bedürfnisse. Nur um es klar zu machen, **daran wird sich nichts ändern**. Bitte kontaktieren Sie uns diesbezüglich auch nicht.

9.3.4 Warum hat Calibre keine Spalte für ...?

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[60] to learn how to create your own columns, or read this blog post[61].

Sie können auch „virtuelle Spalten“ anlegen, die Kombinationen der Metadaten aus anderen Spalten enthalten. Verwenden Sie im Dialog ‚Eigene Spalten hinzufügen‘ die ‚Schnellerstellen‘-Verweise um ganz einfach Spalten für die ISBN oder die Formate eines Buchs anzulegen. Sie können Calibres mächtige Template-Sprache (‚Vorlagen-Sprache‘) einsetzen, um noch viel mehr mit Spalten anzustellen. Für weitere Details siehe *Die Calibre-Vorlagensprache* (die Titelseite 160).

9.3.5 Kann ich eine Spalte erstellen, welche die Formate oder ISBN anzeigt?

Ja. Folgen Sie den Anweisungen zur vorherigen Frage „Warum hat Calibre keine Spalte für …?“

[60] https://calibre-ebook.com/demo#tutorials
[61] https://blog.calibre-ebook.com/calibre-custom-columns/

9.3.6 Wie verschiebe ich meine Calibre-Daten auf einen anderen Computer?

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.

Bemerkung

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.

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 Die Liste der Bücher in Calibre ist leer!

Um zu Verstehen, was geschehen ist, müssen Sie verstehen was eine Calibre-Bibliothek ist. Auf der grundsätzlichsten Ebene, ist diese nur ein Ordner. Immer wenn Sie ein Buch zu Calibre hinzufügen, wird dessen Datei (oder Dateien) in diesen Ordner kopiert (geordnet nach Unterordnern für Autor und Titel). In der obersten Ebene von Calibres Bibliotheksordner finden Sie eine Datei mit dem Namen ‚metadata.db'. Diese Datei ist Calibres Speicher für die Metadaten wie Titel, Autor, Bewertung, Schlagwort usw. für *jedes* Buch in Ihrer Calibre-Bibliothek. Die Bücherliste, die Calibre anzeigt wird aus der Datei ‚metadata.db' generiert.

Es gibt zwei mögliche Gründe, warum Calibre eine leere Bücherliste anzeigt:

- Ihr Calibre-Bibliotheksordner hat den Speicherort gewechselt. Das kann passieren, wenn er auf einem externen Datenträger war und sich dessen Laufwerksbuchstabe geändert hat. Oder, wenn Sie den Ordner versehentlich verschoben haben. In solchen Fällen kann Calibre seine Bibliothek nicht finden und startet deshalb stattdessen mit einer leeren. Um dies zu beheben, klicken Sie mit rechts auf das Calibre-Zeichen in der Werkzeugleiste und wählen ‚Bibliothek wechseln/erstellen'. Klicken Sie auf das kleine blaue Zeichen, um den neuen Speicherort Ihrer Calibre-Bibliothek auszuwählen und dann ‚OK'. Wenn Sie den neuen Ort nicht kennen, durchsuchen Sie Ihren PC nach der Datei `metadata.db`.
- Ihre Datenbankdatei `metadata.db` ist gelöscht/beschädigt worden. In diesem Fall können Sie Calibre die Datenbankdatei aus seinen Sicherungen wiederherstellen lassen. Ein Rechtsklick auf das Calibre-Zeichen in Calibres

Werkzeugleiste und dann Klicks auf ‚Bibliothekswartung'->'Datenbank wiederherstellen', und Calibre wird die Datei ‚metadata.db' automatisch wiederaufbauen.

9.3.8 Warum bekomme ich Fehler mit meiner Calibre-Bibliothek auf einem Netzwerkspeicher/NAS?

Legen Sie Ihre Calibre-Bibliothek nicht auf ein Netzlaufwerk.

Ein Dateisystem ist ein kompliziertes Wesen. Den meisten Netzwerk-Dateisysteme fehlen Funktionen, die Calibre verwendet. Einige unterstützen keine Sperrung von Dateien (‚locking'), einige keine festen Verknüpfungen (‚hardlinking') und andere sind einfach „wackelig". Außerdem ist Calibre eine Anwendung für Einzelbenutzer, sollten Sie also versehentlich zwei Instanzen von Calibre gleichzeitig auf eine Bibliothek im Netzwerk zugreifen lassen, hätte das schlimme Folgen. Schlußendlich arbeiten verschiedene Betriebssysteme mit verschiedenen Einschränkungen von Dateisystemen, sollten Sie also auf Ihr Netzlaufwerk mit unterschiedlichen Betriebssystemen zugreifen, hätte auch das *schlimme Folgen*.

Ziehen Sie die Nutzung des Calibre-Inhalteserver in Betracht, um Ihre Bücher anderen Computern zur Verfügung zu stellen. Lassen Sie Calibre auf einem einzelnen Computer laufen und greifen Sie darauf über den Inhalteserver oder eine Fernzugriffs-Lösung (Remote-Desktop) zu.

Wenn Sie unbedingt die Bibliothek selbst teilen wollen, verwenden Sie ein Dateisynchronisierungs-Werkzeug wie ‚Dropbox' oder ‚rsync' statt eines Netzlaufwerks. Wenn Sie ein solches einsetzen ist es **unbedingt nötig**, dass Sie Calibre und das Synchronisierungswerkzeug nicht gleichzeitig auf die Bibliothek zugreifen lassen. Anders gesagt: Lassen Sie das Synchronisierungswerkzeug und Calibre **nicht zusammen** laufen.

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[62] for details.

9.4 Verschiedenes

Inhalt

- *Amazon is stopping email delivery of MOBI files?* (die Titelseite 145)
- *Ich möchte, dass Calibre Nachrichten von meiner Lieblingsseite herunterlädt.* (die Titelseite 145)
- *Wieso der Name Calibre?* (die Titelseite 146)
- *Warum zeigt Calibre nur einige meiner Schriftarten auf macOS?* (die Titelseite 146)
- *Calibre startet nicht unter Windows?* (die Titelseite 146)
- *Calibre friert gelegentlich ein oder stürzt manchmal ab?* (die Titelseite 147)
- *The calibre E-book viewer and Edit book tools do not work on Windows?* (die Titelseite 147)
- *Using the viewer or doing any conversions results in a permission denied error on Windows* (die Titelseite 148)
- *Calibre startet nicht/stürzt ab auf macOS?* (die Titelseite 148)
- *I get only a black or white screen when running the calibre E-book viewer?* (die Titelseite 148)
- *Ich habe das Installationsprogramm heruntergeladen, aber es funktioniert nicht?* (die Titelseite 148)
- *Mein Antivirenprogramm behauptet, Calibre sei ein Virus/Trojaner?* (die Titelseite 149)
- *Wie erstelle ich eine Datensicherung von Calibre?* (die Titelseite 149)

[62] https://www.mobileread.com/forums/showthread.php?t=205581

- *Wie verwende ich gekaufte EPUB-Bücher mit Calibre (oder was mache ich mit .acsm-Dateien)?* (die Titelseite 149)
- *Ich bekomme einen „Zugriff verweigert“-Fehler* (die Titelseite 150)
- *Kann ich die Metadaten-Kommentare auf meinem Reader anzeigen lassen?* (die Titelseite 150)
- *Wie kann ich Calibre dazu bringen, meinen HTTP-Proxy zu verwenden?* (die Titelseite 150)
- *Ich möchte, dass eine neue Funktion zu Calibre hinzugefügt wird. Was kann ich tun?* (die Titelseite 150)
- *Warum verfügt Calibre nicht über eine automatische Aktualisierung?* (die Titelseite 151)
- *Wie ist Calibre lizenziert?* (die Titelseite 151)
- *Wie kann ich Calibre von meinem USB-Stick aus ausführen?* (die Titelseite 151)
- *How do I run parts of calibre like news download and the Content server on my own Linux server?* (die Titelseite 151)

9.4.1 Amazon is stopping email delivery of MOBI files?

Amazon have announced[63] 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. EPUB-Datei in MOBI konvertieren
2. Then convert the MOBI file back to EPUB and send the resulting EPUB file

This will remove all advanced formatting, embedded fonts, etc., but greatly increase the chances of Amazon accepting the EPUB.

Bemerkung

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 Ich möchte, dass Calibre Nachrichten von meiner Lieblingsseite herunterlädt.

Wenn Sie im Umgang mit Computern tüchtig genug sind, können Sie Calibre das Herunterladen von Nachrichten von jeder beliebigen Webseite beibringen. Um dies zu erlernen, siehe *Hinzufügen Ihrer bevorzugten Nachrichtenwebsite* (die Titelseite 33).

Otherwise, you can request a particular news site by posting in the calibre Recipes forum[64].

[63] https://blog.the-ebook-reader.com/2022/05/03/amazon-dropping-mobi-support-on-send-to-kindle-apps/
[64] https://www.mobileread.com/forums/forumdisplay.php?f=228

9.4.3 Wieso der Name Calibre?

Suchen Sie es sich aus:

- Converter And LIBRary for E-books
- Ein Produkt mit großem „Kaliber“
- A tribute to the SONY Librie which was the first e-ink based e-book reader
- Meine Frau hat es ausgesucht ;-)

Calibre wird Cal-i-ber und nicht Ca-li-bre ausgesprochen. Wenn Sie sich wundern, calibre ist die britische/commonwealth Schreibweise für caliber. Für mich als Inder ist das die natürliche Aussprache.

9.4.4 Warum zeigt Calibre nur einige meiner Schriftarten auf 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 startet nicht unter Windows?

Dafür kann es mehrere Ursachen geben:

- 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[65].
- Some software has been known to interfere with calibre, try rebooting in Safe mode and see if it works. A known culprit is the Sunshine[66] screen sharing software.
- Wenn Sie die Fehlermeldung bekommen, dass Calibre eine Datei nicht öffnen kann, weil es von einem anderen Programm verwendet wird, machen Sie folgendes:
  - Deinstallieren Sie Calibre
  - Starten Sie den Computer neu
  - Installieren Sie Calibre neu, aber starten Sie es nicht mit dem Installations-Assistenten.
  - Deaktivieren Sie vorübergehend Ihr Antivirenprogramm (sicherheitshalber davor vom Internet trennen)
  - Sehen Sie in dem für die Calibre-Bibliothek gewählten Ordner nach. Wenn sich dort eine Datei namens ‚metadata.db‘ befindet, löschen Sie diese.
  - Calibre starten
  - Von jetzt an sollten Sie in der Lage sein, Calibre normal zu starten.
- Falls Sie nach dem Upgrade von Calibre eine Fehlermeldung über ein unerwartetes Beenden einer Python-Funktion erhalten, deinstallieren Sie Calibre erst und löschen dann die Ordner (falls vorhanden) `C:\Programme\Calibre` und `C:\Programme\Calibre2`. Installieren Sie das Programm noch einmal, dann sollte alles in Ordnung sein.
- 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:

[65] https://www.wikihow.com/Bring-an-Off-Screen-Window-Back-on-Windows
[66] https://github.com/LizardByte/Sunshine

```
calibre-debug -g
```

Post any output you see in a help message on the Forum[67].

9.4.6 Calibre friert gelegentlich ein oder stürzt manchmal ab?

Es gibt mehrere bekannte Gründe, die das verursachen können:

- Sie haben kürzlich einen externen Bildschirm oder Fernseher an Ihren PC angeschlossen. In diesem Fall öffnet Calibre neue Fenster, wie das zum Bearbeiten von Metadaten oder das für den Konverter-Dialog, immer auf dem zweiten Bildschirm, wo Sie es nicht bemerken und deshalb davon ausgehen, Calibre sei stehengeblieben. Trennen Sie Ihren Zweitbildschirm und starten Calibre neu.
- 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*.
- Sie benutzen eine USB-Maus oder ein Tablet der Marke Wacom. Es gibt eine Inkompatibilität zwischen dem Treiber von Wacom und dem Grafik-Toolkit das von Calibre verwendet wird. Versuchen Sie eine andere Maus zu verwenden.
- Auf eingigen 64-Bit-Versionen von ‚Windows', gibt es Sicherheitsprogramme/-einstellungen, die die 64-Bit-Version von Calibre behindern. Falls Sie diese verwenden, versuchen Sie es mit einem Wechsel auf die 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[68] 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 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[69] for details.

[67] https://www.mobileread.com/forums/forumdisplay.php?f=166
[68] https://www.nirsoft.net/utils/shexview.html
[69] https://doc.qt.io/qt-6/qtwebengine-debugging.html

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.

Manche Benutzer berichten, dass die Deaktivierung von UAC das Problem löst.

9.4.9 Calibre startet nicht/stürzt ab auf 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 and Default Folder X.

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[70]. 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".

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 *Umgebungsvariablen* (die Titelseite 287) for how to change environment variables.

9.4.11 Ich habe das Installationsprogramm heruntergeladen, aber es funktioniert nicht?

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[71]. 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.

[70] https://typefaceapp.com/help/troubleshooting/macos-font-cache
[71] https://github.com/kovidgoyal/calibre/releases/latest

- If you have installed to a non-standard location, try running the installer as Administrator
- Try rebooting your computer and running a registry cleaner like Wise registry cleaner[72].
- 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[73], which does not need an installer (it is just a ZIP file).

9.4.12 Mein Antivirenprogramm behauptet, Calibre sei ein Virus/Trojaner?

The first thing to check is that you are downloading calibre from the official website[74]. 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.

9.4.13 Wie erstelle ich eine Datensicherung von 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 Wie verwende ich gekaufte EPUB-Bücher mit Calibre (oder was mache ich mit .acsm-Dateien)?

Most purchased EPUB books have *DRM* (die Titelseite 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.

[72] https://www.wisecleaner.com
[73] https://calibre-ebook.com/download_portable
[74] https://calibre-ebook.com/download

9.4.15 Ich bekomme einen „Zugriff verweigert“-Fehler

Ein „Zugriff verweigert“-Fehler kann aus verschiedenen Gründen auftreten, keine haben aber ihre Ursache bei Calibre.

- You can get permission denied errors if you are using an SD card with write protect enabled.
- On macOS if you get permission errors when connecting a device to calibre, you can fix that by looking under *System Preferences* > *Security and Privacy* > *Privacy* > *Files and Folders*.
- 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.
- Wenn deine Datei in einem anderen Programm geöffnet ist.
- 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 Kann ich die Metadaten-Kommentare auf meinem Reader anzeigen lassen?

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 Wie kann ich Calibre dazu bringen, meinen HTTP-Proxy zu verwenden?

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 Ich möchte, dass eine neue Funktion zu Calibre hinzugefügt wird. Was kann ich tun?

Du hast zwei Auswahlmöglichkeiten:

1. Create a patch by hacking on calibre and send it to me for review and inclusion. See Development[75].
2. Open a bug requesting the feature[76]. 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,

[75] https://calibre-ebook.com/get-involved
[76] https://calibre-ebook.com/bugs

calibre has a comprehensive plugin architecture, so you might be able to develop your feature as a plugin, see *Schreibe Deine eigenen Erweiterungen zur Anpassung von Calibres Funktionalität* (die Titelseite 225).

9.4.19 Warum verfügt Calibre nicht über eine automatische Aktualisierung?

Aus vielen Gründen:

- *Es ist nicht notwendig, Calibre jede Woche zu aktualisieren*. Wenn Sie damit zufrieden sind, wie Calibre bei Ihnen läuft, können Sie die automatische Benachrichtigung für Aktualisierungen abschalten. Sie können es jeder Zeit wieder aktivieren. Die Benachrichtigungen können Sie direkt im Benachrichtigungsfenster deaktivieren.
- calibre downloads currently use about 150TB of bandwidth a month[77]. 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* (die Titelseite 345).
- There are third party automatic updaters for calibre made by calibre users in the calibre forum[78].
- Additionally, some third-party updaters such as Norton or Avast may update software behind the user's back. If you find calibre has updated unexpectedly, check for the presence of one.

9.4.20 Wie ist Calibre lizenziert?

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[79]. 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[80].

9.4.21 Wie kann ich Calibre von meinem USB-Stick aus ausführen?

A portable version of calibre is available here[81].

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.

Bemerkung

calibre needs GLIBC >= 2.31 and libstdc++ >= 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

[77] https://calibre-ebook.com/dynamic/downloads
[78] https://www.mobileread.com/forums/forumdisplay.php?f=238
[79] https://download.calibre-ebook.com
[80] https://www.gnu.org/licenses/gpl.html
[81] https://calibre-ebook.com/download_portable

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`.

Sie können den Calibre Netzwerkserver mit dem folgenden Befehl starten:

```
/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" outputfile.epub
```

If you want to generate MOBI, use `outputfile.mobi` instead and use `--output-profile kindle`.

Sie können mit folgendem Befehl heruntergeladene Nachrichten per E-Mail verschicken:

```
/opt/calibre/calibre-smtp
```

I leave figuring out the exact command line as an exercise for the reader.

Schlussendlich können Sie heruntergeladene Nachrichten wie folgt zur Calibre-Bibliothek hinzufügen:

```
/opt/calibre/calibredb add --with-library /path/to/library outfile.epub
```

Remember to read the *Befehlszeilenschnittstelle* (die Titelseite 303) section of the calibre User Manual to learn more about these, and other commands.

KAPITEL 10

Tutorials

Hier finden Sie Tutorials zu Calibres erweiterten Funktionen wie XPath und Vorlagen.

10.1 Untergruppen von Büchern verwalten, wie zum Beispiel „Genre“

Einige Menschen möchten Bücher in ihrer Bibliothek in Untergruppen, ähnlich einem Unterordner organisieren. Der meist genannte Grund ist neben vielen anderen die Erstellung von Genre Hierarchien. Ein Benutzer fragte nach einer Möglichkeit, Lehrbücher nach Thema und Kurs zu organisieren. Ein anderer wollte Geschenke nach Thema und Empfänger verfolgen. Dieses Tutorials wird das Genre-Beispiel für den Rest dieses Beitrags verwenden.

Bevor Sie weitermachen, beachten Sie bitte, dass wir nicht über Ordner auf Festplatten sprechen. Untergruppen sind keine Dateiordner. Es werden keine Bücher kopiert. Die Dateistruktur der Calibre-Bibliothek wird nicht verändert. Stattdessen zeigen wir eine Möglichkeit zum Organisieren und Anzeigen von Büchern in Untergruppen innerhalb der Calibre-Bibliothek.

- *Einrichtung* (die Titelseite 155)
- *Suche* (die Titelseite 157)
- *Beschränkungen* (die Titelseite 157)
- *Nützliche Vorlagefunktionen* (die Titelseite 158)

Die meist genannten Anforderungen für Untergruppen wie Genres sind:

- Eine Untergruppe (z. B. ein Genre) muss Bücher enthalten (zeigen) und keine Kategorien von Büchern. Dies unterscheidet Untergruppen von calibe Benutzerkategorien.
- Ein Buch kann in mehreren Untergruppen (Genres) enthalten sein. Dies unterscheidet Untergruppen von physischen Datei-Ordnern.
- Untergruppen (Genres) müssen eine Hierarchie bilden; Untergruppen können Untergruppen enthalten.

Schlagwörter können die ersten beiden Anforderungen erfüllen. Wenn Sie ein Buch ein Genre-Schlagwort zuweisen, können Sie den Schlagwortbrowser (oder die Suche) benutzen, um die Bücher zu finden, welche dieses Genre verwenden. Dies entspricht der ersten Anforderung. Viele Bücher können gleiche Schlagwörter haben, dies entspricht der zweiten Anforderung. Das Problem ist, dass Schlagwörter nicht die dritte Voraussetzung erfüllen. Sie unterstützen keine Hierarchien.

Calibres Hierarchiefunktion bietet Ihnen die dritte Voraussetzung, Genres als ein „Baum“ zu sehen und bietet die Möglichkeit, ganz einfach Bücher im Genre oder Untergenre zu suchen. Nehmen wir beispielsweise an, dass die Genrestruktur ähnlich der folgenden ist:

```
Genre
    . History
    .. Japanese
    .. Military
    .. Roman
    . Mysteries
    .. English
    .. Vampire
    . Science Fiction
    .. Alternate History
    .. Military
    .. Space Opera
    . Thrillers
    .. Crime
    .. Horror
    etc.
```

Mit der Hierarchiefunktion sehen Sie diese Genres im Schlagwortbrowser in Baumstruktur, wie es im Bild dargestellt ist. In diesem Beispiel ist die äußerste Ebene (Genre) eine benutzerdefinierte Spalte mit den Genres. Genres mit Untergenres werden mit einem kleinen Dreieck neben dem Namen angezeigt. Klickt man auf das Dreieck, öffnet sich das Element und die Untergenres werden angezeigt, wie es im Beispiel mit History und Science Fiction zu sehen ist.

Durch einen Klick auf ein Genre können alle Bücher mit diesem Genre oder deren untergeordneten Genres gesucht werden. Beispielsweise werden durch anklicken von Science-Fiction alle drei untergeordneten Genres Alternate History,

Military und Space Opera angezeigt. Klick auf Alternate History zeigen Bücher mit diesem Genre, die Genres Military und Space Opera werden ignoriert. Natürlich kann ein Buch auch mehrere Genres haben. Wenn ein Buch das Genre Space Opera und Military besitzt, wird das Buch angezeigt, falls eins der verwendeten Genres ausgewählt wird. Die Suche wird im Folgenden näher erläutert.

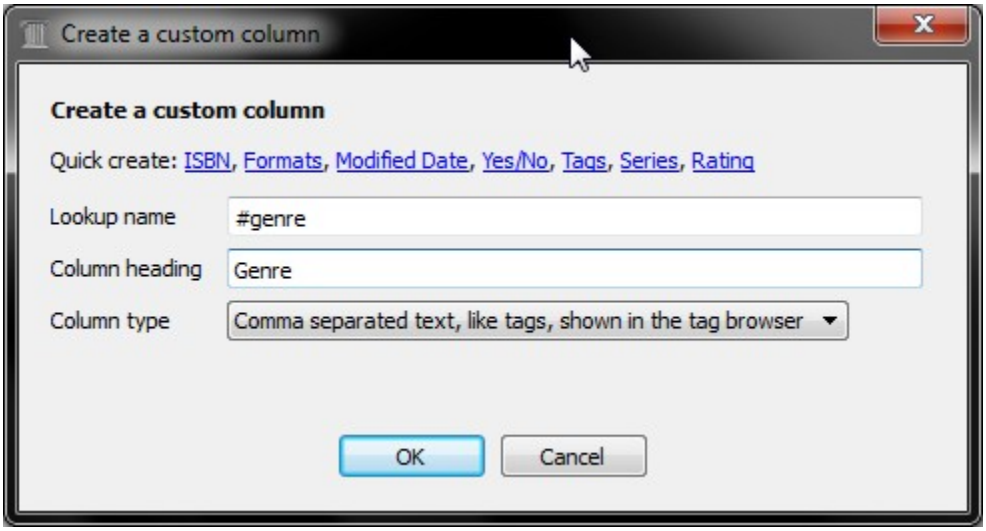
Eine weitere im Bild aufgezeigte Eigenschaft ist, dass das Genre Militär zweimal, einmal unter Geschichte und einmal unter Science-Fiction, angezeigt wird. Weil diese Genres in einer Hierarchie eingebunden sind, sind es zwei separate Genres. Ein Buch kann in einem, einem anderen, oder (in diesem zweifelhaften Fall) in beiden Genres enthalten sein. Beispielsweise können die Bücher von Winston Churchills „The Second World War“ zu „History.Military“ gehören. David Weber's Honor Harrington Bücher könnten zu „Science Fiction.Military“ und ebenfalls zu „Science Fiction.Space Opera.“ Gehören.

Sobald es ein Genre gibt, d. h. mindestens ein Buch dieses Genres enthält, kann es leicht auf andere Bücher angewendet werden, indem man die Bücher aus der Bibliotheks-Ansicht mit der Maus auf das Genre zieht. Genres können auch im Metadaten Editor verwendet werden; mehr dazu weiter unten.

10.1.1 Einrichtung

Nun ist Ihre Frage wahrscheinlich „wie funktioniert das alles?“ Hierzu sind drei Schritte notwendig: 1) das Erstellen einer benutzerdefinierte Spalte, 2) Calibre mitteilen, dass die neue Spalte als Hierarchie behandelt werden soll und 3) das Hinzufügen von 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. It is important that the column type is set to *Comma-separated text, like tags, shown in the Tag browser*.

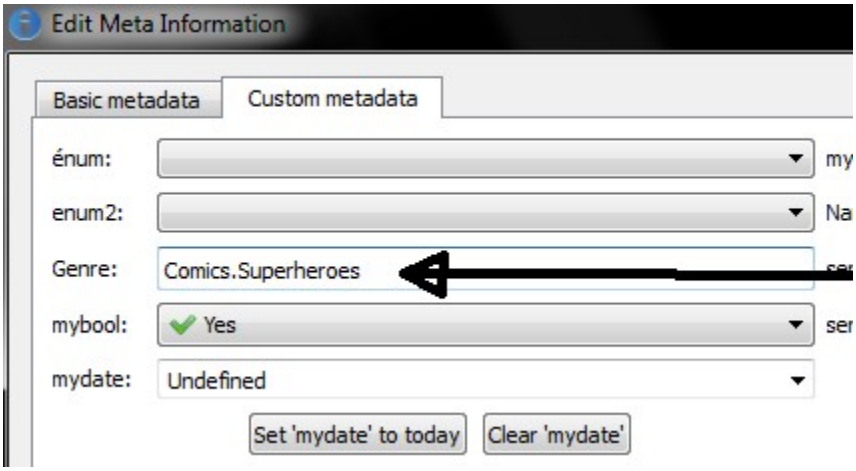


Then after restarting calibre, you must tell calibre that the column is to be treated as a hierarchy. Go to *Preferences → Look & feel → Tag browser → Hierarchy and searching* and choose the new Genre column as having hierarchical items.

An diesem Punkt gibt es noch keine Genres in der Spalte. Wir sind beim fehlenden letzten Schritt: Wie ein Genre auf ein Buch angewendet wird. Ein Genre existiert nicht in Calibre, solange nicht mindestens ein Buch ein Genre zugewiesen bekommt. Um zu lernen, wie man das erste Mal ein Genre verwendet, müssen wir zuvor in einige Details zu einem Genre gehen und sehen, wie diese in den Metadaten zum Buch aussehen.

Eine Hierarchie von „Dingen“ wird durch Erstellen eines Elements erstellt. Dies besteht aus Phrasen, die durch Punkte getrennt sind. Zur Fortsetzung des Genre-Beispiels, würden diese Elemente „History.Military“, „Mysteries.Vampire“, „Science Fiction.Space Opera“ usw. heißen. Um somit ein neues Genre zu erstellen, wird das passende Buch ausgewählt und dessen Metadaten bearbeitet. Geben Sie das neue Genre in die neu erstellte Spalte ein. Unser Beispiel fortführend, wenn Sie das neue Genre „Comics“ mit dem Untergenre „ Superheroes „ einem Buch zuweisen möchten, würden Sie für dieses (comic) Buch „Metadaten bearbeiten“ wählen, in die Registerkarte benutzerdefinierte Metadaten gehen und

dort dann“Comics.Superheroes“, wie im Folgenden dargestellt, eingeben (ignorieren Sie die anderen benutzerdefinierten Spalten).



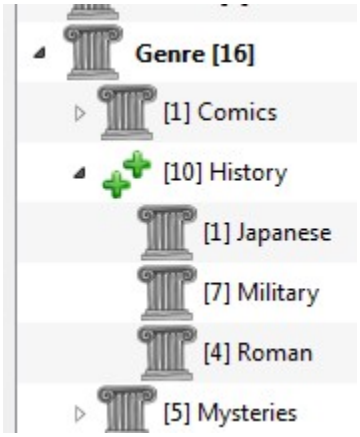
Nachdem obiges umgesetzt wurde, sehen Sie im Schlagwortbrowser:

Um von hier aus einem neuen Buch (vermutlich ein Comic Buch) dieses Genre zuzuweisen, wird dieses Buch auf das Genre im Schlagwortbrowser gezogen oder dem Buch das Genre mithilfe von Metadaten bearbeiten in der gleichen Weise hinzugefügt, wie es oben gezeigt wurde.

Bemerkung

Die Hierarchische Anzeige funktioniert nur, wenn der Schlagwortbrowser darauf eingestellt ist, Elemente nach Namen zu sortieren. Das ist die Standardeinstellung und kann mit einem Klick auf die *Konfigurieren* Schaltfläche unten im Schlagwortbrowser überprüft werden.

10.1.2 Suche



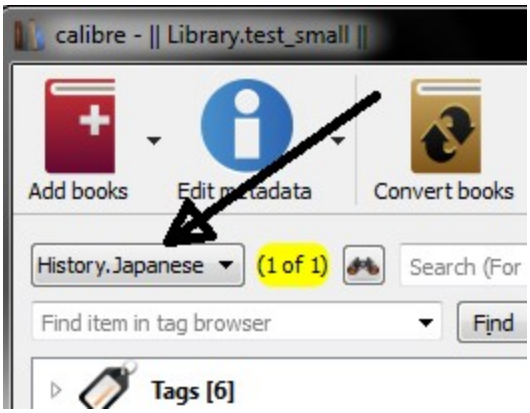
Der einfachste Weg, nach Genres zu suchen ist die Verwendung des Schlagwortbrowsers und ein Klick auf das gewünschte Genre. Der Klick auf ein Genre mit Untergruppen zeigt Ihnen Bücher mit diesem Genre und allen untergeordneten Genres. Dies könnte jedoch eine weitere Frage aufkommen lassen. Nur weil ein Genre Untergruppen enthält, bedeutet dies nicht, dass es kein eigenständiges Genre ist. Beispielsweise kann ein Buch das Genre „History“ enthalten, jedoch nicht das Genre „History.Military“ der Untergruppe. Wie sucht man nun Bücher, die nur in dem Genre „History“ enthalten sind?

Der Suchmechanismus des Schlagwortbrowsers weiß, ob ein Element untergeordneten Elemente aufweist. Ist dies der Fall ist, durchläuft das angeklickte Element 5 statt der üblichen drei Phasen der Suche. Die erste Phase der Suche ist das normale grüne Plus-Zeichen, welches nur Bücher mit diesem Genre (z. B. History) zeigt. Die zweite Phase der Suche ist ein doppeltes Plus-Zeichen (siehe oben), es zeigt Bücher mit diesem Genre und alle Untergenres (z. B. History und History.Military). Die dritte Phase der Suche ist das normale rote Minus-Zeichen, es zeigt alle Bücher ohne dieses Genre. Die vierte Phase der Suche ist das doppelte Minus-Zeichen, es zeigt alle Bücher ohne dieses Genre und den zugehörigen Untergenres. Der fünfte Phase bedeutet zurück zum Anfang und das Genre-Symbol ist nicht mit einem Zeichen unterlegt, d. h. es ist keine Suche aktiv.

10.1.3 Beschränkungen

Wenn du nach einem Genre suchst, dann erstelle dafür eine gespeicherte Suche. Du kannst das ‚Beschränke auf‘-Feld verwenden um eine Virtuelle Bibliothek an Büchern mit diesem Genre zu erstellen. Das ist nützlich, wenn du andere Suchen innerhalb dieses Genres durchführen oder Metadaten für die Bücher in diesem Genre verwalten oder ändern möchtest. Um mit unserem Beispiel fortzufahren, kannst du eine Gespeichert Suche mit dem Namen ‚Geschichte.Japanisch‘ erstellen. Klicke auf das Genre Japanisch im Tag-Browser um eine Suche im Suchfeld zu starten, gib Geschichte.Japanisch in das Feld Gespeicherte Suche ein und drücke die Schaltfläche „Speichere Suche“ (das grüne Kästchen mit dem weißen Plus, auf der rechten Seite).

Nach der Erstellung der gespeicherten Suche können Sie diese zur Einschränkung des Buchbestandes benutzen, beispielsweise für eine virtuelle Bibliothek.



10.1.4 Nützliche Vorlagefunktionen

Vielleicht möchten Sie die Genre-Informationen in einer Vorlage zu verwenden, z. B. in Verbindung mit „Bücher exportieren“ oder „An Gerät senden“. Die Frage wäre dann, „Wie bekomme ich das oberste Genre, den Namen oder die Namen?“ Um dies zu vereinfachen, steht hierfür die Calibre Vorlagefunktion „subitems“ zur Verfügung.

Angenommen, Sie möchten der Vorlage für „Auf Festplatte speichern“ die oberste Genre-Ebene als Genre-Ordner hinzufügen, wie hier in diesem Beispiel das Genre History als Ordner für alle historischen Bücher: „History /The Gathering Storm - Churchill, Winston“. Hierzu müssen Sie die erste Ebene der Hierarchie extrahieren und diese zusammen mit einem Schrägstrich am Anfang des Templates einfügen, um einen Ordner zu erstellen. Die Vorlage unten erreicht dies:

```
{#genre:subitems(0,1)||/}{title} - {authors}
```

Siehe *Die Calibre-Vorlagensprache* (die Titelseite 160) für weiterführende Informationen zu Vorlagen und der Funktion `subitems()`.

10.2 XPath-Tutorial

In dieser Anleitung bekommen Sie eine einfache Einführung in XPath[82], einer Abfragesprache, zum Auswählen von beliebigen Teilen innerhalb eines HTML[83] Dokuments in Calibre. XPath ist ein weit verbreiteter Standard der, wenn Sie danach googeln, viele Anwendungsbeispiele aufzeigt. Diese Anleitung wird sich auf die Verwendung von XPath für Aufgaben rund um eBooks konzentrieren, wie z. B. um Kapitelüberschriften in einem unstrukturierten HTML-Dokument zu finden.

Inhalt

- *Auswahl über Tag-Name* (die Titelseite 159)
- *Auswahl über Attribute* (die Titelseite 159)
- *Auswahl nach Tag-Inhalt* (die Titelseite 159)
- *Ein Beispiel e-Book* (die Titelseite 160)
- *Integrierte XPath-Funktionen* (die Titelseite 160)

[82] https://de.wikipedia.org/wiki/XPath
[83] https://de.wikipedia.org/wiki/HTML

10.2.1 Auswahl über Tag-Name

Die einfachste Form einer Auswahl ist die Auswahl von Tags durch deren Namen. Wenn man z. B. alle <h2>-Tags in einem Dokument finden möchte, ist die dazu passende XPath-Abfrage:

```
//h:h2        (Selects all <h2> tags)
```

Das Präfix // bedeutet *suche auf jeder Ebene des Dokuments*. Angenommen, Sie möchten nun nach -Tags suchen, die sich innerhalb von <a>-Tags befinden. Dies wird durch den folgenden Ausdruck erreicht:

```
//h:a/h:span    (Selects <span> tags inside <a> tags)
```

Wenn Sie nach Tags einer bestimmten Ebene im Dokument suchen möchten, ändern Sie das Präfix wie folgt ab:

```
/h:body/h:div/h:p (Selects <p> tags that are children of <div> tags that are
             children of the <body> tag)
```

Dieser wird Ein sehr kleines eBook, um die Verwendung von XPath zu demonstrieren. im *Ein Beispiel e-Book* (die Titelseite 160) finden und keine anderen -Tags. Das Präfix h: in den obigen Beispielen ist zum Finden von XHTML-Tags erforderlich. Der Grund hierfür liegt in der internen Verarbeitung von Calibre, da alle Inhalte als XHTML-Code interpretiert werden. In XHTML haben Tags einen *Namensraum* und h: ist der Namensraum-Präfix für HTML-Tags.

Nehmen wir nun an, Sie möchten sowohl <h1> als auch <h2> Tags auswählen. Um dies zu erreichen, brauchen wir ein XPath-Konstrukt, welches *Prädikat* genannt wird. Ein *Prädikat* ist ein Test, um gesuchte Tags zu finden. Tests können beliebig umfangreich sein. Mit den weiteren Beispielen werden diese umfangreicher. Man erstellt ein Prädikat, indem der Testausdruck in eckigen Klammern eingeschlossen wird:

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

In diesem XPath-Ausdruck gibt es einige neue Eigenschaften. Die erste ist die Verwendung des Platzhalters *. Dieser bedeutet *finde jeden Tag*. Nun schauen wir uns den Testausdruck name()='h1' or name()='h2' an. *name()* ist ein Beispiel für eine *integrierte Funktion*. Hierbei wird der Name des angegebenen Tags ausgewertet. Wendet man dies wie in unserem Beispiel an, werden alle Tags mit dem Namen *h1* oder *h2* gefunden. Beachten Sie, dass die Funktion *name()* Namensräume ignoriert. Es besteht daher keine Notwendigkeit für die Anwendung des Präfix h:. XPath hat einige nützliche Funktionen integriert. Weitere werden in dieser Anleitung vorgestellt.

10.2.2 Auswahl über Attribute

Um Tags nach ihren Attributen auszuwählen, ist die Verwendung von Prädikaten erforderlich:

```
//*[@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")
```

Der @-Operator bezieht sich auf die Attribute eines Tags. Sie können einige der *XPath built-in functions* (die Titelseite 160) nutzen, um anspruchsvollere Abfragen auf Attributwerte auszuführen.

10.2.3 Auswahl nach Tag-Inhalt

Mithilfe von XPath können Sie auch Tags auswählen, die sich auf einen im Tag enthaltenen Text beziehen. Der beste Weg ist die Verwendung der Fähigkeiten von *regulären Ausdrücken* über die integrierte Funktion *re:test()*:

```
//h:h2[re:test(., 'chapter|section', 'i')] (Selects <h2> tags that contain the words␣
↪chapter or
                                        section)
```

Der `.`-Operator bezieht sich auf den Inhalt eines Tags, sowie sich der `@`-Operator auf seine Attribute bezieht.

10.2.4 Ein Beispiel 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 Integrierte XPath-Funktionen

name()
: Der Name des aktuellen Tags.

contains()
: `contains(s1, s2)` gibt *true* zurück wenn s1 s2 beinhaltet.

re:test()
: `re:test(src, pattern, flags)` gibt *true* zurück, wenn der String *src* dem regulären Ausdruck *pattern* entspricht . Ein möglich anzufügendes Flag ist `i`, das verzichtet auf Groß- und Kleinschreibung. Ein gutes Beispiel dafür finden Sie unter regexp syntax[84]

10.3 Die Calibre-Vorlagensprache

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.

[84] https://docs.python.org/library/re.html

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 `#`. 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 Erweiterte Formatierung

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_textXXXXXsuffix_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[85].

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

[85] https://docs.python.org/3/library/string.html#formatstrings

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. 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* (die Titelseite 171) and *General Program Mode* (die Titelseite 166).

Notes on calling functions in Single Function Mode:

- When functions are used in Single Function Mode, the first parameter, `value`, is automatically replaced by the content of the field specified in the template. For example, when the template `{title:capitalize()}` is processed, the content of the `title` field is passed as the parameter `value` to the capitalize function.
- In the function documentation, 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).
- Some functions use regular expressions. In the template language regular expression matching is case-insensitive.

Functions are documented in *Referenz zur Vorlagefunktion* (die Titelseite 180). The documentation tells you what arguments the functions require and what the functions do. For example, here is the documentation of the *ifempty* (die Titelseite 206) function.

- `ifempty(value, text_if_empty)` – if the `value` is not empty then return that `value`, otherwise return `text_if_empty`.

You see that the function requires two arguments, `value` and `text_if_empty`. However, because we are using Single Function Mode, we omit the `value` argument, passing only `text_if_empty`. For example, this template:

```
{tags:ifempty(No tags on this book)}
```

shows the tags for a book, if any. If it has no tags then it show *No tags on this book*.

The following functions are usable in Single Function Mode because their first parameter is `value`.

- *capitalize* (die Titelseite 187)`(value)` – returns the `value` with the first letter in upper case and the rest lower case.
- *ceiling* (die Titelseite 185)`(value)` – returns the smallest integer greater than or equal to `value`.
- *cmp* (die Titelseite 204)`(value, y, lt, eq, gt)` – compares `value` and `y` after converting both to numbers.
- *contains* (die Titelseite 205)`(value, pattern, text_if_match, text_if_not_match)` – checks if the value is matched by the regular expression `pattern`
- *date_arithmetic* (die Titelseite 190)`(value, calc_spec, fmt)` – Calculate a new date from `value` using `calc_spec`.

- *encode_for_url* (die Titelseite 208) `(value, use_plus)` – returns the `value` encoded for use in a URL as specified by `use_plus`. The value is first URL-encoded. Next, if `use_plus` is `0` then spaces are replaced by `'+'` (plus) signs. If it is `1` then spaces are replaced by `%20`.
- *floor* (die Titelseite 186) `(value)` – returns the largest integer less than or equal to `value`.
- *format_date* (die Titelseite 191) `(value, format_string)` – format the `value`, which must be a date string, using the `format_string`, returning a string.
- *format_duration* (die Titelseite 193) `(value, template, [largest_unit])` – format the value, a number of seconds, into a string showing weeks, days, hours, minutes, and seconds. If the value is a float then it is rounded to the nearest integer.
- *format_number* (die Titelseite 194) `(value, 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}`.
- *fractional_part* (die Titelseite 186) `(value)` – returns the part of the value after the decimal point.
- *human_readable* (die Titelseite 194) `(value)` – expects the `value` to be a number and returns a string representing that number in KB, MB, GB, etc.
- *ifempty* (die Titelseite 206) `(value, text_if_empty)` – if the `value` is not empty then return that `value`, otherwise return `text_if_empty`.
- *language_strings* (die Titelseite 197) `(value, localize)` – return the language names for the language codes (see here for names and codes[86]) passed in `value`.
- *list_contains* (die Titelseite 199) `(value, 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.
- *list_count* (die Titelseite 200) `(value, separator)` – interprets the value as a list of items separated by `separator` and returns the number of items in the list.
- *list_count_matching* (die Titelseite 200) `(value, pattern, separator)` – interprets `value` as a list of items separated by `separator`, returning the number of items in the list that match the regular expression `pattern`.
- *list_item* (die Titelseite 199) `(value, index, separator)` – interpret the `value` as a list of items separated by `separator`, returning the ‚index'th item.
- *list_sort* (die Titelseite 202) `(value, direction, separator)` – return `value` sorted using a case-insensitive lexical sort.
- *lookup* (die Titelseite 199) `(value, [ pattern, key, ]* else_key)` – The patterns will be checked against the `value` in order
- *lowercase* (die Titelseite 187) `(value)` – returns the `value` in lower case.
- *mod* (die Titelseite 186) `(value, y)` – returns the `floor` of the remainder of `value / y`.
- *rating_to_stars* (die Titelseite 194) `(value, use_half_stars)` – Returns the `value` as string of star (★) characters.
- *re* (die Titelseite 206) `(value, pattern, replacement)` – return the `value` after applying the regular expression.
- *re_group* (die Titelseite 206) `(value, pattern [, template_for_group]*)` – return a string made by applying the regular expression `pattern` to `value` and replacing each matched instance
- *round* (die Titelseite 186) `(value)` – returns the nearest integer to `value`.
- *select* (die Titelseite 200) `(value, key)` – interpret the `value` as a comma-separated list of items with each item having the form `id:id_value` (the calibre `identifier` format).

[86] https://www.loc.gov/standards/iso639-2/php/code_list.php

- *shorten* (die Titelseite 206) `(value, left_chars, middle_text, right_chars)` – Return a shortened version of the `value`
- *str_in_list* (die Titelseite 200) `(value, 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.
- *subitems* (die Titelseite 203) `(value, start_index, end_index)` – This function breaks apart lists of tag-like hierarchical items such as genres.
- *sublist* (die Titelseite 204) `(value, 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`.
- *substr* (die Titelseite 207) `(value, start, end)` – returns the `start`'th through the `end`'th characters of `value`
- *swap_around_articles* (die Titelseite 207) `(value, separator)` – returns the `value` with articles moved to the end, separated by a semicolon.
- *swap_around_comma* (die Titelseite 207) `(value)` – given a `value` of the form `B, A`, return `A B`.
- *switch* (die Titelseite 199) `(value, [patternN, valueN,]+ else_value)` – for each `patternN, valueN` pair, checks if the `value` matches the regular expression `patternN`
- *test* (die Titelseite 207) `(value, text_if_not_empty, text_if_empty)` – return `text_if_not_empty` if the value is not empty, otherwise return `text_if_empty`.
- *titlecase* (die Titelseite 187) `(value)` – returns the `value` in title case.
- *transliterate* (die Titelseite 207) `(value)` – Return a string in a latin alphabet formed by approximating the sound of the words in `value`.
- *uppercase* (die Titelseite 187) `(value)` – returns the `value` 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 ]
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
compare_op      ::= '==' | '!=' | '>=' | '>' | '<=' | '<' |
                    'in' | 'inlist' | 'inlist_field' |
                    '==#' | '!=#' | '>=#' | '>#' | '<=#' | '<#'
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, possibly preceded by blanks or tabs.

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[87].
- 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[88]. 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 `$$`. Using `$` is equivalent to using the *field* (die Titelseite 196) function. Using `$$` is equivalent to using the *raw_field* (die Titelseite 198) 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
```

(Fortsetzung auf der nächsten Seite)

[87] https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html
[88] https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html

(Fortsetzung der vorherigen Seite)

```
        a = 'no';
        b = 'yes'
    fi;
    strcat(a, '-', b)
```

Nested `if` example:

```
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
```

For 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* (die Titelseite 203) 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 repeated code in a template 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`, `inlist`, and `inlist_field`. For the `in`, `inlist`, and `inlist_field` operators, the result of the left hand expression is interpreted as a regular expression pattern. They are true if the value of left-hand regular expression matches the value of the right hand expression. The regular expressions are case-insensitive.

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 `inlist_field` operator is true if the left hand regular expression matches any of the items in the field (column) named by the right hand expression, using the separator defined for the field. NB: the `inlist_field` operator requires the right hand expression to evaluate to a field name, while the `inlist` operator requires the right hand expression to evaluate to a string containing a comma-separated list. Because of this difference, `inlist_field` is substantially faster than `inlist` because no string conversions or list constructions are done.

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.

Beispiele

- `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 $#genre` returns `'1'` if any of the values retrieved from the book's genres match the regular expression `science`, e.g., *Science*, *History of Science*, *Science Fiction* etc., otherwise `''`.
- `program: '^science$' inlist $#genre` returns `'1'` if any of the book's genres exactly match the regular expression `^science$`, e.g., *Science*, otherwise `''`. The genres *History of Science* and *Science Fiction* don't match.
- `program: 'asimov' inlist $authors` returns `'1'` if any author matches the regular expression `asimov`, e.g., *Asimov, Isaac* or *Isaac Asimov*, otherwise `''`.
- `program: 'asimov' inlist_field 'authors'` returns `'1'` if any author matches the regular expression `asimov`, e.g., *Asimov, Isaac* or *Isaac Asimov*, otherwise `''`.
- `program: 'asimov$' inlist_field 'authors'` returns `'1'` if any author matches the regular expression `asimov$`, e.g., *Isaac Asimov*, otherwise `''`. It doesn't match *Asimov, Isaac* because of the `$` anchor in the regular expression.
- `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.

Functions in General Program Mode

See *Referenz zur Vorlagefunktion* (die Titelseite 180) for the list of functions built into the template language.

Notes:

- As opposed to *Single Function Mode* (die Titelseite 163), in General Program Mode you must specify the first parameter `value`.
- All parameters are expression_lists (see the grammar above).

10.3.7 More complex programs in template expressions - Template Program Mode

Template Program Mode (*TPM*) is a blend of *General Program Mode* (die Titelseite 166) and *Single Function Mode* (die Titelseite 163). *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* (die Titelseite 163) because you cannot make reference to another metadata field within a template expression. In *TPM* you can, as the following expression demonstrates:

```
{series:'ifempty($, $#genre)'}
```

Das Beispiel zeigt mehrere Dinge:

- *TPM* is used if the expression begins with `:'` and ends with `'}`. Anything else is assumed to be in *Single Function Mode* (die Titelseite 163).

  If the template contains a prefix and suffix, the expression ends with `'|` where the `|` is the delimiter for the prefix. Example:

```
{series:'ifempty($, $#genre)'|prefix | suffix}
```

- Functions must be given all their arguments. For example, the standard built-in functions must be given the initial parameter `value`.
- The variable `$` is usable as the `value` argument and stands for the value of the field named in the template, `series` in this case.
- Leerzeichen werden ignoriert und können überall innerhalb des Ausdrucks verwendet werden.
- Zeichenkettenkonstanten werden in sich entsprechenden Anführungszeichen eingeschlossen, entweder `'` oder `"`.

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 *]]*. Advice: if your program contains `{` and `}` characters then you should use *General Program Mode*.

10.3.8 Python Template Mode

Python Template Mode (PTM) lets you write templates using native Python and the calibre API[89]. 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.
    # 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:

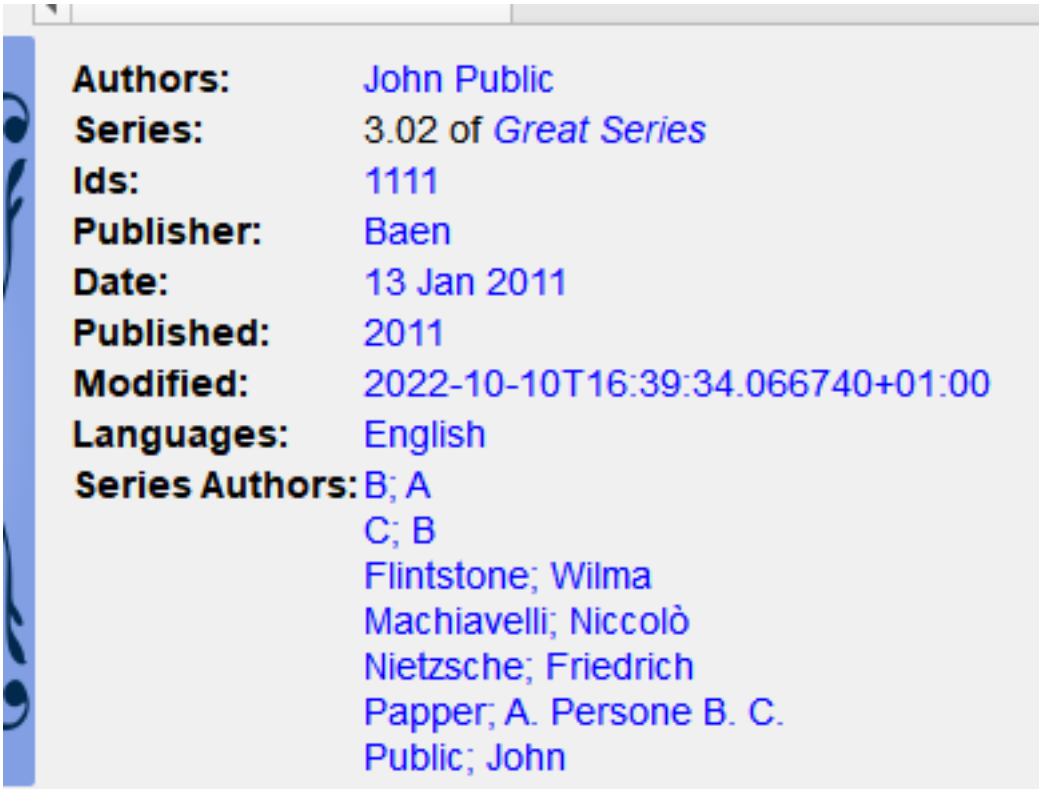
[89] https://manual.calibre-ebook.com/develop.html#api-documentation-for-various-parts-of-calibre

```
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 Templates and URLs

You can use templates to construct URLs. Two cases are described here:

- Custom column *Book details* search URLs
- The calibre URL scheme

Custom column book details search URLs

When you create a custom column you can provide a URL to be used in *Book details* using a template. For example, if you have a custom column for *Translators* you can define a URL to take you to a site for translators. Book details search URLs can be provided for *Text*, *Enumerated*, *Series*, and *Column built from other column* column types.

When an item with a *search template* is clicked in *Book details* the template is evaluated. It is provided the normal book metadata. It is also provided three additional fields:

- `item_value`: the value of the clicked item.
- `item_value_quoted`: the value of clicked item, URL-encoded. Special characters are escaped to make them valid in URLs and spaces are replaced by `'+'` (plus) signs.
- `item_value_no_plus`: the value of clicked item, URL-encoded. Special characters are escaped to make them valid in URLs and spaces are replaced by the `%20`, not plus.

There are several ways to construct the URL. The following use Wikipedia as an example.

The simplest is a basic template:

```
https://en.wikipedia.org/w/index.php?search={item_value_encoded}
```

In some cases you might want to do more processing. There are four template functions you can use, depending on the complexity of the processing.

- *make_url* (die Titelseite 208)`(path, [query_name, query_value]+)` – this function is the easiest way to construct a query URL. It uses a `path`, the web site and page you want to query, and `query_name`, `query_value` pairs from which the query is built. In general, the `query_value` must be URL-encoded. With this function it is always encoded and spaces are always replaced with `'+'` signs.
- *make_url_extended* (die Titelseite 208)`(...)` – this function is similar to *make_url()* (die Titelseite 208) but gives you more control over the URL components. The components of a URL are

  scheme://**authority**/**path**?**query string**.

  See Uniform Resource Locator[90] on Wikipedia for more detail.

  The function has two variants:

  ```
  make_url_extended(scheme, authority, path, [query_name, query_value]+)
  ```

  and

  ```
  make_url_extended(scheme, authority, path, query_string)
  ```

- *query_string* (die Titelseite 209)`([query_name, query_value, how_to_encode]+)` – returns a URL query string constructed from the `query_name, query_value, how_to_encode` triads. A query string is a series of items where each item looks like `query_name=query_value` where `query_value` is URL-encoded as instructed. The query items are separated by `'&'` (ampersand) characters.
- *encode_for_url* (die Titelseite 208)`(value, use_plus)` – returns the `value` encoded for use in a URL as specified by `use_plus`. The value is first URL-encoded. Next, if `use_plus` is `0` then spaces are replaced by `'+'` (plus) signs. If it is `1` then spaces are replaced by `%20`.

[90] https://en.wikipedia.org/wiki/URL

For example, assume you have a custom column *Translators* (`#translators`) where the names are *Last name, First name*. You might need to convert the name to *First name Last name* when creating the URL. You can use the *make_url* (die Titelseite 208) function to do this:

```
program: make_url('https://en.wikipedia.org/w/index.php', 'search', swap_around_comma(
↪$item_value))
```

If we assume that the translator's name is *Boy-Żeleński, Tadeusz* then the above template produces the link:

```
https://en.wikipedia.org/w/index.php?search=Tadeusz+Boy-%C5%BBele%C5%84ski
```

Note that the person's first name is now first, the space is now a plus, and that the non-English characters in the last name are URL-encoded.

The functions *make_url_extended* (die Titelseite 208), *query_string* (die Titelseite 209), and *encode_for_url* (die Titelseite 208) might be useful depending upon any additional processing complexity.

The calibre URL scheme

Calibre supports several different URLs to navigate your calibre libraries. This section shows how to use templates to construct some of the URLs. See *Das calibre:// URL Schema* (die Titelseite 251) for details on the URLs available.

- Switch to a specific library. The syntax of this URL is:

  ```
  calibre://switch-library/Library_Name
  ```

  `Library_Name` must be replaced with the name of the calibre library you wish to open. The library name is shown in the title bar of the window. It is a simple name, not the file path to the library. You must spell it as shown in the title bar, including letter case. The character _ (underscore) stands for the current library. If the name contains any spaces or special characters then it must be hex encoded using the *to_hex* (die Titelseite 210) function, as in the following example:

  ```
  program: strcat('calibre://switch-library/_hex_-', to_hex(current_library_name()))
  ```

  The template generates the URL:

  ```
  calibre://switch-library/_hex_-4c6962726172792e746573745f736d616c6c
  ```

  You can replace the `current_library_name()` function with the actual name of the library, as in:

  ```
  program: strcat('calibre://switch-library/_hex_-', to_hex('Library.test_small'))
  ```

- Links to show books. These links select a book in the calibre library. The syntax for this URL is:

  ```
  calibre://show-book/Library_Name/book_id
  ```

  The `book id` is the numeric calibre id for the book, available to templates as `$id`. As above, the library name might need to be hex encoded. Here is an example:

  ```
  program: strcat('calibre://show-book/_hex_-', to_hex(current_library_name()), '/',
  ↪ $id)
  ```

  It produces the URL:

  ```
  calibre://show-book/_hex_-4c6962726172792e746573745f736d616c6c/1353
  ```

- Searching for books. These links search for books in the specified calibre library. The syntax for this URL is:

```
calibre://search/Library_Name?q=query
calibre://search/Library_Name?eq=hex_encoded_query
```

where *query* is any valid calibre search expression. You must hex encode any query containing spaces or special characters, which generally means all of them. For example, the calibre search expression for searching for a hierarchical tag beginning with ‚AA' is `tags:"=.AA"`. This template constructs a search URL for that expression:

```
program: strcat('calibre://search/_hex_-', to_hex(current_library_name()), '?eq=',
↪ to_hex('tags:"=.AA"'))
```

The resulting URL is:

```
calibre://search/_hex_-4c6962726172792e746573745f736d616c6c?
↪eq=746167733a223d2e414122
```

Here is an example of the same URL built using the :ref:`ff_make_url_extended` function instead of *strcat* (die Titelseite 206):

```
program: make_url_extended('calibre', '', 'search/_hex_-' & to_hex(current_
↪library_name()),
                           'eq', to_hex('tags:"=.AA"'))
```

- Open a book details window on a book in some library. The syntax for this URL is:

```
calibre://book-details/Library_Name/book_id
```

An example template is:

```
program: strcat('calibre://book-details/_hex_-', to_hex(current_library_name()),
↪'/', $id)
```

which produces the URL:

```
calibre://book-details/_hex_-4c6962726172792e746573745f736d616c6c/1353
```

- Open the notes associated with an author/series/etc. The syntax of the URL is:

```
calibre://book-details/Library_Name/Field_Name/id_Item_Id
calibre://book-details/Library_Name/Field_Name/hex_Hex_Encoded_Item_Name
```

`Field_Name` is the lookup name of the field. If the field is a custom column then replace the `#` character with an underscore (_). `Item_Id` is the internal numeric ID of the value in the field. There isn't a template function that returns the `Item_Id`, so templates will normally use the second form, `Hex_Encoded_Item_Name`. Here is a sample template that opens the note for the person `Boy-Żeleński, Tadeusz` in the field `#authtest`:

```
program: strcat('calibre://show-note/_hex_-', to_hex(current_library_name()),
                '/_authtest/hex_', to_hex('Boy-Żeleński, Tadeusz'))
```

which produces the URL:

```
calibre://show-note/_hex_-4c6962726172792e746573745f736d616c6c/_authtest/hex_
↪426f792dc5bb656c65c584736b692c205461646575737a
```

10.3.10 Gespeicherte Vorlagen

Both *General Program Mode* (die Titelseite 166) and *Python Template Mode* (die Titelseite 172) 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.

10.3.11 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,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                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`.

10.3.12 Notes on the difference between modes

The three program modes, *Single Function Mode* (die Titelseite 163) (SFM), *Template Program Mode* (die Titelseite 171) (*TPM*), and *General Program Mode* (die Titelseite 166) (*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* (die Titelseite 171) 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*.

10.3.13 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. Note that you can use *Python Templates* for a similar purpose. As calling user-defined functions is faster than calling a Python template, user-defined functions might be more efficient depending on the complexity of what the function or template does.

10.3.14 Special notes for using templates in different contexts

In the GUI (*Columns made from other columns* and *Template searches*):

- GPM templates work as before.
- Python templates have full access to the calibre database.

In icon rules:

- icon rule templates have no book data so field-based functions such as *format_date_field* (die Titelseite 192), *list_count_field* (die Titelseite 200), and *check_yes_no* (die Titelseite 205) won't work.

In the Content server:

- Templates have access to the new API but not the old API (LibraryDatabase).
- Because of the above, the following formatter functions are not guaranteed to work in GPM templates (composite columns, icon rules, etc) and should be avoided if you use the content server:
  - *connected_device_name* (die Titelseite 195)
  - *connected_device_uuid* (die Titelseite 195)
  - *current_virtual_library_name* (die Titelseite 196)
  - *is_marked* (die Titelseite 197)
  - *virtual_libraries* (die Titelseite 198)

10.3.15 Besondere Hinweise für Speicher-/Sendevorlagen

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.

Um das zu erreichen:

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.

10.3.16 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.

10.3.17 Referenz zur Vorlagefunktion

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.

- *Andere* (die Titelseite 184)
  - *arguments* (die Titelseite 184)
  - *assign* (die Titelseite 184)
  - *globals* (die Titelseite 185)
  - *is_dark_mode* (die Titelseite 185)
  - *print* (die Titelseite 185)
  - *set_globals* (die Titelseite 185)
- *Arithmetik* (die Titelseite 185)

    - *add* (die Titelseite 185)
    - *ceiling* (die Titelseite 185)
    - *divide* (die Titelseite 185)
    - *floor* (die Titelseite 186)
    - *fractional_part* (die Titelseite 186)
    - *mod* (die Titelseite 186)
    - *multiply* (die Titelseite 186)
    - *round* (die Titelseite 186)
    - *subtract* (die Titelseite 186)
- *Boolesch* (die Titelseite 186)
    - *and* (die Titelseite 186)
    - *not* (die Titelseite 186)
    - *or* (die Titelseite 187)
- *Case changes* (die Titelseite 187)
    - *capitalize* (die Titelseite 187)
    - *lowercase* (die Titelseite 187)
    - *titlecase* (die Titelseite 187)
    - *uppercase* (die Titelseite 187)
- *Database functions* (die Titelseite 187)
    - *book_count* (die Titelseite 187)
    - *book_values* (die Titelseite 188)
    - *extra_file_modtime* (die Titelseite 188)
    - *extra_file_names* (die Titelseite 188)
    - *extra_file_size* (die Titelseite 189)
    - *get_link* (die Titelseite 189)
    - *get_note* (die Titelseite 189)
    - *has_extra_files* (die Titelseite 190)
    - *has_note* (die Titelseite 190)
- *Datumsfunktionen* (die Titelseite 190)
    - *date_arithmetic* (die Titelseite 190)
    - *days_between* (die Titelseite 191)
    - *today* (die Titelseite 191)
- *Formatierungswerte* (die Titelseite 191)
    - *finish_formatting* (die Titelseite 191)
    - *format_date* (die Titelseite 191)

- *format_date_field* (die Titelseite 192)
- *format_duration* (die Titelseite 193)
- *format_number* (die Titelseite 194)
- *human_readable* (die Titelseite 194)
- *rating_to_stars* (die Titelseite 194)

- *Get values from metadata* (die Titelseite 194)
  - *annotation_count* (die Titelseite 194)
  - *approximate_formats* (die Titelseite 194)
  - *author_links* (die Titelseite 194)
  - *author_sorts* (die Titelseite 195)
  - *booksize* (die Titelseite 195)
  - *connected_device_name* (die Titelseite 195)
  - *connected_device_uuid* (die Titelseite 195)
  - *current_library_name* (die Titelseite 195)
  - *current_library_path* (die Titelseite 195)
  - *current_virtual_library_name* (die Titelseite 196)
  - *field* (die Titelseite 196)
  - *formats_modtimes* (die Titelseite 196)
  - *formats_path_segments* (die Titelseite 196)
  - *formats_paths* (die Titelseite 197)
  - *formats_sizes* (die Titelseite 197)
  - *has_cover* (die Titelseite 197)
  - *is_marked* (die Titelseite 197)
  - *language_codes* (die Titelseite 197)
  - *language_strings* (die Titelseite 197)
  - *ondevice* (die Titelseite 198)
  - *raw_field* (die Titelseite 198)
  - *raw_list* (die Titelseite 198)
  - *series_sort* (die Titelseite 198)
  - *user_categories* (die Titelseite 198)
  - *virtual_libraries* (die Titelseite 198)

- *Iterate over values* (die Titelseite 198)
  - *first_non_empty* (die Titelseite 198)
  - *lookup* (die Titelseite 199)
  - *switch* (die Titelseite 199)

  - *switch_if* (die Titelseite 199)
- *List lookup* (die Titelseite 199)
  - *identifier_in_list* (die Titelseite 199)
  - *list_contains* (die Titelseite 199)
  - *list_item* (die Titelseite 199)
  - *select* (die Titelseite 200)
  - *str_in_list* (die Titelseite 200)
- *List manipulation* (die Titelseite 200)
  - *list_count* (die Titelseite 200)
  - *list_count_field* (die Titelseite 200)
  - *list_count_matching* (die Titelseite 200)
  - *list_difference* (die Titelseite 201)
  - *list_equals* (die Titelseite 201)
  - *list_intersection* (die Titelseite 201)
  - *list_join* (die Titelseite 201)
  - *list_re* (die Titelseite 202)
  - *list_re_group* (die Titelseite 202)
  - *list_remove_duplicates* (die Titelseite 202)
  - *list_sort* (die Titelseite 202)
  - *list_split* (die Titelseite 202)
  - *list_union* (die Titelseite 202)
  - *range* (die Titelseite 203)
  - *subitems* (die Titelseite 203)
  - *sublist* (die Titelseite 204)
- *Rekursion* (die Titelseite 204)
  - *eval* (die Titelseite 204)
  - *template* (die Titelseite 204)
- *Relational* (die Titelseite 204)
  - *cmp* (die Titelseite 204)
  - *first_matching_cmp* (die Titelseite 204)
  - *strcmp* (die Titelseite 205)
  - *strcmpcase* (die Titelseite 205)
- *String manipulation* (die Titelseite 205)
  - *character* (die Titelseite 205)
  - *check_yes_no* (die Titelseite 205)

- *contains* (die Titelseite 205)
- *field_exists* (die Titelseite 206)
- *ifempty* (die Titelseite 206)
- *re* (die Titelseite 206)
- *re_group* (die Titelseite 206)
- *shorten* (die Titelseite 206)
- *strcat* (die Titelseite 206)
- *strcat_max* (die Titelseite 207)
- *strlen* (die Titelseite 207)
- *substr* (die Titelseite 207)
- *swap_around_articles* (die Titelseite 207)
- *swap_around_comma* (die Titelseite 207)
- *test* (die Titelseite 207)
- *transliterate* (die Titelseite 207)

• *URL functions* (die Titelseite 208)
- *encode_for_url* (die Titelseite 208)
- *make_url* (die Titelseite 208)
- *make_url_extended* (die Titelseite 208)
- *query_string* (die Titelseite 209)
- *to_hex* (die Titelseite 210)
- *urls_from_identifiers* (die Titelseite 210)

• *API of the Metadata objects* (die Titelseite 210)

Andere

arguments

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 with the supplied names, the `id``s, making them effectively parameters. The variables are positional; they get the value of the argument given in the call in the same position. If the corresponding argument 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

class calibre.utils.formatter_functions.**BuiltinAssign**

`assign(id, value)` – assigns `value` to `id`, then returns `value`. `id` must be an identifier, not an expression. In most cases you can use the = operator instead of this function.

globals

class calibre.utils.formatter_functions.**BuiltinGlobals**

`globals(id[=expression] [, id[=expression]]*)` – Retrieves „global variables“ that can be passed into the formatter. The name `id` is the name of the global variable. It both declares and initializes local variables with the names of the global variables passed in (the `id` parameters. If the corresponding variable is not provided in the 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.)

is_dark_mode

class calibre.utils.formatter_functions.**BuiltinIsDarkMode**

`is_dark_mode()` – returns `'1'` if calibre is running in dark mode, `''` (the empty string) otherwise. This function can be used in advanced color and icon rules to choose different colors/icons according to the mode. Example:

```
if is_dark_mode() then 'dark.png' else 'light.png' fi
```

print

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 into a black hole. The `print` function always returns its first argument.

set_globals

class calibre.utils.formatter_functions.**BuiltinSetGlobals**

`set_globals(id[=expression] [, id[=expression]]*)` – Sets *globalvariables* 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.

Arithmetik

add

class calibre.utils.formatter_functions.**BuiltinAdd**

`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.

ceiling

class calibre.utils.formatter_functions.**BuiltinCeiling**

`ceiling(value)` – returns the smallest integer greater than or equal to `value`. Throws an exception if `value` is not a number.

divide

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 usually be replaced by the `/` operator.

floor

class calibre.utils.formatter_functions.**BuiltinFloor**

`floor(value)` – returns the largest integer less than or equal to `value`. Throws an exception if `value` is not a number.

fractional_part

class calibre.utils.formatter_functions.**BuiltinFractionalPart**

`fractional_part(value)` – returns the part of the value after the decimal point. For example, `fractional_part(3.14)` returns `0.14`. Throws an exception if `value` is not a number.

mod

class calibre.utils.formatter_functions.**BuiltinMod**

`mod(value, y)` – returns the `floor` of the remainder of `value / y`. Throws an exception if either `value` or `y` is not a number.

multiply

class calibre.utils.formatter_functions.**BuiltinMultiply**

`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.

round

class calibre.utils.formatter_functions.**BuiltinRound**

`round(value)` – returns the nearest integer to `value`. Throws an exception if `value` is not a number.

subtract

class calibre.utils.formatter_functions.**BuiltinSubtract**

`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.

Boolesch

and

class calibre.utils.formatter_functions.**BuiltinAnd**

`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 when 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.

not

class calibre.utils.formatter_functions.**BuiltinNot**

`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.

or

class calibre.utils.formatter_functions.**BuiltinOr**

`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.

Case changes

capitalize

class calibre.utils.formatter_functions.**BuiltinCapitalize**

`capitalize(value)` – returns the `value` with the first letter in upper case and the rest lower case.

lowercase

class calibre.utils.formatter_functions.**BuiltinLowercase**

`lowercase(value)` – returns the `value` in lower case.

titlecase

class calibre.utils.formatter_functions.**BuiltinTitlecase**

`titlecase(value)` – returns the `value` in title case.

uppercase

class calibre.utils.formatter_functions.**BuiltinUppercase**

`uppercase(value)` – returns the `value` in upper case.

Database functions

book_count

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 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:

- 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:
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
            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.

- 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.

This function can be used only in the GUI and the content server.

book_values

class calibre.utils.formatter_functions.**BuiltinBookValues**

`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. This function can be used only in the GUI and the content server.

extra_file_modtime

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`. The modtime is formatted according to `format_string` (see *format_date()* (die Titelseite 191) 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()* (die Titelseite 190), *extra_file_names()* (die Titelseite 188) and *extra_file_size()* (die Titelseite 189). The epoch is OS dependent. This function can be used only in the GUI and the content server.

extra_file_names

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. See also the functions *has_extra_files()* (die Titelseite 190), *extra_file_modtime()* (die Titelseite 188) and *extra_file_size()* (die Titelseite 189). This function can be used only in the GUI and the content server.

extra_file_size

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`. See also the functions *has_extra_files()* (die Titelseite 190), *extra_file_names()* (die Titelseite 188) and *extra_file_modtime()* (die Titelseite 188). This function can be used only in the GUI and the content server.

get_link

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 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 = '';
    for t in $tags:
        l = get_link('tags', t);
        if l then
            ans = list_join(', ', ans, ',', t & ':' & get_link('tags', t), ',')
        fi
    rof;
ans
```

This function works only in the GUI and the content server.

get_note

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 including images. If `plain_text` is `1` (or `'1'`), return the note's plain text. If the note doesn't exist, return the empty string in both cases. Example:

- Return the HTML of the note attached to the tag *Fiction*:

```
program:
    get_note('tags', 'Fiction', '')
```

- Return the plain text of the note attached to the author *Isaac Asimov*:

```
program:
    get_note('authors', 'Isaac Asimov', 1)
```

This function works only in the GUI and the content server.

has_extra_files

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. See also the functions *extra_file_names()* (die Titelseite 188), *extra_file_size()* (die Titelseite 189) and *extra_file_modtime()* (die Titelseite 188). This function can be used only in the GUI and the content server.

has_note

class calibre.utils.formatter_functions.**BuiltinHasNote**

`has_note(field_name, field_value)`. Check if a field has a note. This function has two variants:

- if `field_value` is not `''` (the empty string) return `'1'` if the value `field_value` in the field `field_name` has a note, otherwise `''`.

  Example: `has_note('tags', 'Fiction')` returns `'1'` if the tag `fiction` has an attached note, otherwise `''`.

- If `field_value` is `''` then return a list of values in `field_name` that have a note. If no item in the field has a note, return `''`. This variant is useful for showing column icons if any value in the field has a note, rather than a specific value.

  Example: `has_note('authors', '')` returns a list of authors that have notes, or `''` if no author has a note.

You can test if all the values in `field_name` have a note by comparing the list length of this function's return value against the list length of the values in `field_name`. Example:

```
list_count(has_note('authors', ''), '&') ==# list_count_field('authors')
```

This function works only in the GUI and the content server.

Datumsfunktionen

date_arithmetic

class calibre.utils.formatter_functions.**BuiltinDateArithmetic**

`date_arithmetic(value, calc_spec, fmt)` – Calculate a new date from `value` 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

class calibre.utils.formatter_functions.**BuiltinDaysBetween**

`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.

today

class calibre.utils.formatter_functions.**BuiltinToday**

`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 ISO[91] date/time format.

Formatierungswerte

finish_formatting

class calibre.utils.formatter_functions.**BuiltinFinishFormatting**

`finish_formatting(value, format, prefix, suffix)` – apply the `format`, `prefix`, and `suffix` to the `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", " - ", " - ")
```

Another example: for the template:

```
{series:re(([^\s])[^\s]+(\s|$),\1)}{series_index:0>2s| - | - }{title}
```

use:

```
program:
    strcat(
        re(field('series'), '([^\s])[^\s]+(\s|$)', '\1'),
        finish_formatting(field('series_index'), '0>2s', ' - ', ' - '),
        field('title')
    )
```

format_date

class calibre.utils.formatter_functions.**BuiltinFormatDate**

`format_date(value, format_string)` – format the `value`, which must be a date string, using the `format_string`, returning a string. It is best if the date is in ISO format as using other date formats often causes errors because the actual date value cannot be unambiguously determined. Note that the `format_date_field()` function is both faster and more reliable.

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)

[91] https://en.wikipedia.org/wiki/ISO_8601

- `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 lowercase localized string for am or pm
- `AP` : use a 12-hour clock instead of a 24-hour clock, with ‚AP‘ replaced by the uppercase 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
- `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:

```
format_date(val, '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 to contain `MMMM`. Using `format_date_field()` avoids this problem.

format_date_field

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 a date field, either standard or custom. See *format_date()* (die Titelseite 191) 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 is also more reliable because it works directly on the underlying date. 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')
```

format_duration

class calibre.utils.formatter_functions.**BuiltinFormatDuration**

`format_duration(value, template, [largest_unit])` – format the value, a number of seconds, into a string showing weeks, days, hours, minutes, and seconds. If the value is a float then it is rounded to the nearest integer. You choose how to format the value using a template consisting of value selectors surrounded by `[` and `]` characters. The selectors are:

- `[w]`: weeks
- `[d]`: days
- `[h]`: hours
- `[m]`: minutes
- `[s]`: seconds

You can put arbitrary text between selectors.

The following examples use a duration of 2 days (172,800 seconds) 1 hour (3,600 seconds) and 20 seconds, which totals to 176,420 seconds.

- `format_duration(176420, '[d][h][m][s]')` will return the value `2d 1h 0m 20s`.
- `format_duration(176420, '[h][m][s]')` will return the value `49h 0m 20s`.
- `format_duration(176420, 'Your reading time is [d][h][m][s]')` returns the value `Your reading time is 49h 0m 20s`.
- `format_duration(176420, '[w][d][h][m][s]')` will return the value `2d 1h 0m 20s`. Note that the zero weeks value is not returned.

If you want to see zero values for items such as weeks in the above example, use an uppercase selector. For example, the following uses `'W'` to show zero weeks:

`format_duration(176420, '[W][d][h][m][s]')` returns `0w 2d 1h 0m 20s`.

By default the text following a value is the selector followed by a space. You can change that to whatever text you want. The format for a selector with your text is the selector followed by a colon followed by text segments separated by `'|'` characters. You must include any space characters you want in the output.

You can provide from one to three text segments.

- If you provide one segment, as in `[w: weeks ]` then that segment is used for all values.
- If you provide two segments, as in `[w: weeks | week ]` then the first segment is used for 0 and more than 1. The second segment is used for 1.
- If you provide three segments, as in `[w: weeks | week | weeks ]` then the first segment is used for 0, the second segment is used for 1, and the third segment is used for more than 1.

The second form is equivalent to the third form in many languages.

For example, the selector:

- `[w: weeks | week | weeks ]` produces `'0 weeks '`, `'1 week '`, or `'2 weeks '`.
- `[w: weeks | week ]` produces `'0 weeks '`, `'1 week '`, or `'2 weeks '`.
- `[w: weeks ]` produces `0 weeks '`, `1 weeks '`, or `2 weeks '`.

The optional `largest_unit` parameter specifies the largest of weeks, days, hours, minutes, and seconds that will be produced by the template. It must be one of the value selectors. This can be useful to truncate a value.

`format_duration(176420, '[h][m][s]', 'd')` will return the value `1h 0m 20s` instead of `49h 0m 20s`.

format_number

class calibre.utils.formatter_functions.**BuiltinFormatNumber**

`format_number(value, 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[92] and the Python[93] documentation for more examples. Returns the empty string if formatting fails.

human_readable

class calibre.utils.formatter_functions.**BuiltinHumanReadable**

`human_readable(value)` – expects the `value` to be a number and returns a string representing that number in KB, MB, GB, etc.

rating_to_stars

class calibre.utils.formatter_functions.**BuiltinRatingToStars**

`rating_to_stars(value, use_half_stars)` – Returns the `value` 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.

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 and the content server.

approximate_formats

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

`approximate_formats()` – return a comma-separated list of formats associated with the book. Because the list comes from calibre's database instead of the file system, there is no guarantee that the list is correct, although it probably is. Note that resulting format names are always uppercase, as in EPUB. The `approximate_formats()` function is much faster than the `formats_...` functions.

This function works only in the GUI. If you want to use these values in save-to-disk or send-to-device templates then you must make a custom „Column built from other columns“, use the function in that column's template, and use that column's value in your save/send templates.

author_links

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

`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.

[92] https://manual.calibre-ebook.com/template_lang.html
[93] https://docs.python.org/3/library/string.html#formatstrings

An author is separated from its link value by the `val_separator` string with no added spaces. Assuming the `val_separator` is a colon, `author:link value` pairs are separated by the `pair_separator` string argument with no added spaces. It is up to you to choose separators that do not occur in author names or links. An author is included even if the author link is empty.

author_sorts

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

`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, which can be 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.

booksize

class calibre.utils.formatter_functions.**BuiltinBooksize**

`booksize()` – returns the value of the calibre `size` field. Returns ‚' if the book has no formats.

This function works only in the GUI. If you want to use this value in save-to-disk or send-to-device templates then you must make a custom „Column built from other columns", use the function in that column's template, and use that column's value in your save/send templates

connected_device_name

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceName**

`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

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceUUID**

`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

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**

`current_library_name()` – return the last name on the path to the current calibre library.

current_library_path

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryPath**

`current_library_path()` – return the full path to the current calibre library.

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()
```

This function works only in the GUI.

field

class calibre.utils.formatter_functions.**BuiltinField**

`field(lookup_name)` – returns the value of the metadata field with lookup name `lookup_name`. The `$` prefix can be used instead of the function, as in `$tags`.

formats_modtimes

class calibre.utils.formatter_functions.**BuiltinFormatsModtimes**

`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()* (die Titelseite 191) function for details. You can use the *select()* (die Titelseite 200) function to get the modification time for a specific format. Note that format names are always uppercase, as in EPUB.

formats_path_segments

class calibre.utils.formatter_functions.**BuiltinFormatsPathSegments**

`formats_path_segments(with_author, with_title, with_format, with_ext, sep)` – return parts of the path to a book format in the calibre library separated by `sep`. The parameter `sep` should usually be a slash (`'/'`). One use is to be sure that paths generated in Save to disk and Send to device templates are shortened consistently. Another is to be sure the paths on the device match the paths in the calibre library.

A book path consists of 3 segments: the author, the title including the calibre database id in parentheses, and the format (author - title). Calibre can shorten any of the three because of file name length limitations. You choose which segments to include by passing `1` for that segment. If you don't want a segment then pass `0` or the empty string for that segment. For example, the following returns just the format name without the extension:

```
formats_path_segments(0, 0, 1, 0, '/')
```

Because there is only one segment the separator is ignored.

If there are multiple formats (multiple extensions) then one of the extensions will be picked at random. If you care about which extension is used then get the path without the extension then add the desired extension to it.

Examples: Assume there is a book in the calibre library with an epub format by Joe Blogs with title ‚Help'. It would have the path

```
Joe Blogs/Help - (calibre_id)/Help - Joe Blogs.epub
```

The following shows what is returned for various parameters:

- `formats_path_segments(0, 0, 1, 0, '/')` returns *Help - Joe Blogs*

- `formats_path_segments(0, 0, 1, 1, '/')` returns *Help - Joe Blogs.epub*
- `formats_path_segments(1, 0, 1, 1, '/')` returns *Joe Blogs/Help - Joe Blogs.epub*
- `formats_path_segments(1, 0, 1, 0, '/')` returns *Joe Blogs/Help - Joe Blogs*
- `formats_path_segments(0, 1, 0, 0, '/')` returns *Help - (calibre_id)*

formats_paths

class calibre.utils.formatter_functions.**BuiltinFormatsPaths**

`formats_paths([separator])` – return a `separator`-separated list of colon-separated items `FMT:PATH` giving the full path to the formats of a book. The `separator` argument is optional. If not supplied then the separator is `', '` (comma space). If the separator is a comma then 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

class calibre.utils.formatter_functions.**BuiltinFormatsSizes**

`formats_sizes()` – return a comma-separated list of colon-separated `FMT:SIZE` items giving the sizes of the formats of a book in bytes. You can use the `select()` function to get the size for a specific format. Note that format names are always uppercase, as in EPUB.

has_cover

class calibre.utils.formatter_functions.**BuiltinHasCover**

`has_cover()` – return `'Yes'` if the book has a cover, otherwise the empty string.

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'` (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

class calibre.utils.formatter_functions.**BuiltinLanguageCodes**

`language_codes(lang_strings)` – return the language codes[94] 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.

language_strings

class calibre.utils.formatter_functions.**BuiltinLanguageStrings**

`language_strings(value, localize)` – return the language names for the language codes (see here for names and codes[95]) passed in `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.

[94] https://www.loc.gov/standards/iso639-2/php/code_list.php
[95] https://www.loc.gov/standards/iso639-2/php/code_list.php

ondevice

class calibre.utils.formatter_functions.**BuiltinOndevice**

`ondevice()` – return the string `'Yes'` if `ondevice` is set, otherwise return the empty string. This function works only in the GUI. If you want to use this value in save-to-disk or send-to-device templates then you must make a custom „Column built from other columns“, use the function in that column's template, and use that column's value in your save/send templates.

raw_field

class calibre.utils.formatter_functions.**BuiltinRawField**

`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`). The `$$` prefix can be used instead of the function, as in `$$pubdate`.

raw_list

class calibre.utils.formatter_functions.**BuiltinRawList**

`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`.

series_sort

class calibre.utils.formatter_functions.**BuiltinSeriesSort**

`series_sort()` – returns the series sort value.

user_categories

class calibre.utils.formatter_functions.**BuiltinUserCategories**

`user_categories()` – return a comma-separated list of the user categories that contain this book. This function works only in the GUI. If you want to use these values in save-to-disk or send-to-device templates then you must make a custom *Column built from other columns*, use the function in that column's template, and use that column's value in your save/send templates

virtual_libraries

class calibre.utils.formatter_functions.**BuiltinVirtualLibraries**

`virtual_libraries()` – return a comma-separated list of Virtual libraries that contain this book. This function works only in the GUI. If you want to use these values in save-to-disk or send-to-device templates then you must make a custom „Column built from other columns“, use the function in that column's template, and use that column's value in your save/send templates.

Iterate over values

first_non_empty

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

class calibre.utils.formatter_functions.**BuiltinLookup**

`lookup(value, [ 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 also the *switch()* (die Titelseite 199) function.

switch

class calibre.utils.formatter_functions.**BuiltinSwitch**

`switch(value, [patternN, valueN,]+ else_value)` – for each `patternN, valueN` pair, checks if the `value` matches the regular expression `patternN` and if so returns the associated `valueN`. If no `patternN` matches, then `else_value` is returned. You can have as many `patternN, valueN` pairs as you wish. The first match is returned.

switch_if

class calibre.utils.formatter_functions.**BuiltinSwitchIf**

`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.

List lookup

identifier_in_list

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 (`''`).

list_contains

class calibre.utils.formatter_functions.**BuiltinInList**

`list_contains(value, 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.

Aliases: in_list(), list_contains()

list_item

class calibre.utils.formatter_functions.**BuiltinListitem**

`list_item(value, index, separator)` – interpret the `value` as a list of items separated by `separator`, returning the ‚index'th 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. The separator has the same meaning as in the count function, usually comma but is ampersand for author-like lists.

select

class calibre.utils.formatter_functions.**BuiltinSelect**

`select(value, key)` – interpret the `value` as a comma-separated list of items with each item having the form `id:id_value` (the calibre `identifier` format). The function finds the first pair with the id equal to `key` and returns the corresponding `id_value`. If no id matches then the function returns the empty string.

str_in_list

class calibre.utils.formatter_functions.**BuiltinStrInList**

`str_in_list(value, 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.

List manipulation

list_count

class calibre.utils.formatter_functions.**BuiltinCount**

`list_count(value, 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:list_count(,)}`, `{authors:list_count(&)}`.

Aliases: `count()`, `list_count()`

list_count_field

class calibre.utils.formatter_functions.**BuiltinFieldListCount**

`list_count_field(lookup_name)` – returns the count of items in the field with the lookup name `lookup_name`. The field must be multi-valued such as `authors` or `tags`, otherwise the function raises an error. This function is much faster than `list_count()` because it operates directly on calibre data without converting it to a string first. Example: `list_count_field('tags')`.

list_count_matching

class calibre.utils.formatter_functions.**BuiltinListCountMatching**

`list_count_matching(value, pattern, separator)` – interprets `value` 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

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

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

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

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, ',')
```

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

class calibre.utils.formatter_functions.**BuiltinListRe**

`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

class calibre.utils.formatter_functions.**BuiltinListReGroup**

`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

class calibre.utils.formatter_functions.**BuiltinListRemoveDuplicates**

`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

class calibre.utils.formatter_functions.**BuiltinListSort**

`list_sort(value, direction, separator)` – return `value` sorted using a case-insensitive lexical sort. If `direction` is zero (number or character), `value` is sorted ascending, otherwise descending. The list items are separated by `separator`, as are the items in the returned list.

list_split

class calibre.utils.formatter_functions.**BuiltinListSplit**

`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.

Example:

```
list_split('one:two:foo', ':', 'var')
```

is equivalent to:

```
var_0 = 'one'
    var_1 = 'two'
    var_2 = 'foo'
```

list_union

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: `merge_lists()`, `list_union()`

range

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 = 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

class calibre.utils.formatter_functions.**BuiltinSubitems**

`subitems(value, 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.

Examples:

- 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

class calibre.utils.formatter_functions.**BuiltinSublist**

`sublist(value, 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“
- `{tags:sublist(-1,0,\,)}` returns „C“
- `{tags:sublist(0,-1,\,)}` returns „A, B“

Rekursion

eval

class calibre.utils.formatter_functions.**BuiltinEval**

`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[96], 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.

template

class calibre.utils.formatter_functions.**BuiltinTemplate**

`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. If not using General Program Mode, because the `{` and `}` characters are special, you must use `[[` for the `{` character and `]]` for the } character; they are converted automatically. For example, `template(\'[[title_sort]]\')` will evaluate the template `{title_sort}` and return its value. Note also that prefixes and suffixes (the `|prefix|suffix` syntax) cannot be used in the argument to this function when using template program mode.

Relational

cmp

class calibre.utils.formatter_functions.**BuiltinCmp**

`cmp(value, y, lt, eq, gt)` – compares `value` and `y` after converting both to numbers. Returns `lt` if `value <# y`, `eq` if `value ==# y`, otherwise `gt`. This function can usually be replaced with one of the numeric compare operators (`==#`, `<#`, `>#`, etc).

first_matching_cmp

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

[96] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

`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"`.

strcmp

class calibre.utils.formatter_functions.**BuiltinStrcmp**

`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

class calibre.utils.formatter_functions.**BuiltinStrcmpcase**

`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.

String manipulation

character

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`. This function is used to put these characters into the output of templates.

check_yes_no

class calibre.utils.formatter_functions.**BuiltinCheckYesNo**

`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.

contains

class calibre.utils.formatter_functions.**BuiltinContains**

`contains(value, 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_not_match`.

field_exists

class calibre.utils.formatter_functions.**BuiltinFieldExists**

`field_exists(lookup_name)` – checks if a field (column) with the lookup name `lookup_name` exists, returning `'1'` if so and the empty string if not.

ifempty

class calibre.utils.formatter_functions.**BuiltinIfempty**

`ifempty(value, text_if_empty)` – if the `value` is not empty then return that `value`, otherwise return `text_if_empty`.

re

class calibre.utils.formatter_functions.**BuiltinRe**

`re(value, 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[97].

re_group

class calibre.utils.formatter_functions.**BuiltinReGroup**

`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[98], 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* )(.*)", "{$:uppercase()}", "{$}")'}
```

shorten

class calibre.utils.formatter_functions.**BuiltinShorten**

`shorten(value, 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 inthe 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 *TheDome* would not be changed.

strcat

class calibre.utils.formatter_functions.**BuiltinStrcat**

`strcat(a [, b]*)` – returns a string formed by concatenating all the arguments. Can take any number of arguments. In most cases you can use the `&` operator instead of this function.

[97] https://docs.python.org/3/library/re.html
[98] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

strcat_max

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`. 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.

strlen

class calibre.utils.formatter_functions.**BuiltinStrlen**

`strlen(value)` – Returns the length of the string `value`.

substr

class calibre.utils.formatter_functions.**BuiltinSubstr**

`substr(value, start, end)` – returns the `start`'th through the `end`'th characters of `value`. The first character in `value` 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'`.

swap_around_articles

class calibre.utils.formatter_functions.**BuiltinSwapAroundArticles**

`swap_around_articles(value, separator)` – returns the `value` with articles moved to the end, separated by a semicolon. 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 or the separator is the empty string 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

class calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

`swap_around_comma(value)` – 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.

test

class calibre.utils.formatter_functions.**BuiltinTest**

`test(value, text_if_not_empty, text_if_empty)` – return `text_if_not_empty` if the value is not empty, otherwise return `text_if_empty`.

transliterate

class calibre.utils.formatter_functions.**BuiltinTransliterate**

`transliterate(value)` – Return a string in a latin alphabet formed by approximating the sound of the words in `value`. For example, if `value` is `Фёдор Михáйлович Достоевский` this function returns `Fiodor Mikhailovich Dostoievskii`.

URL functions

encode_for_url

class calibre.utils.formatter_functions.**BuiltinEncodeForURL**

`encode_for_url(value, use_plus)` – returns the `value` encoded for use in a URL as specified by `use_plus`. The value is first URL-encoded. Next, if `use_plus` is `0` then spaces are replaced by `'+'` (plus) signs. If it is `1` then spaces are replaced by `%20`.

If you do not want the value to be encoding but to have spaces replaced then use the *re()* (die Titelseite 206) function, as in `re($series, ' ', '%20')`

See also the functions *make_url()* (die Titelseite 208), *make_url_extended()* (die Titelseite 208) and *query_string()* (die Titelseite 209).

make_url

class calibre.utils.formatter_functions.**BuiltinMakeUrl**

`make_url(path, [query_name, query_value]+)` – this function is the easiest way to construct a query URL. It uses a `path`, the web site and page you want to query, and `query_name`, `query_value` pairs from which the query is built. In general, the `query_value` must be URL-encoded. With this function it is always encoded and spaces are always replaced with `'+'` signs.

At least one `query_name, query_value` pair must be provided.

Example: constructing a Wikipedia search URL for the author *Niccolò Machiavelli*:

```
make_url('https://en.wikipedia.org/w/index.php', 'search', 'Niccolò Machiavelli')
```

returns

```
https://en.wikipedia.org/w/index.php?search=Niccol%C3%B2+Machiavelli
```

If you are writing a custom column book details URL template then use `$item_name` or `field('item_name')` to obtain the value of the field that was clicked on. Example: if *Niccolò Machiavelli* was clicked then you can construct the URL using:

```
make_url('https://en.wikipedia.org/w/index.php', 'search', $item_name)
```

See also the functions *make_url_extended()* (die Titelseite 208), *query_string()* (die Titelseite 209) and *encode_for_url()* (die Titelseite 208).

make_url_extended

class calibre.utils.formatter_functions.**BuiltinMakeUrlExtended**

`make_url_extended(...)` – this function is similar to *make_url()* (die Titelseite 208) but gives you more control over the URL components. The components of a URL are

scheme://**authority**/**path**?**query string**.

See Uniform Resource Locator[99] on Wikipedia for more detail.

The function has two variants:

[99] https://en.wikipedia.org/wiki/URL

```
make_url_extended(scheme, authority, path, [query_name, query_value]+)
```

and

```
make_url_extended(scheme, authority, path, query_string)
```

This function returns a URL constructed from the `scheme`, `authority`, `path`, and either the `query_string` or a query string constructed from the query argument pairs. The `authority` can be empty, which is the case for `calibre` scheme URLs. You must supply either a `query_string` or at least one `query_name, query_value` pair. If you supply `query_string` and it is empty then the resulting URL will not have a query string section.

Example 1: constructing a Wikipedia search URL for the author *Niccolò Machiavelli*:

```
make_url_extended('https', 'en.wikipedia.org', '/w/index.php', 'search', 'Niccolò␣
↪Machiavelli')
```

returns

```
https://en.wikipedia.org/w/index.php?search=Niccol%C3%B2+Machiavelli
```

See the *query_string()* (die Titelseite 209) function for an example using `make_url_extended()` with a `query_string`.

If you are writing a custom column book details URL template then use `$item_name` or `field('item_name')` to obtain the value of the field that was clicked on. Example: if *Niccolò Machiavelli* was clicked on then you can construct the URL using :

```
make_url_extended('https', 'en.wikipedia.org', '/w/index.php', 'search', $item_name')
```

See also the functions *make_url()* (die Titelseite 208), *query_string()* (die Titelseite 209) and *encode_for_url()* (die Titelseite 208).

query_string

class calibre.utils.formatter_functions.**BuiltinQueryString**

`query_string([query_name, query_value, how_to_encode]+)`– returns a URL query string constructed from the `query_name, query_value, how_to_encode` triads. A query string is a series of items where each item looks like `query_name=query_value` where `query_value` is URL-encoded as instructed. The query items are separated by `'&'` (ampersand) characters.

If `how_to_encode` is `0` then `query_value` is encoded and spaces are replaced with `'+'` (plus) signs. If `how_to_encode` is `1` then `query_value` is encoded with spaces replaced by `%20`. If `how_to_encode` is `2` then `query_value` is returned unchanged; no encoding is done and spaces are not replaced. If you want `query_value` not to be encoded but spaces to be replaced then use the *re()* (die Titelseite 206) function, as in `re($series, ' ', '%20')`

You use this function if you need specific control over how the parts of the query string are constructed. You could then use the resultingquery string in *make_url_extended()* (die Titelseite 208), as in

```
make_url_extended(
       'https', 'your_host', 'your_path',
       query_string('encoded', 'Hendrik Bäßler', 0, 'unencoded', 'Hendrik Bäßler', 2))
```

giving you

```
https://your_host/your_path?encoded=Hendrik+B%C3%A4%C3%9Fler&unencoded=Hendrik Bäßler
```

You must have at least one `query_name, query_value, how_to_encode` triad, but can have as many as you wish.

The returned value is a URL query string with all the specified items, for example: `name1=val1[&nameN=valN]*`. Note that the `'?'` *path / query string* separator is not included in the returned result.

If you are writing a custom column book details URL template then use `$item_name` or `field('item_name')` to obtain the unencoded value of the field that was clicked. You also have `item_value_quoted` where the value is already encoded with plus signs replacing spaces, and `item_value_no_plus` where the value is already encoded with `%20` replacing spaces.

See also the functions *make_url()* (die Titelseite 208), *make_url_extended()* (die Titelseite 208) and *encode_for_url()* (die Titelseite 208).

to_hex

class calibre.utils.formatter_functions.**BuiltinToHex**

`to_hex(val)` – returns the string `val` encoded into hex. This is useful when constructing calibre URLs.

urls_from_identifiers

class calibre.utils.formatter_functions.**BuiltinUrlsFromIdentifiers**

`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.

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=('Unbekannt',)*, *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()* (die Titelseite 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 is in *STANDARD_METADATA_FIELDS* (die Titelseite 212).

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')]
    '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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    # 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(
                           USER_METADATA_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)

# All fields except custom fields
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
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 Alles über die Verwendung von regulären Ausdrücken in Calibre

Reguläre Ausdrücke werden an vielen Stellen in calibre verwendet, um den Inhalt und die Metadaten von E-Büchern auf raffinierte Weise zu manipulieren. Dieses Tutorial ist eine vorsichtige Einführung, mit der man den Umgang mit regulären Ausdrücken erlernen kann.

Inhalt

- *Vorweg ein Wort der Warnung - und der Ermutigung* (die Titelseite 215)
- *Wo in calibre sind reguläre Ausdrücke verwendbar?* (die Titelseite 215)
- *Was um Himmels Willen ist ein „regulärer Ausdruck“?* (die Titelseite 215)
- *Bitte etwas genauer!* (die Titelseite 216)
- *Das hört sich nicht schlecht an. Was kommt als Nächstes?* (die Titelseite 216)

- *Hey, toll! Langsam bekommt alles einen Sinn!* (die Titelseite 216)
- *Tja, das mit den Sonderzeichen ist ja gut und schön. Aber wie kann ich nach einem Punkt oder einem Fragezeichen suchen?* (die Titelseite 217)
- *So, und welche Mengen sind nun am brauchbarsten?* (die Titelseite 217)
- *Aber wenn ich jetzt nach ein paar unterschiedlichen Zeichenfolgen gleichzeitig suchen möchte - wird die Sache dann sehr kompliziert?* (die Titelseite 218)
- *Da fehlt noch …* (die Titelseite 218)
- *Hatten Sie anfangs nicht gesagt, dass reguläre Ausdrücke auch unabhängig von Groß- und Kleinschreibung gemacht werden könnten?* (die Titelseite 218)
- *Mir scheint, ich verstehe allmählich etwas von regulären Ausdrücken … und wie benutze ich sie nun in calibre?* (die Titelseite 219)
  - *Konvertierungen* (die Titelseite 219)
  - *Hinzufügen von Büchern* (die Titelseite 219)
  - *Gleichzeitiges Bearbeiten von Metadaten* (die Titelseite 220)
- *Schnellreferenz* (die Titelseite 220)
- *Danksagung* (die Titelseite 224)

10.4.1 Vorweg ein Wort der Warnung - und der Ermutigung

Nun wird es unvermeidbar ein wenig technisch - immerhin sind reguläre Ausdrücke ein technisches Werkzeug, mit dem man technisches Zeug erledigt. Ich werde den einen oder anderen Fachbegriff verwenden und Gedankengänge, die kompliziert oder auch verwickelt wirken mögen. Ich werde versuchen, diese Gedanken so gut ich kann zu erklären, aber ich werde nicht ohne sie auskommen. Nur sollten Sie sich von diesem Fachchinesisch nicht abschrecken lassen, denn ich habe versucht, alle neuen Begriffe zu erklären. Und obwohl reguläre Ausdrücke an sich wie mysteriöse, schwarze Magie wirken (oder, nüchterner gesagt, wie ein Durcheinander irgendwelcher Buchstaben, Zahlen und Zeichen), verspreche ich hiermit, dass sie gar nicht so kompliziert sind. Selbst diejenigen, die reguläre Ausdrücke wirklich gut verstehen, haben schon mal Probleme, die komplexeren zu durchschauen, aber sie aufzuschreiben ist gar nicht so schwer - Sie bauen den Ausdruck schrittweise zusammen. Also: Treten Sie vor und folgen Sie mir in den Kaninchenbau!

10.4.2 Wo in calibre sind reguläre Ausdrücke verwendbar?

Es gibt nur wenige Stellen, an denen Calibre Reguläre Ausdrücke verwendet. Da wären, die *Search & replace* in den Konvertierungsoptionen, Metadatenerkennung auf Grund von Dateinamen in the Importeinstellungen und Suchen & Ersetzen beim massenhaften Bearbeiten der Metadaten von Büchern. Der Calibre-Bucheditor kann ebenfalls Reguläre Ausdrücke in seiner *Search and replace*-Funktion verwenden. Schließlich kannst du Reguläre Ausdrücke auch beim Suchen in der Calibre-Buchliste und innerhalb des Calibre-E-Book-Betrachters verwenden.

10.4.3 Was um Himmels Willen ist ein „regulärer Ausdruck“?

Reguläre Ausdrücke beschreiben Mengen von Zeichen oder Zeichenfolgen. Ein einzelner regulärer Ausdruck *entspricht* unter Umständen einer ganzen Reihe verschiedener Zeichenfolgen. Und das macht reguläre Ausdrücke so mächtig: Sie beschreiben kurz und knapp eine möglicherweise große Zahl von Varianten.

Bemerkung

Dabei benutze ich den Begriff Zeichenfolge so wie in einer Programmiersprache, nämlich eine Folge von einem oder mehreren Zeichen, wobei als Zeichen die eigentlichen Buchstaben und Zahlen, aber auch Satzzeichen und sogenannten Leerraum (Umbrüche, Tabulatorschritte usw.) gelten. Man beachte, dass meistens Klein- und Großbuchstaben nicht als gleichwertig angesehen werden. Es ist also nicht egal, ob man „a“ oder „A“ schreibt. In der Suchleiste von calibre spielt die Groß- oder Kleinschreibung keine Rolle, wohl aber bei den Einstellungen für die Konvertierung. Es gibt zwar eine Möglichkeit, alle regulären Ausdrücke unabhängig von Klein- und Großbuchstaben zu machen, aber dazu kommen wir später. Es wird kompliziert, weil reguläre Ausdrücke Abweichungen in den Zeichenfolgen zulassen, denen sie entsprechen. Anders gesagt: Ein einziger passt zu verschiedenen Zeichenfolgen - deshalb werden reguläre Ausdrücke ja überhaupt verwendet. Mehr dazu in Kürze.

10.4.4 Bitte etwas genauer!

Klar, darum machen wir das ja! Der wichtigste Gedanke bei regulären Ausdrücken zuerst: *Eine Zeichenfolge an sich ist ein regulärer Ausdruck, der sich selbst entspricht*. Wenn ich also einen regulären Ausdruck für die Zeichenfolge `\"Hallo, Welt!"` suche, kann ich einfach `"Hallo, Welt!"` benutzen. Ja, es wirklich so simpel! Sie sollten beachten, dass dies *nur* für die exakte Zeichenfolge `"Hallo, Welt!"` gilt, nicht etwa für `"Hallo, wElt!"` oder `"hallo, welt!"` oder ähnliche Zeichenfolgen.

10.4.5 Das hört sich nicht schlecht an. Was kommt als Nächstes?

Als Nächstes kommen jetzt die richtig guten Sachen. Sie erinnern sich daran, dass reguläre Ausdrücke zu verschiedenen Zeichenfolgen passen können? Da wird es nun etwas komplizierter. Mal angenommen, als praktisches Beispiel, Sie wollen ein Buch konvertieren mit einer ganz hässlichen Zeilenangabe in der Fußzeile, so etwas wie „Seite 5 von 423“. Offensichtlich steigt die Seitenzahl von 1 bis 423, und Sie bräuchten 423 verschiedene Zeichenfolgen, oder? Falsch: Tatsächlich können Sie mit regulären Ausdrücken Zeichenmengen festlegen, die zueinander passen. Zur Festlegung einer solchen Menge schreiben Sie alle betreffenden Zeichen in eckige Klammern. Zum Beispiel entspricht die Menge `[abc]` entweder dem Zeichen „a“, „b“ oder „c“. *Mengen entsprechen immer nur einem der Zeichen dieser Menge*. Sie „verstehen“ übrigens auch Zeichen-Bereiche, so dass Sie für alle Kleinbuchstaben (ohne Umlaute) die Menge `[a-z]` und für sämtliche Klein- und Großbuchstaben (wieder ohne Umlaute) die Menge `[a-zA-Z]` schreiben können - und so weiter. Soweit klar? Mit dem regulären Ausdruck `Seite [0-9] von 423` hätten Sie also etwas Passendes für die ersten neun Seiten und benötigten nun nur noch drei Ausdrücke: `Seite [0-9][0-9] von 423` wäre der zweite Ausdruck, entsprechend den zweistelligen Seitenzahlen, und den dritten Ausdruck können Sie sicherlich selbst finden. Ja, versuchen Sie es und schreiben Sie ihn auf!

10.4.6 Hey, toll! Langsam bekommt alles einen Sinn!

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!

Bemerkung

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 Tja, das mit den Sonderzeichen ist ja gut und schön. Aber wie kann ich nach einem Punkt oder einem Fragezeichen suchen?

Natürlich geht auch so etwas: Sie brauchen nur einen „Backslash“ vor ein Sonderzeichen zu setzen, und es wird als das Zeichen selbst verstanden, ohne irgendeine Sonderbedeutung. Eine solche Kombination aus einem „Backslash“ mit einem einzelnen Zeichen nennt man Escape-Sequenz, und unter „Escaping“ versteht man, vor ein Sonderzeichen einen „Backslash“ zu schreiben. Solche Escape-Sequenzen gelten als einzelne Elemente. Es gibt auch Escape-Sequenzen, die nicht nur Sonderzeichen betreffen; zum Beispiel ist `"\t"` die Sequenz für einen Tabulatorschritt. Wir kommen später noch zu ähnlichen Escape-Sequenzen. Was die Sonderzeichen betrifft, sollten Sie einfach mal folgendes annehmen: Jedes Zeichen in dieser Einführung, das eine besondere Funktion hat, gilt als Sonderzeichen und benötigt den „Backslash“, wenn man das Zeichen selbst meint.

10.4.8 So, und welche Mengen sind nun am brauchbarsten?

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

Bemerkung

„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.

Bemerkung

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.

Ein letzter Hinweis zu Mengen: Sie können eine Menge auch so festlegen, dass jedes Zeichen gemeint ist *außer* denen in der Menge! Dazu schreiben Sie als allererstes Zeichen ein `"^"` in die Menge. So entspricht `[^a]` jedem Zeichen außer dem „a“. Man spricht auch vom Komplement dieser Menge. Die Abkürzungen mit Hilfe von Escape-Sequenzen funktionieren

auch mit Komplementbildung: `"\D"` bedeutet so viel wie „alles außer Zahlen“, anders ausgedrückt also `[^0-9]`. Der oben erwähnte Ausdruck `<p[^>]*>` versucht, wie man sieht, jedem Zeichen zu entsprechen mit Ausnahme der schließenden spitzen Klammer.

10.4.9 Aber wenn ich jetzt nach ein paar unterschiedlichen Zeichenfolgen gleichzeitig suchen möchte - wird die Sache dann sehr kompliziert?

Keine Angst, das Leben ist noch immer schön und leicht. Betrachten Sie dieses Beispiel: Das Buch, das Sie konvertieren hat „Titel“ auf jeder ungeraden, und „Autor“ auf jeder geraden Seite stehen. Das sieht im Druck toll aus, stimmt's? Nur: Bei Ebooks ist das lästig. Sie können ganze Ausdrücke mit normalen Klammern gruppieren, und das Zeichen `"|"` erlaubt Ihnen *entweder* den Ausdruck links *oder* denjenigen rechts davon zu finden. Kombinieren Sie diese beiden und sie haben es. Das war zu schnell? OK, als Erstes kombinieren wir die Ausdrücke für gerade und ungerade Seiten, erhalten also `(Title)(Author)` für unsere zwei benötigten Ausdrücke. Nun vereinfachen wir das Ganze, indem wir den senkrechten Strich verwenden (`"|"` wird senkrechter Strich oder Verkettungszeichen genannt): Wenn Sie den Ausdruck `(Title|Author)` verwenden finden Sie entweder „Titel“ (auf den ungeraden Seiten) oder „Autor“ (auf den geraden Seiten). Na, war das nicht einfach?

Sie können den senkrechten Strich natürlich auch ohne gruppierende Klammern verwenden. Erinnern Sie sich, dass ich sagte Quantifizierer wiederholten das vorangegangene Element? Nun, der senkrechte Strich funktioniert ein wenig anders: Der Ausdruck „Titel|Autor“ wird also entweder die Zeichenfolge „Titel“ oder die Zeichenfolge „Autor“ finden, genau wie das vorhergehende Beispiel mit der Gruppierung. *Der senkrechte Strich wählt aus zwischen der ganzen Zeichenfolge vor und der nach ihm*. Wollten Sie also die Zeichenfolgen „Calibre“ und „calibre“ finden und dabei nur zwischen dem großen und dem kleinen „c“ wählen, müssten Sie den Ausdruck `(c|C)alibre` verwenden, bei dem die Gruppierung sicherstellt, dass nur das „c“ gewählt wird. Sollten Sie `c|Calibre` verwenden, fänden Sie die Zeichenfolgen „c“ oder „Calibre“, was nicht unsere Absicht war. Kurz gesagt: Verwenden Sie im Zweifel die Gruppierung zusammen mit dem senkrechten Strich.

10.4.10 Da fehlt noch ...

… es gäbe da noch eine letzte, wirklich geschickte Sache, die Sie mit Gruppen anstellen können. Wenn Sie einen Gruppierungsausdruck haben, den Sie zuvor eingesetzt haben, können Sie Referenzen auf diese Gruppe im weiteren Ausdruck verwenden: Gruppen werden beginnend mit 1 nummeriert, und Sie werden referenziert, indem die Nummer der Gruppe mit einer „Escape-Sequenz“ versehen wird, folglich würde die fünfte Gruppe als `\5` referenziert. Daher fänden Sie bei einer Suche nach `([^ ]+) \1` in der Zeichenfolge „Test Test“ die komplette Zeichenfolge!

10.4.11 Hatten Sie anfangs nicht gesagt, dass reguläre Ausdrücke auch unabhängig von Groß- und Kleinschreibung gemacht werden könnten?

Ja, das habe ich. Danke für's Aufpassen. Sie können Calibre sagen, wie es mit bestimmten Dingen umgehen soll, indem Sie sogenannte „Flags“ verwenden. Sie fügen Flags in ihren Ausdruck mithilfe des speziellen Konstrukts `(?Flag hierher)` ein, in welchem Sie selbstverständlich „Flags hierher“ durch die genauen Flags, die Sie wollen, ersetzen würden. Um die Groß-/Kleinschreibung zu ignorieren lautet die Flag `i`, folglich würden Sie `(?i)` in Ihren Ausdruck einfügen. Also würde `(?i)test` mit „Test“, „tEst“, „TEst“ übereinstimmen und jegliche Variation der Groß-/Kleinschreibung, die Ihnen einfällt.

Das Flag `s` sorgt dafür, dass mit dem Punkt jedes beliebige Zeichen gefunden wird, *einschließlich* „Newline“. Wenn du mehrere Flags in einem Ausdruck verwenden möchtest, fasse sie einfach in der selben Anweisung zusammen: `(?is)` würde Groß-/Kleinschreibung ignorieren und der Punkt würde alle Zeichen finden. Es spielt keine Rolle, welches Flag du zuerst angibst: `(?si)` würde das selbe bewirken wie die Kombination oben.

10.4.12 Mir scheint, ich verstehe allmählich etwas von regulären Ausdrücken ... und wie benutze ich sie nun in calibre?

Konvertierungen

Lass uns mit den Umwandlungseinstellungen beginnen. Im Teil *Search & replace* kannst du eine Regexp (kurz für Regular expression) eingeben, der die Zeichenkette beschreibt, die während der Umwandlung ersetzt wird. Der angenehme Teil ist der Assistent. Klicke auf den Zauberstab und du erhälst eine Vorschau dessen, was Calibre während des Umwandlungprozesses „sieht". Scrolle zu der Zeichenkette, die du entfernen möchtest, wähle sie aus, kopiere sie und füge sie in das Regexp-Feld oben im Fenster ein. Wenn es variable Teile gibt, wie Seitenzahlen oder ähnliches, verwende Mengen und Quantifizierer um diese abzudecken und wenn du schon mal dabei bist: Denke daran Sonderzeichen zu maskieren, falls es welche gibt. Drücke auf die Schaltfläche *Test* und Calibre hebt die Teile hervor, die ersetzt würden, wenn du diesen Regexp anwendest. Sobald du zufrieden bist, drücke OK und starte den Prozess. Sei vorsichtig careful wenn deine Quelle mit Tags wie in diesem Beispiel versehen ist:

```
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[100]). 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 `</b>` 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 `</b>`, thus we could remove everything between those tags using `<b.*?>.*?</b>`. But using this expression would be a bad idea, because it removes everything enclosed by <b>- 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.*?</b>` 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.

Hinzufügen von Büchern

Etwas anderes, wofür Sie reguläre Ausdrücke verwenden können, ist das Extrahieren von Metadaten aus Dateinamen. Sie finden diese Funktion im Bereich „Bücher hinzufügen" der Einstellungen.Es gibt hierbei eine Spezialfunktion: Sie können Feldnamen für die Metadaten-Felder vewenden, beispielsweise bedeutete `(?P<title>)`, dass Calibre diesen Teil der Zeichenfolge als Buchtitel einsetzen würde. Die erlaubten Feldnamen sind in den Fenstern aufgelistet, zusammen mit einem weiteren netten Testfeld. Ein Beispiel: Nehmen wir an, Sie wollen einen ganzen Haufen Dateien importieren, die ähnlich wie `Klassische Texte: Die göttliche Komödie von Dante Alighieri.mobi` benannt sind. (Selbstverständlich befindet sich das schon in Ihrer Bibliothek, denn wir alle lieben klassische italienische Dichtkunst.) ... oder `Monumentalwerke der Science-Fiction: Die Foundation-Trilogie von Isaac Asimov.epub`. Dies ist offensichtlich ein Namensschema, aus dem Calibre keinerlei sinnvolle Informationen extrahieren wird – sein Standardausdruck für die Extraktion von Metadaten ist `(?P<title>.+) - (?P<author>[^_]+)`. Ein regulärer Ausdruck, der hier funktionieren würde, wäre `[a-zA-Z]+: (?P<title>.+) von (?P<author>.+)`. Beachten Sie bitte, dass Sie in der Gruppe für das jeweilige Metadaten-Feld Ausdrücke einsetzen müssen, um zu beschreiben was eigentlich auf das Feld passt. Und beachten Sie außerdem, dass Sie beim verwenden von Calibres vorgegebenem Testfeld die Dateiendung zu ihrem Versuchs-Dateinamen hinzufügen müssen, andernfalls werden Sie trotz eines korrekten Ausdrucks nichts finden.

[100] https://www.mobileread.com/forums/showthread.php?t=75594"

Gleichzeitiges Bearbeiten von Metadaten

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[101].

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 Schnellreferenz

Quick reference for regexp syntax

This checklist summarizes the most commonly used/hard to remember parts of the regexp engine available in most parts of calibre.

Inhalt

- *Character classes* (die Titelseite 221)
- *Shorthand character classes* (die Titelseite 221)
- *The quantifiers* (die Titelseite 221)
- *Greed* (die Titelseite 222)
- *Alternation* (die Titelseite 222)
- *Exclusion* (die Titelseite 222)
- *Anchors* (die Titelseite 222)
- *Groups* (die Titelseite 222)
- *Lookarounds* (die Titelseite 223)
- *Rekursion* (die Titelseite 223)
- *Special characters* (die Titelseite 224)
- *Meta-characters* (die Titelseite 224)

[101] https://docs.python.org/library/re.html

- *Modes* (die Titelseite 224)

Character classes

Character classes are useful to represent different groups of characters, succinctly.

Beispiele

Repre-sentation	Class
`[a-z]`	Kleinbuchstaben. Enthält keine Zeichen mit Akzentzeichen und Ligaturen
`[a-z0-9]`	Kleinbuchstaben von a bis z oder Zahlen von 0 bis 9
`[A-Za-z-]`	Groß-oder Kleinbuchstaben oder ein Bindestrich. Um den Bindestrich in eine Klasse einzuschließen, müssen Sie ihn am Anfang oder am Ende platzieren, um ihn nicht mit dem Bindestrich zu verwechseln, der einen Bereich von Zeichen angibt.
`[^0-9]`	Ein beliebiges Zeichen mit Ausnahme einer Ziffer. Die Einfügemarke (^), die am Anfang der Klasse platziert wird, schließt die Zeichen der Klasse (ergänzte Klasse) aus.
`[[a-z]--`	The lowercase consonants. A class can be included in a class. The characters `--` exclude what follows them
`[\w--[\d_]]`	All letters (including foreign accented characters). Abbreviated classes can be used inside a class

Beispiel:

```
<[^<>]+> to select an HTML tag
```

Shorthand character classes

Representa-tion	Class
`\d`	Eine Ziffer (identisch mit `[0-9]`)
`\D`	Beliebiges nicht numerisches Zeichen (identisch mit ``[^ 0-9] ``‘)
`\w`	An alphanumeric character (`[a-zA-Z0-9]`) including characters with accent mark and ligatures
`\W`	Jedes „nicht-Wort“-Zeichen
`\s`	Leerzeichen, unterbrechungsfreie Leerzeichen, Tabulator, Rücklauf Linie
`\S`	Any “non-whitespace” character
`.`	Any character except newline. Use the “dot all” checkbox or the `(?s)` regexp modifier to include the newline character.

The quantifiers

Quantifier	Number of occurrences of the expression preceding the quantifier
`?`	0 or 1 occurrence of the expression. Same as `{0,1}`
`+`	1 or more occurrences of the expression. Same as `{1,}`
`*`	0, 1 or more occurrences of the expression. Same as `{0,}`
`{n}`	Exactly n occurrences of the expression
`{min,max}`	Number of occurrences between the minimum and maximum values included
`{min,}`	Number of occurrences between the minimum value included and the infinite
`{,max}`	Number of occurrences between 0 and the maximum value included

Greed

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
`?=`	Positive lookahead (to be placed after the selection)
`?!`	Negative lookahead (to be placed after the selection)
`?<=`	Positive lookbehind (to be placed before the selection)
`?<!`	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 lookaround 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.

Rekursion

Representation	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

Representation	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 Danksagung

Ich bedanke mich für Hinweise, Korrekturen und so weiter bei:

- ldolse
- kovidgoyal

- chaley
- dwanthny
- kacir
- Starson17
- Orpheu

For more about regexps see The Python User Manual[102]. The actual regular expression library used by calibre is: regex[103] which supports several useful enhancements over the Python standard library one.

10.5 Schreibe Deine eigenen Erweiterungen zur Anpassung von Calibres Funktionalität

Calibre ist Modular aufgebaut. Fast alle Funktionen in Calibre sind mit Plugin's erstellt. Sie werden zum Umwandeln, zum herunterladen von Newsfeeds (dann nennen wir sie Rezepte - recipes), als Teil der Benutzeroberfläche, zum Verbinden zwischen verschiedenen Komponenten, zum Bearbeiten von Dateien beim Import und so weiter genutzt. Eine komplette Liste aller eingebauten Plugin's sehen Sie unter: guilabel:*Preferences->Advanced->Plugins*.

Hier werden Sie lernen, wie Sie ihre eigene Erweiterung erstellen, um neue Funktionen zu |app| hinzuzufügen.

Inhalte

- *Aufbau einer Calibre-Erweiterung* (die Titelseite 226)
- *Ein Benutzeroberflächen-Plugin* (die Titelseite 227)
  - *__init__.py* (die Titelseite 228)
  - *ui.py* (die Titelseite 229)
  - *main.py* (die Titelseite 230)
  - *Ressourcen aus der Plugin ZIP-Datei nutzen* (die Titelseite 233)
  - *Benutzer-Einstellungen des Plug-Ins aktivieren* (die Titelseite 233)
- *Buch-Plug-In bearbeiten* (die Titelseite 235)
  - *main.py* (die Titelseite 236)
- *Hinzufügen von Übersetzungen zu Ihrer Erweiterung* (die Titelseite 239)
- *Die Erweiterungs-API* (die Titelseite 240)
- *Fehlerdiagnose bei Erweiterungen* (die Titelseite 240)
- *Weitere Erweiterungs-Beispiele* (die Titelseite 240)
- *Teilen Ihrer Erweiterungen mit anderen* (die Titelseite 240)

Bemerkung

Dies gilt nur für Calibre Versionen >= 0.8.60

[102] https://docs.python.org/library/re.html
[103] https://bitbucket.org/mrabarnett/mrab-regex/src/hg/

10.5.1 Aufbau einer Calibre-Erweiterung

Ein Calibre Plugin ist sehr einfach. Es besteht aus einer Zip-Datei, die eine Python - (vgl. Programmiersprache) Datei und weiteren Teilen die dafür notwendig sind - wie Bilder - enthält. Am besten Sie sehen sich ein Beispiel an.

Angenommen Sie haben eine Installation von Calibre die Sie dafür nutzen eigene e-Dokumente im EPUB und MOBI format zu publizieren. Für alle Ihre Dateien soll Calibre den ‚publisher' -Metatag auf ‚Hello world' setzen. Unten sehen Sie wie das gemacht wird. Erstellen Sie eine (Text-) Datei mit dem Namen file:*__init__.py* (Diese Bezeichnung muß jedem Plugin als Dateiname gegeben werden) öffnen Sie die Datei in einem Editor und schreiben Sie folgenden Programmcode hinein:

```
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
```

Das war es schon. Um dieses kleine Programm nun als Plugin zu Calibre hinzuzufügen, wechseln Sie in den Ordner Ihrer kleinen Datei und führen Sie folgenden Befehl (auf der Komandozeile) aus.:

```
calibre-customize -b .
```

Bemerkung

Für macOS sind die Komandozeilentools in Calibre integriert. Sollte Calibre z.B. in `/Applications` installiert sein, dann finden Sie die Tools unter `/Applications/calibre.app/Contents/MacOS/`.

You can download the Hello World plugin from helloworld_plugin.zip[104].

Jedes Mal wenn Sie nun mit Calibre ein Buch konvertieren, wird die Methode `run()` des Plugin's ausgeführt und damit der ‚publisher' Metatag auf „Hello World" geändert. Das ist ein einfaches Plugin, machen Sie sich an ein etwas komplexeres mit dem Sie ein Element zur Benutzeroberfläche hinzufügen.

[104] https://calibre-ebook.com/downloads/helloworld_plugin.zip

10.5.2 Ein Benutzeroberflächen-Plugin

Diese Plugin wir auf ein paar Dateien verteilt (damit die einzelnen Teile überschaubar bleiben). Sie erkennen daran wie Sie auf, in der Plugin-Datei enthaltene, Resourcen (Bilder oder Daten) zugreifen, Nutzern erlauben Ihr Plugin zu konfigurieren, wie Sie Elemente der Benutzeroberfläche erstellen und wie Sie die Datenbank der Bücher in Calibre ansprechen und darin suchen.

You can download this plugin from interface_demo_plugin.zip[105]

Wichtig ist das die ZIP-Datei, die Ihr Plugin ist, nicht ganz wenige Dateien enthält. Unten sehen Sie eine Beschreibung achten Sie besonders auf `plugin-import-name-interface_demo.txt`.

plugin-import-name-interface_demo.txt
: Um in einem Plugin mehrere (Programm-) Dateien zu nutzen (Module nennt man das in Python) müssen diese einem genauen Namensschema entsprechen. Die Datei an sich hat keinen Inhalt. Die Form des Dateinamens ist : `plugin-import-name-**some_name**.txt`. Immer wenn in der ZIP-Datei eine solche Datei existiert können Sie innerhalb des Plugins Programmcode aus allen anderen enthaltenen .py Dateien auf folgende Art nutzen:

```
from calibre_plugins.some_name.some_module import some_object
```

Der Präfix `calibre_plugins` muss immer in der Datei stehen. `some_name` wird mit dme Namen der oben beschriebenen leeren Datei ersetzt. `some_module` ist der Name einer weiteren Datei `some_module.py` innerhalb der ZIP-Datei. Beachten Sie das das genau so nützlich und nutzbar ist wie Importe in der Programmiersprache Python selbst. Sie können ‚packages' und ‚subpackages' von .py Modulen innerhalb der ZIP-Datei erstellen wie Sie das mit Python auch tun würden. Das Funktioniert genau so wie die Nutzung von __init__.py Dateien. (vgl. python.org)

Der Name den Sie für eine Datei bei `some_name` wählen, ist in allen Plugins verfügbar und damit global! Er sollte natürlich nicht schon vorhanden sein, da Sie ansonsten die schon vorhandenen Funktionen ersetzen. Wählen Sie also den Name möglichst eindeutig und „beschreibend" aus. Zusätzlich muss der Name den Python vorschriften entsprechen: Nur Buchstaben, Nummern und der Unterstrich sind erlaubt.

__init__.py
: Wie schon zuvor ist das eine Datei, die eine Plugin Klasse definiert

main.py
: Diese Datei enthält den eigentlichen Code, der etwas nützliches tut

ui.py
: Diese Datei definiert das Benutzer Interface des Plugin's

images/icon.png
: Das Symbol für diese Erweiterung

about.txt
: Eine Textdatei mit einer Beschreibung des Plugin

translations
: Ein Ordner, der .mo-Dateien mit Übersetzungen der Benutzeroberfläche Ihrer Erweiterung in verschiedenen Sprachen enthält. Details nachfolgend.

Nun ein Blick auf den Code.

[105] https://calibre-ebook.com/downloads/interface_demo_plugin.zip

__init__.py

Zuerst die obligatorische ``__init__.py`, in dieser werden plugin metadaten gesetzt.

```
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
        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.
        '''
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        # 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()
```

Das einzig erwähnenswerte Datum hier ist das Datenfeld `actual_plugin`. Da Calibre sowohl ein Komandozeilen- als auch ein grafisches Interface hat, sollte nicht gleich in der ersten Datei schon eine grafische Schnittstelle geladen werden. Das Feld „actual_plugin“ weist darauf hin das sich eben dort ein User Interface befindet. Dieses wird eben nur dann ausgeführt, wenn das Plugin innerhalb der grafischen Oberfläche gestartet wird.

Denke daran, damit das funktionieren kann, muß in der Plugin - ZIP- Datei eine Datei mit dem Namen wie: plugin-import-name-some_name.txt file enthalten sein, wie weiter oben gezeigt.

Zudem gibt es ein paar Methoden die das Konfigurieren des Plugin durch die Nutzer ermöglichen. Wie das geht erfahren Sie weiter unten.

ui.py

Schauen Sie nun die Datei ui.py an. In dieser definieren Sie wie eine grafische Benutzeroberfläche Ihres Plugin aussehen wird. Der Sourcecode (Quellcode) ist sehr gut kommentiert und sollte selbsterklärend sein:

```
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 don't 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,
            'Run the Interface Plugin Demo', 'Ctrl+Shift+F1')

    def genesis(self):
        # This method is called once per plugin, do initial setup here
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        # 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

Die aktuelle Logik um das Benutzerinterface für einen Plugin Demo Dialog zu implementieren.

```
from calibre_plugins.interface_demo.config import prefs


class DemoDialog(QDialog):

    def __init__(self, gui, icon, do_user_config):
        QDialog.__init__(self, gui)
        self.gui = gui
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        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,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        # 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'))

    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.setEditText('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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        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
                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 don't 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',
                f'Updated the metadata in the files of {len(ids)} book(s)',
                show=True)

    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

Ressourcen aus der Plugin ZIP-Datei nutzen

Calibres Plugin ‚loading system' stellt eine Reihe von eingebauten Funktionen bereit, mit denen man Dateien und Inhalte aus der Plugin ZIP-Datei bekommt.

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.

Benutzer-Einstellungen des Plug-Ins aktivieren

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
        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 calibre.utils.config import JSONConfig
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# 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 don't 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)
        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()
```

Das `prefs`-Objekt ist nun überall im gesamten Erweiterungs-Code verfügbar über ein einfaches:

```
from calibre_plugins.interface_demo.config import prefs
```

You can see the `prefs` object being used in main.py:

```
    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

10.5.3 Buch-Plug-In bearbeiten

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[106].

The first step, as for all plugins is to create the import name empty txt file, as described *above* (die Titelseite 227). 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.

```

```

(Fortsetzung auf der nächsten Seite)

[106] https://calibre-ebook.com/downloads/editor_demo_plugin.zip

(Fortsetzung der vorherigen Seite)

```
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* (die Titelseite 371) 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* (die Titelseite 371) class for more details.

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.

```
from css_parser.css import CSSRule
from qt.core import QAction, QInputDialog

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


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: F821
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
            # 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
                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)
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                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.
        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:f}')
            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()* (die Titelsei-

te 372). 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* (die Titelseite 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* (die Titelseite 364) 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 *API documentation for the e-book editing tools* (die Titelseite 363).

10.5.4 Hinzufügen von Übersetzungen zu Ihrer Erweiterung

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[107] program for this.

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
```

Verschieben Sie die .mo-Dateien in den `translations`-Ordner Ihrer Erweiterung.

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
```

„de“ mit dem Sprachcode der zu testenden Sprache ersetzen.

For translations with plurals, use the `ngettext()` function instead of `_()`. For example:

```
ngettext('Delete a book', 'Delete {} books', num_books).format(num_books)
```

[107] https://poedit.net/

10.5.5 Die Erweiterungs-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 *API-Dokumentation für Erweiterungen* (die Titelseite 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* (die Titelseite 346).

10.5.6 Fehlerdiagnose bei Erweiterungen

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
```

Damit wird das laufende Calibre Programm beendet. Warten Sie, bis sich Calibre vollständig beendet hat. Dann aktualisieren Sie Ihre Erweiterung und starten Calibre neu.

10.5.7 Weitere Erweiterungs-Beispiele

You can find a list of many sophisticated calibre plugins here[108].

10.5.8 Teilen Ihrer Erweiterungen mit anderen

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[109].

10.6 Typesetting mathematics in e-books

The calibre E-book viewer has the ability to display mathematics embedded in e-books (EPUB and HTML files). You can typeset the mathematics directly with TeX or MathML or AsciiMath. The calibre E-book viewer uses the excellent MathJax[110] library to do this. This is a brief tutorial on creating e-books with mathematics in them that work well with the calibre E-book viewer.

10.6.1 Eine einfache HTML-Datei mit Formeln

You can write mathematics inline inside a simple HTML file and the calibre E-book viewer will render it into properly typeset mathematics. In the example below, we use TeX notation for mathematics. You will see that you can use normal TeX commands, with the small caveat that ampersands and less than and greater than signs have to be written as & < and > respectively.

[108] https://www.mobileread.com/forums/showthread.php?t=118764
[109] https://www.mobileread.com/forums/forumdisplay.php?f=237
[110] https://www.mathjax.org

The first step is to tell calibre that this will contains mathematics. You do this by adding the following snippet of code to the <head> section of the HTML file:

```
<script type="text/x-mathjax-config"></script>
```

Fertig, jetzt können Sie die Formel genauso, wie in einer .tex-Datei, eingeben. Beispielsweise hier für die Lorentz-Gleichung:

```
<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} &amp; = \sigma(y-x) \\
\dot{y} &amp; = \rho x - y - xz \\
\dot{z} &amp; = -\beta z + xy
\end{align}
</p>
```

This snippet looks like the following screen shot in the calibre E-book viewer.

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Abb. 1: *Die Lorentz-Gleichung*

The complete HTML file, with more equations and inline mathematics is reproduced below. You can convert this HTML file to EPUB in calibre to end up with an e-book you can distribute easily to other people.

```
<!DOCTYPE html>
<html>
<!-- Copyright (c) 2012 Design Science, Inc. -->
<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 recpgnize 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;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
  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} &amp; = \sigma(y-x) \label{lorenz}\\
\dot{y} &amp; = \rho x - y - xz \\
\dot{z} &amp; = -\beta z + xy
\end{align}
</p>

<h2>The Cauchy-Schwarz Inequality</h2>

<p>\[
\left( \sum_{k=1}^n a_k b_k \right)^{\!\!2} \leq
 \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
\]</p>

<h2>A Cross Product Formula</h2>

<p>\[
  \mathbf{V}_1 \times \mathbf{V}_2 =
   \begin{vmatrix}
    \mathbf{i} &amp; \mathbf{j} &amp; \mathbf{k} \\
    \frac{\partial X}{\partial u} &amp; \frac{\partial Y}{\partial u} &amp; 0 \\
    \frac{\partial X}{\partial v} &amp; \frac{\partial Y}{\partial v} &amp; 0 \\
   \end{vmatrix}
\]</p>

<h2>The probability of getting \(k\) heads when flipping \(n\) coins is:</h2>

<p>\[P(E) = {n \choose k} p^k (1-p)^{ n-k} \]</p>

<h2>An Identity of Ramanujan</h2>

<p>\[
   \frac{1}{(\sqrt{\phi \sqrt{5}}-\phi) e^{\frac25 \pi}} =
     1+\frac{e^{-2\pi}} {1+\frac{e^{-4\pi}} {1+\frac{e^{-6\pi}}
      {1+\frac{e^{-8\pi}} {1+\ldots} } } }
\]</p>
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
<h2>A Rogers-Ramanujan Identity</h2>

<p>\[
  1 +  \frac{q^2}{(1-q)}+\frac{q^6}{(1-q)(1-q^2)}+\cdots =
    \prod_{j=0}^{\infty}\frac{1}{(1-q^{5j+2})(1-q^{5j+3})},
     \quad\quad \text{for $|q|&lt;1$}.
\]</p>

<h2>Maxwell's Equations</h2>

<p>
\begin{align}
  \nabla \times \vec{\mathbf{B}} -\, \frac1c\, \frac{\partial\vec{\mathbf{E}}}{\
↪partial t} &amp;= \frac{4\pi}{c}\vec{\mathbf{j}} \\
  \nabla \cdot \vec{\mathbf{E}} &amp;= 4 \pi \rho \\
  \nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial\vec{\mathbf{B}}}{\
↪partial t} &amp;= \vec{\mathbf{0}} \\
  \nabla \cdot \vec{\mathbf{B}} &amp;= 0
\end{align}
</p>

<h2>Inline Mathematics</h2>

<p>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.</p>

<h2>References to equations</h2>

<p>Here is a reference to the Lorenz Equations (\ref{lorenz}). Clicking on the␣
↪equation number will take you back to the equation.</p>

</body>
</html>
```

Bemerkung

The calibre E-book viewer supports MathML as well as TeX, but you must include the `<script type="text/x-mathjax-config"></script>` line in your HTML file otherwise the MathML will not render.

10.6.2 Weitere Informationen

Since the calibre E-book viewer uses the MathJax library to render mathematics, the best place to find out more about mathematics in e-books and get help is the MathJax website[111].

[111] https://www.mathjax.org

10.7 Erstellen von AZW3-, EPUB- und MOBI-Katalogen

Mit Calibres Funktion „Katalog erstellen“ können Sie Kataloge Ihrer eigenen Bibliothek in einer Vielzahl von Formaten erstellen. Diese Hilfedatei beschreibt die Katalogisierungsoptionen beim Erzeugen eines Katalogs im AZW3-, EPUB- und MOBI-Format.

- *Auswählen von zu katalogisierenden Büchern* (die Titelseite 244)
- *Eingeschlossene Abschnitte* (die Titelseite 245)
- *Präfixe* (die Titelseite 245)
- *Ausgeschlossene Bücher* (die Titelseite 246)
- *Ausgeschlossene Genres* (die Titelseite 246)
- *Weitere Optionen* (die Titelseite 247)
- *Benutzerdefinierte Katalog-Titelbilder:* (die Titelseite 248)
- *Weitere Hilfsressourcen* (die Titelseite 248)

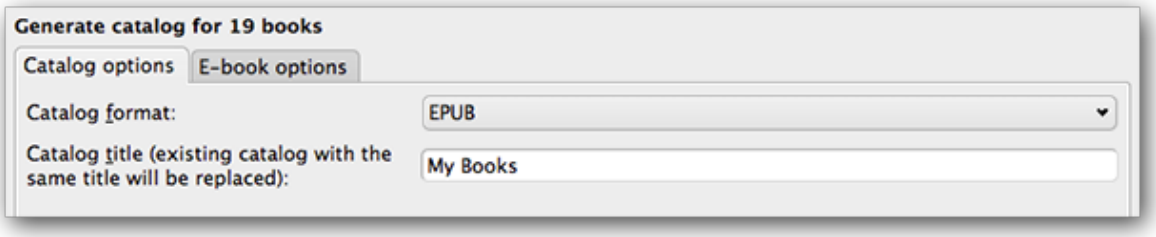
10.7.1 Auswählen von zu katalogisierenden Büchern

Falls Sie ihre komplette Bibliothek katalogisieren möchten, entfernen Sie alle Such- oder Filterkriterien im Hauptfenster. Mit der Selektion eines einzelnen Buches im Hauptfenster werden alle Bücher der Bibliothek in den zu erzeugenden Katalog übernommen. Einzelne Bücher können nach verschiedenen Kriterien ausgeschlossen werden. Für weitere Informationen siehe unter dem Abschnitt *Ausgeschlossene Genres* (die Titelseite 246).

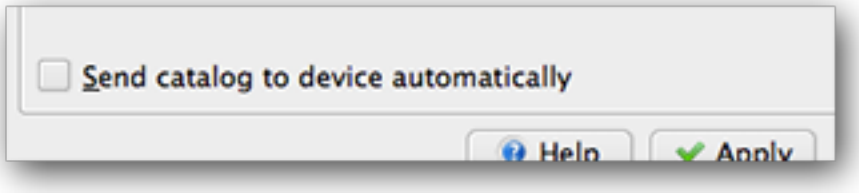
Wenn Sie nur *einige* Bücher aus der Bibliothek katalogisieren wollen, haben Sie zwei Möglichkeiten:

- Wählen Sie mehrere Bücher gleichzeitig zur Katalogisierung aus. Mit mehr als einem ausgewählten Buch in Calibres Hauptfenster werden nur die ausgewählten Bücher katalogisiert.
- Benutzen Sie das Suchfeld oder den Tag-Browser, um die angezeigten Bücher zu filtern. Es werden nur die angezeigten Bücher katalogisiert.

Um mit der Katalogerstellung zu beginnen, wählen Sie den Menüpunkt *Bücher konvertieren > Erstellen Sie einen Katalog der Bücher in Ihrer Calibre-Bibliothek*. Sie können auch eine Schaltfläche *Katalog erstellen* zu einer Symbolleiste in :guilabel:``Präferenzen > Benutzeroberfläche > Symbolleisten & Menüs` hinzufügen, um den Zugang zum Dialogfeld ‚Katalog erstellen‘ zu erleichtern.

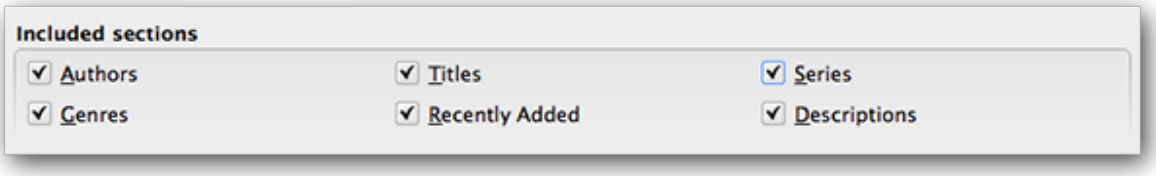


Unter *Katalogeinstellungen* wählen Sie **AZW3, EPUB oder MOBI** als Katalogformat aus. Es stehen weitere Formate als Auswahl zur Verfügung. Im Feld für *Katalog-Titel* wird der Name des Katalogs angegeben. Ein bestehender Katalog mit dem gleichen Titel wird durch den neu generierten Katalog ersetzt.



Bei Aktivierung von *Katalog automatisch ans Gerät senden* wird der generierte Katalog automatisch nach Abschluss an ein angeschlossenes Gerät gesendet.

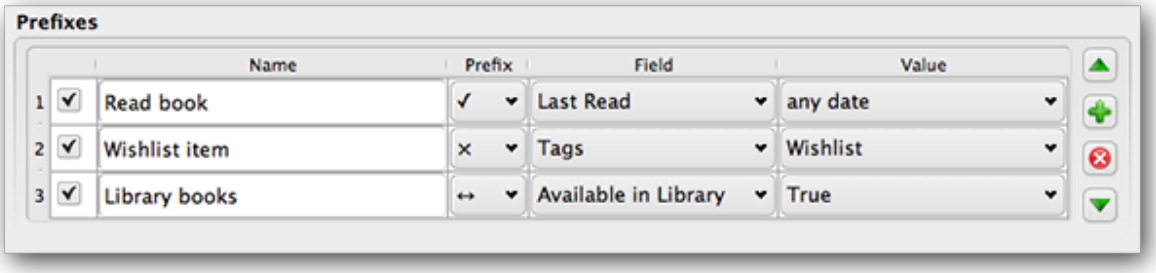
10.7.2 Eingeschlossene Abschnitte



Abschnitte, die durch ein Häkchen aktiviert sind, werden in den generierten Katalog aufgenommen:

- *Autoren* - Darstellung aller Bücher, nach Autor sortiert, in einem Listenformat. Bücher ohne Serie werden vor Serienbüchern gelistet.
- *Titel* - alle Bücher werden als Liste nach Titel sortiert.
- *Serien* - alle Serien-Bücher werden als Liste nach Serienname und -nummer sortiert.
- *Genres* - einzelne Genres werden als Liste nach Autor und Serie sortiert.
- *Zuletzt hinzugefügt* - alle Bücher in umgekehrter chronologischer Reihenfolge. Die Liste enthält alle Bücher, die innerhalb der letzten 30 Tage und danach von Monat zu Monat hinzugefügt wurden.
- *Beschreibungen* - detaillierte Beschreibungsseite für jedes Buch, einschließlich einer Titelbildvorschau und Kommentaren. Sortiert nach Autor, Bücher ohne Serie werden vor Serienbüchern gelistet.

10.7.3 Präfixe



Durch Präfix-Regeln können Sie Bücherlisten ein Präfix hinzufügen, wenn bestimmte Kriterien erfüllt sind. Mit diesen Regeln können Sie beispielsweise gelesene Bücher mit einem Häkchen versehen oder Bücher Ihrer Wunschliste mit einem X markieren.

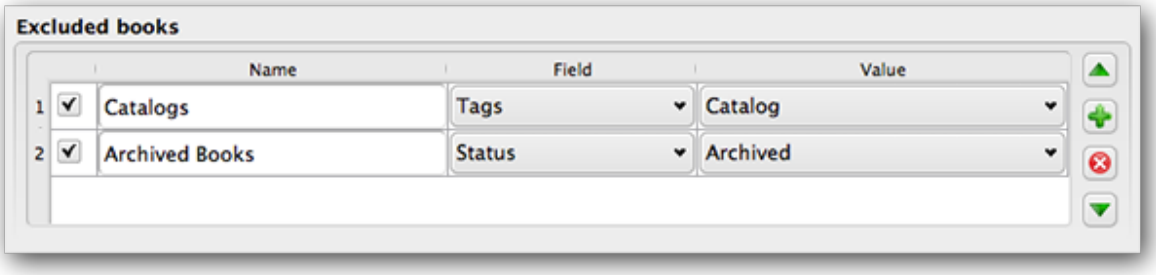
Das Kontrollkästchen in der ersten Spalte aktiviert eine Regel. *Name* ist ein von Ihnen gewählter Name für eine Regel. *Feld* ist entweder ein *Schlagwort* oder eine benutzerdefinierte Spalte aus Ihrer Bibliothek. *Wert* ist der Inhalt des *Felds*, dem die Regel entsprechen soll. Wenn eine Präfix-Regel erfüllt ist, wird das Buch mit dem ausgewählten *Präfix* markiert.

Im obigen Beispiel wurden drei Präfix-Regeln verwendet:

1. *Read book* gibt an, dass jedes Buch mit einem Datum in der benutzerdefinierten Spalte mit Namen *Last read* ein Häkchen-Symbol vorangestellt bekommt.
2. *Wishlist* - Diese Regel bestimmt, dass jedes Buch mit dem Schlagwort *Wishlist* ein X-Symbol vorangestellt bekommt.
3. *Library books* bestimmt, dass jedes Buch mit dem Wert True (oder Ja) in der benutzerdefinierten Spalte *Available in Library* durch ein vorangestelltes Doppelpfeil-Symbol gekennzeichnet wird.

Die erste zutreffende Regel bestimmt das Präfix. Abgeschaltete oder unvollständige Regeln werden ignoriert.

10.7.4 Ausgeschlossene Bücher



Durch Ausschlussregeln können Sie Bücher angeben, die nicht katalogisiert werden sollen.

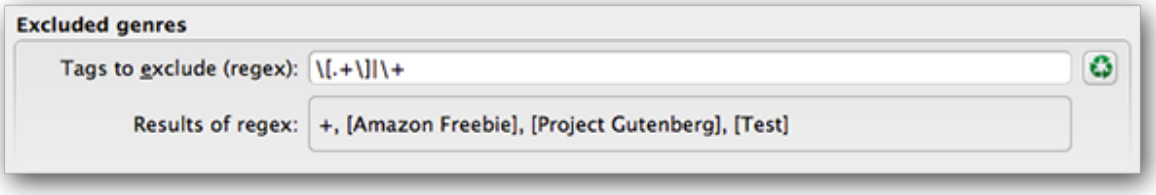
Das Kontrollkästchen in der ersten Spalte aktiviert eine Regel. *Name* ist der von Ihnen gewählte Name für eine Regel. *Feld* ist entweder ein *Schlagwort* oder eine benutzerdefinierte Spalte aus Ihrer Bibliothek. *Wert* ist der Inhalt des *Felds*, dem die Regel entsprechen soll. Wenn eine Ausnahmeregel erfüllt ist, wird das Buch aus dem zu erzeugenden Katalog ausgeschlossen.

Im obigen Beispiel wurden zwei Ausschlussregeln angegeben:

1. Die Regel *Catalogs* legt fest, dass jedes Buch mit dem Schlagwort *Catalog* bei der Generierung des Katalogs ausgeschlossen wird.
2. Die Regel *Archived Books* bestimmt, dass jedes Buch mit dem Wert *Archived* in der benutzerdefinierten Spalte *Status* aus der Generierung des Katalogs ausgeschlossen wird.

Alle Regeln werden für jedes Buch ausgewertet. Abgeschaltete oder unvollständige Regeln werden ignoriert.

10.7.5 Ausgeschlossene Genres



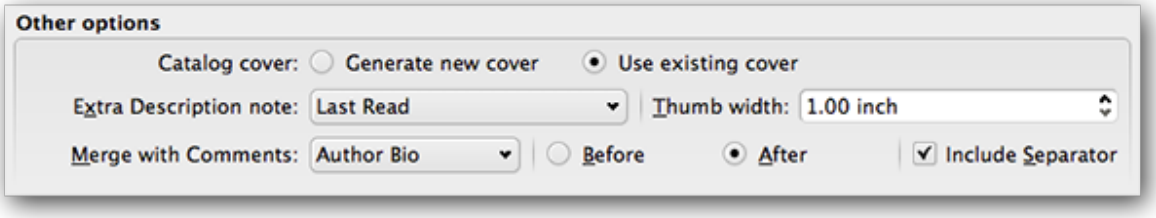
Wenn der Katalog erzeugt wird, werden die Schlagwörter Ihrer Datenbank als Genres verwendet. Sie könnten beispielsweise die Schlagwörter `Fiction` und `Nonfiction` verwenden. Die Schlagwörter werden im erzeugten Katalog zu Genres und die Bücher werden entsprechend der Schlagwörter in die zugehörigen Genre-Listen einsortiert. Ein Buch wird in jedem Genre-Abschnitt aufgeführt, für den es ein entsprechendes Schlagwort besitzt.

Sie möchten möglicherweise bestimmte Schlagwörter für andere Zwecke nutzen, vielleicht ein + um ein gelesenes Buch zu kennzeichnen oder ein Schlagwort in eckigen Klammern wie `[Amazon Freebie]` um z. B. Buchquellen anzugeben. Mit dem Regex *Ausgeschlossen Genres* geben Sie an, welche Schlagwörter bei der Erzeugung des Katalogs ausgeschlossen werden sollen. Das voreingestellte Regex-Muster `\[.+\]\+` schließt alle Bücher mit Schlagwörtern in eckigen Klammern der Form `[Schlagwort]` sowie alle mit + als gelesen gekennzeichneten Bücher von der Katalogerzeugung aus.

Sie können Schlagwörter auch direkt in einem Regex benutzen. Beispielsweise `[Amazon Freebie]` oder `[Project Gutenberg]`. Wenn mehrere Schlagwörter für den Ausschluss auflistet werden sollen, sind diese durch ein Pipezeichen (senkrechter Strich) zu trennen: `[Amazon Freebie]|[Project Gutenberg]`.

Ergebnis des Regex zeigt Ihnen, welche Schlagwörter bei der Erstellung des Katalogs ausgeschlossen werden. Diese basieren auf den verwendeten Schlagwörtern in Ihrer Datenbank und dem oben eingegebenen Regex-Muster. Die Ergebnisse werden aktualisiert, sobald Sie das Regex-Muster ändern.

10.7.6 Weitere Optionen



Katalog Titelbild bestimmt, ob ein neues Titelbild erzeugt oder ein vorhandenes Titelbild verwendet werden soll. Es ist möglich ein benutzerdefiniertes Titelbild für Ihre Kataloge zu erstellen - siehe Custom catalog covers für weitere Informationen. Wenn Sie ein benutzerdefiniertes Titelbild erstellt haben und es wiederverwenden möchten, aktivieren Sie *Vorhandenes Titelbild verwenden* oder ansonsten *Neues Titelbild erzeugen*.

Extra Description note gibt den Inhalt einer benutzerdefinierten Spalte an, die auf der Beschreibungsseite neben dem Miniaturbild des Covers eingefügt werden soll. Zum Beispiel möchten Sie das Datum, an dem Sie ein Buch zum letzten Mal gelesen haben, mit Hilfe einer benutzerdefinierten Spalte mit dem Namen: guilabel: *Last Read* anzeigen. Weitere Informationen zur erweiterten Verwendung der Notiz-Beschreibungs-Funktion finden Sie unter this post in the calibre forum[112].

Thumb width gibt eine Breiteneinstellung für Miniaturansichten des Covers an, die auf den Beschreibungsseiten enthalten sind. Miniaturansichten werden zwischengespeichert, um die Leistung zu verbessern. Um mit verschiedenen Breiten zu experimentieren, probieren Sie, einen Katalog mit nur wenigen Büchern zu erstellen, bis Sie Ihre bevorzugte Breite bestimmt haben. Beim ersten Erzeugen eines Katalogs mit einer neuen Miniaturansichtsbreite wird die Leistung langsamer, aber die folgenden Generierungen des Katalogs nutzen den Miniaturansichts-Zwischenspeicher.

guilabel:`Mit Kommentaren zusammenführen` gibt eine benutzerdefinierte Spalte an, deren Inhalt während der Katalogerstellung zerstörungsfrei mit den Kommentar-Metadaten zusammengeführt wird. Zum Beispiel könnten Sie eine benutzerdefinierte Spalte :guilabel:`Author bio` haben, die Sie an die Kommentar-Metadaten anhängen möchten. Sie können den benutzerdefinierten Spalteninhalt *vor oder nach* dem Kommentarabschnitt einfügen und optional den angehängten Inhalt mit einem horizontalen Regeltrennzeichen trennen. Zu den zulässigen benutzerdefinierten Spaltentypen gehören ``Text, Kommentare und Composite.

[112] https://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395

10.7.7 Benutzerdefinierte Katalog-Titelbilder:

Mit dem Generate Cover plugin[113] installiert, können Sie benutzerdefinierte Titelseiten für Ihren Katalog erstellen. Um das Plugin zu installieren, gehe zu *Einstellungen* > *Erweitert* > *Erweiterungen* > *Plugins* > *Neue Erweiterungen beziehen*.

10.7.8 Weitere Hilfsressourcen

Weitere Informationen zur Katalogfunktion von Calibre finden Sie im MobileRead-Forum unter dem Punkt „Creating Catalogs: Start here <https://www.mobileread.com/forums/showthread.php?t=118556>`_“, wo Sie Informationen zum Anpassen der Katalogvorlagen und zum Einreichen eines Fehlerberichts finden.

Um Fragen zu stellen oder die Katalogfunktion von Calibre mit anderen Benutzern zu diskutieren, besuchen Sie das MobileRead-Forum Library Management[114].

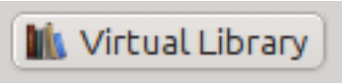
10.8 Virtuelle Bibliotheken

In Calibre ist eine Virtuelle Bibliothek eine Möglichkeit Calibre mitzuteilen, nur einen Teil einer normalen Bibliothek zu öffnen. Zum Beispiel möchtest du nur mit Büchern eines bestimmten Autors arbeiten oder Büchern die mit einem bestimmten Schlagwort versehen sind. Die Verwendung von Virtuellen Bibliotheken ist die bevorzugte Weise deine große Büchersammlung zu unterteilen. Es ist vorteilhafter als deine Bibliothek in viele kleine Bibliotheken zu zerteilen. Wenn du deine ganze Bibliothek durchsuchen möchtest, kannst du einfach zur gesamten Bibliothek zurückwechseln. Es gibt keine Möglichkeit in Calibre mehrere getrennte kleine Bibliotheken gleichzeitig zu durchsuchen.

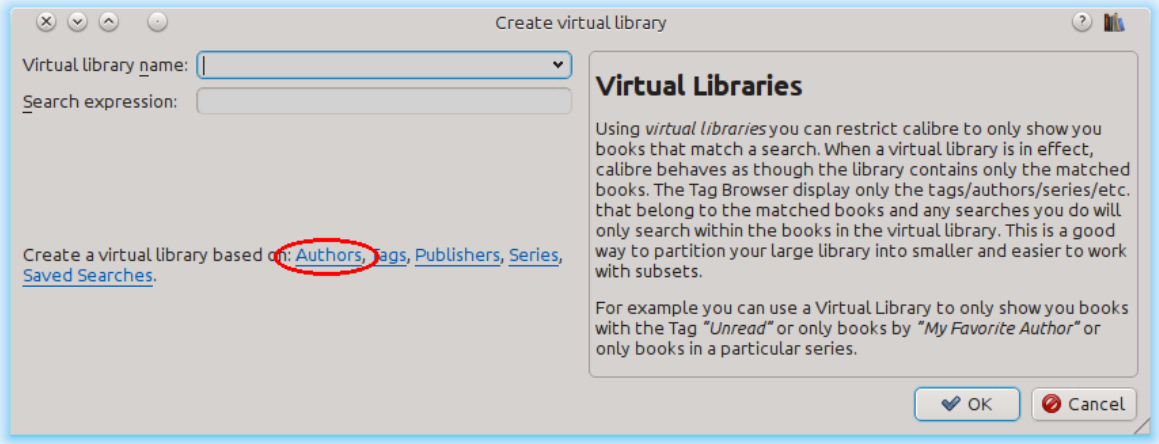
Eine Virtuelle Bibliothek unterscheided siche von einer einfachen Suche. Eine Suche wird die Liste an Büchern nur in der Bücherliste einschränken. Ein Virtuelle Bibliothek tut dies auch, aber darüber hinaus beschränkt es die Einträge, die im *Tag browser* auf der linken Seite gezeigt werden. Der Tag-Browser zeigt dann nur Tags, Autoren, Serien, Verlage, etc. die aus Büchern der Virtuellen Bibliothek stammen. Eine Virtuelle Bibliothek verhält sich daher als würde die tatsächliche Bibliothek nur einen einschränkten Satz an Büchern enthielte.

[113] https://www.mobileread.com/forums/showthread.php?t=124219
[114] https://www.mobileread.com/forums/forumdisplay.php?f=236

10.8.1 Erstellen von virtuellen Bibliotheken



Um eine virtuelle Bibliothek zu nutzen klicken Sie *Virtual library* das Sie links neben der Suchleiste finden. Wählen Sie dort dann die *Create Virtual library* Option. Lassen Sie uns als erstes Beispiel eine virtuelle Bibliothek erstellen, die nur Bücher eines bestimmten Autors anzeigt. Wählen Sie *Authors* wie unten zu sehen ist aus. Dann wählen Sie einen Autor und bestätigen OK.



Der Virtuelle Bibliothek erstellen-Dialog wurde für dich ausgefüllt. Klick auf OK und du wirst sehen, dass eine neue Virtuelle Bibliothek erstellt und dorthin automatisch gewechselt wurde. Sie zeigt nur Bücher des ausgewählten Autors. Soweit es Calibre betrifft ist das, als ob deine Bibliothek nur Bücher des gewählten Autors enthält.

Du kannst jederzeit zur vollständigen Bibliothek durch erneutes Klicken auf *Virtual library* und durch Auswählen des Eintrags <*None*> wechseln.

Virtuelle Bibliotheken basieren auf Suchen. Sie können Ihre eigenen Suchbegriffe als Grundlage einer virtuellen Bibliothek verwenden. Dann wird diese Bibliothek nur die Bücher anzeigen, die der Suche entsprechen. Geben Sie daher zunächst einen Suchebegriff in die Suchleiste ein oder nutzen Sie *Tag browser* um eine Suche zu definieren. Entspricht das Ergebnis Ihren Erwartungen? Erstellen Sie damit eine eigene Bibliothek: klicken Sie auf den *Virtual library* Button, wählen Sie *Create library* aus und geben sie Ihrer virtuellen Bibliothek einen Namen. Die Suchfunktionen sind sehr mächtig, vieles was Sie damit tun können finden Sie als Beispiel unter *Die Suchoberfläche* (die Titelseite 12).

Beispiele für nützliche virtuelle Bibliotheken

- **Bücher am letzten Tag zu Calibre hinzugefügt:**
  date:>1daysago
- **Bücher im letzten Monat zu Calibre hinzugefügt:**
  date:>30daysago
- **Bücher mit Bewertung von 5 Sternen:**
  rating:5
- **Bücher mit Bewertung von mindestens 4 Sternen:**
  rating:>=4
- **Bücher ohne Bewertung:**
  rating:false

- **Heruntergeladene Zeitschriften über den Nachrichtenabruf von Calibre:**
  tags:=News and author:=calibre
- **Bücher ohne Schlagwörter:**
  tags:false
- **Bücher ohne Titelbilder:**
  cover:false

10.8.2 Arbeiten mit virtuellen Bibliotheken

Du kannst eine zuvor erstellte Virtuelle Bibliothek bearbeiten oder entfernen durch Klicken von *Virtual library* und Auswählen der passenden Aktion.

Sie können Calibre veranlassen, beim Öffnen der aktuellen Bibliothek immer nur eine bestimmte virtuelle Bibliothek anzuwenden. Dies wird unter unter *Einstellungen → Benutzeroberfläche → Verhalten* festgelegt.

Sie können einfach die aktuelle Suche für eine temporäre virtuelle Bibliothek nutzen indem sie den *Virtual library* Button klicken und dann **current search* auswählen.

Alle vorhandenen virtuellen Bibliotheken können als Tab über der Buchliste angezeigt werden. Das hilft Ihnen schnell zwischen verschiedenen (Teil-) Gruppen Ihrer Bücher hin und her zu wechseln. Klicken Sie auf den *Virtual library* Button und wählen Sie *Show Virtual libraries as tabs* um das zu aktivieren. Sie können diese Tab's nach belieben mit der Maus anordnen oder einzelne auch schließen. Haben Sie versehendlich die falsche virtuelle Bibliothek geschlossen, dann können Sie sie wieder sichtbar machen, indem Sie einen Rechtsklick auf die Tab-Leiste machen und den Tab wieder aktivieren.

10.8.3 Verwenden von Virtuellen Bibliotheken in Suchen

Suchen in einer virtuellen Bibliothek geht einfach, nutzen Sie `vl:` prefix. Ein Beispiel: `vl:Read` zeigt alle Bücher in der virtuellen Bibliothek *Read* an. Die Suche nach `vl:Read and vl:"Science Fiction"` zeigt die Bücher die gleichzeitig in beiden Bibliotheken sind.

Der auf `vl:` folgende Wert muss der Name einer Virtuellen Bibliothek sein. Wenn der Name der Virtuellen Bibliothek Leerzeichen enthält, schließe ihn in Anführungszeichen ein.

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 Verwendung zusätzlicher Einschränkungen

Die Anzeige der Bücher in einer virtuellen Bibliothek können Sie weiter einschränken in dem Sie *Additional restrictions* benutzen. Sind diese zusätzlichen Einschränkungen einmal gespeichert, können Sie sie später einer aktuellen virtuellen Bibliothek zuweisen. Nehmen wir beispielsweise an Sie haben eine virtuelle Bibliothek mit der Eigenschaft *Historical Fiction* versehen. Nun speichern Sie eine Suche nach ‚ungelesenen' Büchern ab. Klicken Sie dann auf den *Virtual Library* Button und wählen Sie die *Additional restriction* Option damit nur die entsprechenden historischen und ungelesenen Bücher angezeigt werden. Mehr zum Thema erfahren Sie unter *Speichern von Suchen* (die Titelseite 17).

KAPITEL 11

Das calibre:// URL Schema

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
```

Will open calibre with the library named `Some Library`. Library names are the folder name of the library folder with spaces replaced by underscores. The special value _ means the current library. The various types of URLs are documented below.

You can even place these links inside HTML files or Word documents or similar and the operating system will automatically run calibre to perform the specified action.

- *Switch to a specific library* (die Titelseite 251)
- *Show a specific book in calibre* (die Titelseite 252)
- *Open a specific book in the E-book viewer at a specific position* (die Titelseite 252)
- *Suche nach Büchern* (die Titelseite 252)
- *Open a book details window on a book in some library* (die Titelseite 253)
- *Open the notes associated with an author/series/etc.* (die Titelseite 253)
- *Hex encoding of URL parameters* (die Titelseite 253)

11.1 Switch to a specific library

The URL syntax is:

```
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* (die Titelseite 253) 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 Show a specific book in calibre

The URL syntax is:

```
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 Open a specific book in the E-book viewer at a specific position

The URL syntax is:

```
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 Suche nach Büchern

The URL syntax is:

```
calibre://search/Library_Name?q=query
calibre://search/Library_Name?eq=hex_encoded_query
```

Here query is any valid *search expression* (die Titelseite 12). If the search expression is complicated, *encode it as a hex string* (die Titelseite 253) and use `eq` instead. Leaving out the query will cause the current search to be cleared.

By default, if a Virtual library is selected, calibre will clear it before doing the search to ensure all books are found. If you want to preserve the Virtual library, use:

```
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

The URL syntax is:

```
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.

The URL syntax is:

```
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 Hex encoding of URL parameters

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`.

KAPITEL 12

Anpassen von Calibre

calibre has a highly modular design. Various parts of it can be customized. Here, you will learn:

- how to use environment variables and *tweaks* to customize calibre's behavior,
- how to specify your own static resources like icons and templates to override the defaults
- how to use *plugins* to add functionality to calibre.
- how to share icon themes and plugins with other calibre users.
- to see how to create *recipes* to add new sources of online content to calibre visit the Section *Hinzufügen Ihrer bevorzugten Nachrichtenwebsite* (die Titelseite 33).

Bemerkung

Note that although icon themes and plugins are indexed and downloadable via calibre's builtin updater, they are not part of calibre, and their canonical locations for support and source code are on the Mobileread forums[115] in their support threads.

- *Umgebungsvariablen* (die Titelseite 287)
- *Optimierungen* (die Titelseite 288)
- *Überschreiben von Symbolen, Vorlagen usw.* (die Titelseite 300)
- *Erstellen eines eigenen Symboldesigns für Calibre* (die Titelseite 301)
- *Anpassen von Calibre durch Erweiterungen* (die Titelseite 301)

[115] https://www.mobileread.com/forums/forumdisplay.php?f=166

12.1 API-Dokumentation für Erweiterungen

Definiert diverse abstrakte Basisklassen, die als Unterklassen verwendet werden können, um mächtige Erweiterungen zu erstellen. Die nützlichen Klassen sind:

- *Erweiterung* (die Titelseite 256)
- *Dateityp-Erweiterungen* (die Titelseite 258)
- *Metadaten-Erweiterungen* (die Titelseite 260)
- *Katalog-Erweiterungen* (die Titelseite 261)
- *Metadata download plugins* (die Titelseite 262)
- *Conversion plugins* (die Titelseite 265)
- *Gerätetreiber* (die Titelseite 268)
- *User interface actions* (die Titelseite 281)
- *Preferences plugins* (die Titelseite 285)

12.1.1 Erweiterung

class calibre.customize.**Plugin**(*plugin_path*)

Eine Calibre-Erweiterung. Hilfreiche Attribute sind unter anderem:

- `self.installation_type`: Stores how the plugin was installed.
- **`self.plugin_path`: Stores path to the ZIP file that contains** Erweiterung enthält, oder None, falls es eine integrierte Erweiterung ist
- **`self.site_customization`: Speichert eine eingegebene Anpassungszeichenkette** eingegeben wurde.

Methoden die in Unterklassen überschrieben werden sollten:

- *`initialize()`* (die Titelseite 257)
- *`customization_help()`* (die Titelseite 258)

Nützliche Verfahren:

- *`temporary_file()`* (die Titelseite 258)
- `__enter__()`
- *`load_resources()`* (die Titelseite 257)

`supported_platforms = []`

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

`name = 'Trivial Plugin'`

Der Name dieser Erweiterung. Sie müssen etwas anderes festlegen als Trivial-Erweiterung, damit es funktioniert.

`version = (1, 0, 0)`

Die Version dieser Erweiterung als 3-Tupel (major, minor, revision)

description = 'Macht rein gar nichts'

Eine kurze Beschreibung, was diese Erweiterung macht.

author = 'Unbekannt'

Der Autor dieser Erweiterung

priority = 1

When more than one plugin exists for a filetype, the plugins are run in order of decreasing priority. Plugins with higher priority will be run first. The highest possible priority is `sys.maxsize`. Default priority is 1.

minimum_calibre_version = (0, 4, 118)

Die älteste Version von Calibre, die diese Erweiterung erfordert

installation_type = None

The way this plugin is installed

can_be_disabled = True

Falls *False*, wird der Benutzer keine Möglichkeit haben, die Erweiterung zu deaktivieren. Bitte mit Vorsicht verwenden.

type = 'Basis'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

initialize()

Called once when calibre plugins are initialized. Plugins are re-initialized every time a new plugin is added. Also note that if the plugin is run in a worker process, such as for adding books, then the plugin will be initialized for every new worker process.

Perform any plugin specific initialization here, such as extracting resources from the plugin ZIP file. The path to the ZIP file is available as `self.plugin_path`.

Beachten Sie, dass `self.site_customization` an diesem Punkt **nicht** verfügbar ist.

config_widget()

Implement this method and *save_settings()* (die Titelseite 257) in your plugin to use a custom configuration dialog, rather then relying on the simple string based default customization.

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.

save_settings(*config_widget*)

Save the settings specified by the user with config_widget.

Parameter

config_widget – The widget returned by *config_widget()* (die Titelseite 257).

do_user_config(*parent=None*)

This method shows a configuration dialog for this plugin. It returns True if the user clicks OK, False otherwise. The changes are automatically applied.

load_resources(*names*)

If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

Zum Beispiel, um ein Bild zu laden:

```
pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png'])['images/icon.png
↪'])
icon = QIcon(pixmap)
```

Parameter
names – List of paths to resources in the ZIP file using / as separator

Rückgabe
A dictionary of the form `{name: file_contents}`. Any names that were not found in the ZIP file will not be present in the dictionary.

customization_help(*gui=False*)

Return a string giving help on how to customize this plugin. By default raise a `NotImplementedError`, which indicates that the plugin does not require customization.

If you re-implement this method in your subclass, the user will be asked to enter a string as customization for this plugin. The customization string will be available as `self.site_customization`.

Site customization could be anything, for example, the path to a needed binary on the user's computer.

Parameter
gui – Liefert HTML-Hilfe falls Tue, gibt ansonsten Hilfe als einfachen Text zurück.

temporary_file(*suffix*)

Return a file-like object that is a temporary file on the file system. This file will remain available even after being closed and will only be removed on interpreter shutdown. Use the `name` member of the returned object to access the full path to the created temporary file.

Parameter
suffix – Das Suffix der temporären Datei.

cli_main(*args*)

This method is the main entry point for your plugins command line interface. It is called when the user does: calibre-debug -r „Plugin Name“. Any arguments passed are present in the args variable.

12.1.2 Dateityp-Erweiterungen

class calibre.customize.**FileTypePlugin**(*plugin_path*)

Bases: *Plugin* (die Titelseite 256)

Eine Erweiterung, die einem bestimmten Satz von Dateitypen zugeordnet ist.

file_types = {}

Set of file types for which this plugin should be run. Use ‚*‘ for all file types. For example: `{'lit', 'mobi', 'prc'}`

on_import = False

Falls *True*, wird diese Erweiterung immer dann ausgeführt, wenn Bücher zur Datenbank hinzugefügt werden

on_postimport = False

If True, this plugin is run after books are added to the database. In this case the postimport and postadd methods of the plugin are called.

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

If True, this plugin is run just before a conversion

on_postprocess = False

If True, this plugin is run after conversion on the final file produced by the conversion output plugin.

type = 'Dateityp'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

run(*path_to_ebook*)

Run the plugin. Must be implemented in subclasses. It should perform whatever modifications are required on the e-book and return the absolute path to the modified e-book. If no modifications are needed, it should return the path to the original e-book. If an error is encountered it should raise an Exception. The default implementation simply return the path to the original e-book. Note that the path to the original file (before any file type plugins are run, is available as self.original_path_to_file).

The modified e-book file should be created with the `temporary_file()` method.

Parameter
path_to_ebook – Absoluter Pfad des eBooks.

Rückgabe
Absoluter Pfad des geänderten eBooks.

postimport(*book_id*, *book_format*, *db*)

Called post import, i.e., after the book file has been added to the database. Note that this is different from *postadd()* (die Titelseite 260) which is called when the book record is created for the first time. This method is called whenever a new file is added to a book record. It is useful for modifying the book record based on the contents of the newly added file.

Parameter

- **book_id** – Datenbankkennung des hinzugefügten Buches.
- **book_format** – Der Dateityp des Buchs das hinzugefügt wurde.
- **db** – Bibliotheksdatenbank.

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.

Parameter

- **book_id** – Datenbankkennung des hinzugefügten Buches.
- **book_format** – Der Dateityp des Buchs das hinzugefügt wurde.
- **db** – Bibliotheksdatenbank.

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.

Parameter

- **book_id** – Datenbankkennung des hinzugefügten Buches.
- **book_format** – Der Dateityp des Buchs das hinzugefügt wurde.
- **db** – Bibliotheksdatenbank.

postadd(*book_id*, *fmt_map*, *db*)

Called post add, i.e. after a book has been added to the db. Note that this is different from *postimport()* (die Titelseite 259), which is called after a single book file has been added to a book. postadd() is called only when an entire book record with possibly more than one book file has been created for the first time. This is useful if you wish to modify the book record in the database when the book is first added to calibre.

Parameter

- **book_id** – Datenbankkennung des hinzugefügten Buches.
- **fmt_map** – Map of file format to path from which the file format was added. Note that this might or might not point to an actual existing file, as sometimes files are added as streams. In which case it might be a dummy value or a non-existent path.
- **db** – Bibliotheksdatenbank

12.1.3 Metadaten-Erweiterungen

class calibre.customize.**MetadataReaderPlugin**(**args*, ***kwargs*)

Bases: *Plugin* (die Titelseite 256)

A plugin that implements reading metadata from a set of file types.

file_types = {}

Set of file types for which this plugin should be run. For example: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

version = (8, 7, 0)

Die Version dieser Erweiterung als 3-Tupel (major, minor, revision)

author = 'Kovid Goyal'

Der Autor dieser Erweiterung

type = 'Metadatenleser'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

get_metadata(*stream*, *type*)

Return metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data.

Parameter

type – The type of file. Guaranteed to be one of the entries in *file_types* (die Titelseite 260).

Rückgabe

A `calibre.ebooks.metadata.book.Metadata` object

class calibre.customize.**MetadataWriterPlugin**(**args*, ***kwargs*)

Bases: *Plugin* (die Titelseite 256)

A plugin that implements reading metadata from a set of file types.

file_types = {}

Set of file types for which this plugin should be run. For example: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

version = (8, 7, 0)

Die Version dieser Erweiterung als 3-Tupel (major, minor, revision)

author = 'Kovid Goyal'

Der Autor dieser Erweiterung

type = 'Metadatenschreiber'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

set_metadata(*stream*, *mi*, *type*)

Set metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data.

Parameter

- **type** – The type of file. Guaranteed to be one of the entries in *file_types* (die Titelseite 260).
- **mi** – A `calibre.ebooks.metadata.book.Metadata` object

12.1.4 Katalog-Erweiterungen

class calibre.customize.**CatalogPlugin**(*plugin_path*)

Bases: *Plugin* (die Titelseite 256)

Eine Erweiterung, die einen Kataloggenerator implementiert.

file_types = {}

Output file type for which this plugin should be run. For example: ‚epub' or ‚xml'

type = 'Katalogerzeugung'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

cli_options = []

CLI parser options specific to this plugin, declared as *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()

If plugin is not a built-in, copy the plugin's .ui and .py files from the ZIP file to $TMPDIR. Tab will be dynamically generated and added to the Catalog Options dialog in calibre.gui2.dialogs.catalog.py:Catalog

run(*path_to_output*, *opts*, *db*, *ids*, *notification=None*)

Run the plugin. Must be implemented in subclasses. It should generate the catalog in the format specified in file_types, returning the absolute path to the generated catalog file. If an error is encountered it should raise an Exception.

The generated catalog file should be created with the `temporary_file()` method.

Parameter

- **path_to_output** – Absoluter Pfad zur generierten Katalogdatei.
- **opts** – A dictionary of keyword arguments
- **db** – Ein LibraryDatabase2-Objekt

12.1.5 Metadata download plugins

class calibre.ebooks.metadata.sources.base.**Source**(**args*, ***kwargs*)

Bases: *Plugin* (die Titelseite 256)

type = 'Metadatenquelle'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

author = 'Kovid Goyal'

Der Autor dieser Erweiterung

supported_platforms = ['windows', 'osx', 'linux']

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

capabilities = frozenset({})

Set of capabilities supported by this plugin. Useful capabilities are: ‚identify', ‚cover'

touched_fields = frozenset({})

List of metadata fields that can potentially be download by this plugin during the identify phase

has_html_comments = False

Set this to True if your plugin returns HTML formatted comments

supports_gzip_transfer_encoding = False

Setting this to True means that the browser object will indicate that it supports gzip transfer encoding. This can speedup downloads but make sure that the source actually supports gzip transfer encoding correctly first

ignore_ssl_errors = False

Set this to True to ignore HTTPS certificate errors when connecting to this source.

cached_cover_url_is_reliable = True

Cached cover URLs can sometimes be unreliable (i.e. the download could fail or the returned image could be bogus). If that is often the case with this source, set to False

options = ()

A list of `Option` objects. They will be used to automatically construct the configuration widget for this plugin

config_help_message = None

A string that is displayed at the top of the config widget for this plugin

can_get_multiple_covers = False

If True this source can return multiple covers for a given query

auto_trim_covers = False

If set to True covers downloaded by this plugin are automatically trimmed.

prefer_results_with_isbn = True

If set to True, and this source returns multiple results for a query, some of which have ISBNs and some of which do not, the results without ISBNs will be ignored

is_configured()

Return False if your plugin needs to be configured before it can be used. For example, it might need a username/password/API key.

customization_help()

Return a string giving help on how to customize this plugin. By default raise a `NotImplementedError`, which indicates that the plugin does not require customization.

If you re-implement this method in your subclass, the user will be asked to enter a string as customization for this plugin. The customization string will be available as `self.site_customization`.

Site customization could be anything, for example, the path to a needed binary on the user's computer.

Parameter

gui – Liefert HTML-Hilfe falls Tue, gibt ansonsten Hilfe als einfachen Text zurück.

config_widget()

Implement this method and *save_settings()* (die Titelseite 263) in your plugin to use a custom configuration dialog, rather then relying on the simple string based default customization.

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.

save_settings(*config_widget*)

Save the settings specified by the user with config_widget.

Parameter

config_widget – The widget returned by *config_widget()* (die Titelseite 263).

get_author_tokens(*authors*, *only_first_author=True*)

Take a list of authors and return a list of tokens useful for an AND search query. This function tries to return tokens in first name middle names last name order, by assuming that if a comma is in the author name, the name is in lastname, other names form.

get_title_tokens(*title*, *strip_joiners=True*, *strip_subtitle=False*)

Take a title and return a list of tokens useful for an AND search query. Excludes connectives(optionally) and punctuation.

split_jobs(*jobs*, *num*)

Split a list of jobs into at most num groups, as evenly as possible

test_fields(*mi*)

Return the first field from self.touched_fields that is null on the mi object

clean_downloaded_metadata(*mi*)

Call this method in your plugin's identify method to normalize metadata before putting the Metadata object into result_queue. You can of course, use a custom algorithm suited to your metadata source.

get_book_url(*identifiers*)

Return a 3-tuple or None. The 3-tuple is of the form: (identifier_type, identifier_value, URL). The URL is the URL for the book identified by identifiers at this source. identifier_type, identifier_value specify the identifier corresponding to the URL. This URL must be browsable to by a human using a browser. It is meant to provide a clickable link for the user to easily visit the books page at this source. If no URL is found, return None. This method must be quick, and consistent, so only implement it if it is possible to construct the URL from a known scheme given identifiers.

get_book_url_name(*idtype*, *idval*, *url*)

Return a human readable name from the return value of 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()* (die Titelseite 263).

get_cached_cover_url(*identifiers*)

Return cached cover URL for the book identified by the identifiers dictionary or None if no such URL exists.

Note that this method must only return validated URLs, i.e. not URLS that could result in a generic cover image or a not found error.

id_from_url(*url*)

Parse a URL and return a tuple of the form: (identifier_type, identifier_value). If the URL does not match the pattern for the metadata source, return None.

identify_results_keygen(*title=None*, *authors=None*, *identifiers={}*)

Return a function that is used to generate a key that can sort Metadata objects by their relevance given a search query (title, authors, identifiers).

These keys are used to sort the results of a call to *identify()* (die Titelseite 264).

For details on the default algorithm see *InternalMetadataCompareKeyGen* (die Titelseite 265). Reimplement this function in your plugin if the default algorithm is not suitable.

identify(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*)

Identify a book by its Title/Author/ISBN/etc.

If identifiers(s) are specified and no match is found and this metadata source does not store all related identifiers (for example, all ISBNs of a book), this method should retry with just the title and author (assuming they were specified).

If this metadata source also provides covers, the URL to the cover should be cached so that a subsequent call to the get covers API with the same ISBN/special identifier does not need to get the cover URL again. Use the caching API for this.

Every Metadata object put into result_queue by this method must have a *source_relevance* attribute that is an integer indicating the order in which the results were returned by the metadata source for this query. This integer will be used by `compare_identify_results()`. If the order is unimportant, set it to zero for every result.

Make sure that any cover/ISBN mapping information is cached before the Metadata object is put into result_queue.

Parameter

- **log** – A log object, use it to output debugging information/errors
- **result_queue** – A result Queue, results should be put into it. Each result is a Metadata object
- **abort** – If abort.is_set() returns True, abort further processing and return as soon as possible
- **title** – Der Titel dieses Buchs, kann auch None sein
- **authors** – Eine Liste von Autoren dieses Buchs, kann auch None sein
- **identifiers** – A dictionary of other identifiers, most commonly {‚isbn‘:‘1234…‘}
- **timeout** – Timeout in seconds, no network request should hang for longer than timeout.

Rückgabe
None falls keine Fehler aufgetreten sind, andernfalls eine Unicode-Darstellung des Fehlers die dem Benutzer angezeigt werden kann

download_cover(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*, *get_best_cover=False*)

Download a cover and put it into result_queue. The parameters all have the same meaning as for *identify()* (die Titelseite 264). Put (self, cover_data) into result_queue.

This method should use cached cover URLs for efficiency whenever possible. When cached data is not present, most plugins simply call identify and use its results.

If the parameter get_best_cover is True and this plugin can get multiple covers, it should only get the „best“ one.

class calibre.ebooks.metadata.sources.base.**InternalMetadataCompareKeyGen**(*mi*, *source_plugin*, *title*, *authors*, *identifiers*)

Generate a sort key for comparison of the relevance of Metadata objects, given a search query. This is used only to compare results from the same metadata source, not across different sources.

The sort key ensures that an ascending order sort is a sort by order of decreasing relevance.

Der Algorithmus ist:

- Prefer results that have at least one identifier the same as for the query
- Prefer results with a cached cover URL
- Prefer results with all available fields filled in
- Prefer results with the same language as the current user interface language
- Prefer results that are an exact title match to the query
- Prefer results with longer comments (greater than 10% longer)
- **Use the relevance of the result as reported by the metadata source's search** engine

12.1.6 Conversion plugins

class calibre.customize.conversion.**InputFormatPlugin**(**args*)

Bases: *Plugin* (die Titelseite 256)

InputFormatPlugins are responsible for converting a document into HTML+OPF+CSS+etc. The results of the conversion *must* be encoded in UTF-8. The main action happens in *convert()* (die Titelseite 266).

type = 'Konvertierungseingabe'
Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

can_be_disabled = False
Falls *False*, wird der Benutzer keine Möglichkeit haben, die Erweiterung zu deaktivieren. Bitte mit Vorsicht verwenden.

supported_platforms = ['windows', 'osx', 'linux']
Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

file_types = {}
Set of file types for which this plugin should be run For example: `set(['azw', 'mobi', 'prc'])`

is_image_collection = False

If True, this input plugin generates a collection of images, one per HTML file. This can be set dynamically, in the convert method if the input files can be both image collections and non-image collections. If you set this to True, you must implement the get_images() method that returns a list of images.

core_usage = 1

Number of CPU cores used by this plugin. A value of -1 means that it uses all available cores

for_viewer = False

If set to True, the input plugin will perform special processing to make its output suitable for viewing

output_encoding = 'utf-8'

The encoding that this input plugin creates files in. A value of None means that the encoding is undefined and must be detected individually

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Options shared by all Input format plugins. Do not override in sub-classes. Use *options* (die Titelseite 266) instead. Every option must be an instance of `OptionRecommendation`.

options = {}

Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = {}

A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

get_images()

Return a list of absolute paths to the images, if this input plugin represents an image collection. The list of images is in the same order as the spine and the TOC.

convert(*stream*, *options*, *file_ext*, *log*, *accelerators*)

This method must be implemented in sub-classes. It must return the path to the created OPF file or an `OEBBook` instance. All output should be contained in the current folder. If this plugin creates files outside the current folder they must be deleted/marked for deletion before this method returns.

Parameter

- **stream** – A file like object that contains the input file.
- **options** – Options to customize the conversion process. Guaranteed to have attributes corresponding to all the options declared by this plugin. In addition, it will have a verbose attribute that takes integral values from zero upwards. Higher numbers mean be more verbose. Another useful attribute is `input_profile` that is an instance of `calibre.customize.profiles.InputProfile`.
- **file_ext** – The extension (without the .) of the input file. It is guaranteed to be one of the *file_types* supported by this plugin.
- **log** – A `calibre.utils.logging.Log` object. All output should use this object.
- **accelerators** – A dictionary of various information that the input plugin can get easily that would speed up the subsequent stages of the conversion.

postprocess_book(*oeb*, *opts*, *log*)

Called to allow the input plugin to perform postprocessing after the book has been parsed.

specialize(*oeb*, *opts*, *log*, *output_fmt*)

Called to allow the input plugin to specialize the parsed book for a particular output format. Called after postprocess_book and before any transforms are performed on the parsed book.

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin input plugins for examples.

class calibre.customize.conversion.**OutputFormatPlugin**(**args*)

Bases: *Plugin* (die Titelseite 256)

OutputFormatPlugins are responsible for converting an OEB document (OPF+HTML) into an output e-book.

The OEB document can be assumed to be encoded in UTF-8. The main action happens in *convert()* (die Titelseite 267).

type = 'Konvertierungsausgabe'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

can_be_disabled = False

Falls *False*, wird der Benutzer keine Möglichkeit haben, die Erweiterung zu deaktivieren. Bitte mit Vorsicht verwenden.

supported_platforms = ['windows', 'osx', 'linux']

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

file_type = None

Der Dateityp (Dateierweiterung ohne führenden Punkt), den diese Erweiterung ausgibt

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Options shared by all Input format plugins. Do not override in sub-classes. Use *options* (die Titelseite 267) instead. Every option must be an instance of `OptionRecommendation`.

options = {}

Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = {}

A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

property description

str(object='') -> str str(bytes_or_buffer[, encoding[, errors]]) -> str

Create a new string object from the given object. If encoding or errors is specified, then the object must expose a data buffer that will be decoded using the given encoding and error handler. Otherwise, returns the result of object.__str__() (if defined) or repr(object). encoding defaults to ‚utf-8'. errors defaults to ‚strict'.

convert(*oeb_book*, *output*, *input_plugin*, *opts*, *log*)

Render the contents of *oeb_book* (which is an instance of `calibre.ebooks.oeb.OEBBook`) to the file specified by output.

Parameter

- **output** – Either a file like object or a string. If it is a string it is the path to a folder that may or may not exist. The output plugin should write its output into that folder. If it is a file like object, the output plugin should write its output into the file.
- **input_plugin** – The input plugin that was used at the beginning of the conversion pipeline.
- **opts** – Conversion options. Guaranteed to have attributes corresponding to the OptionRecommendations of this plugin.
- **log** – The logger. Print debug/info messages etc. using this.

specialize_options(*log*, *opts*, *input_fmt*)

Can be used to change the values of conversion options, as used by the conversion pipeline.

specialize_css_for_output(*log*, *opts*, *item*, *stylizer*)

Can be used to make changes to the CSS during the CSS flattening process.

Parameter

- **item** – The item (HTML file) being processed
- **stylizer** – A Stylizer object containing the flattened styles for item. You can get the style for any element by stylizer.style(element).

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin output plugins for examples.

12.1.7 Gerätetreiber

The base class for all device drivers is *DevicePlugin* (die Titelseite 268). However, if your device exposes itself as a USBMS drive to the operating system, you should use the USBMS class instead as it implements all the logic needed to support these kinds of devices.

class calibre.devices.interface.**DevicePlugin**(*plugin_path*)

Bases: *Plugin* (die Titelseite 256)

Defines the interface that should be implemented by backends that communicate with an e-book reader.

type = 'Geräteschnittstelle'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

FORMATS = ['lrf', 'rtf', 'pdf', 'txt']

Geordnete Liste unterstützter Formate

VENDOR_ID = 0

VENDOR_ID can be either an integer, a list of integers or a dictionary If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

An integer or a list of integers

BCD = None

BCD can be either None to not distinguish between devices based on BCD, or it can be a list of the BCD numbers of all devices supported by this driver.

THUMBNAIL_HEIGHT = 68

Höhe der Vorschaubilder auf dem Gerät

THUMBNAIL_COMPRESSION_QUALITY = 75

Compression quality for thumbnails. Set this closer to 100 to have better quality thumbnails with fewer compression artifacts. Of course, the thumbnails get larger as well.

WANTS_UPDATED_THUMBNAILS = False

Set this to True if the device supports updating cover thumbnails during sync_booklists. Setting it to true will ask device.py to refresh the cover thumbnails during book matching

CAN_SET_METADATA = ['title', 'authors', 'collections']

Whether the metadata on books can be set via the GUI.

CAN_DO_DEVICE_DB_PLUGBOARD = False

Whether the device can handle device_db metadata plugboards

path_sep = '/'

Path separator for paths to books on device

icon = 'reader.png'

Symbol für dieses Gerät

UserAnnotation

alias of `Annotation`

OPEN_FEEDBACK_MESSAGE = None

GUI displays this as a message if not None in the status bar. Useful if opening can take a long time

VIRTUAL_BOOK_EXTENSIONS = frozenset({})

Set of extensions that are „virtual books“ on the device and therefore cannot be viewed/saved/added to library. For example: `frozenset(['kobo'])`

VIRTUAL_BOOK_EXTENSION_MESSAGE = None

Message to display to user for virtual book extensions.

NUKE_COMMENTS = None

Whether to nuke comments in the copy of the book sent to the device. If not None this should be short string that the comments will be replaced by.

MANAGES_DEVICE_PRESENCE = False

If True indicates that this driver completely manages device detection, ejecting and so forth. If you set this to True, you *must* implement the detect_managed_devices and debug_managed_device_detection methods. A driver with this set to true is responsible for detection of devices, managing a blacklist of devices, a list of ejected devices and so forth. calibre will periodically call the detect_managed_devices() method and if it returns a detected device, calibre will call open(). open() will be called every time a device is returned even if previous calls to open() failed, therefore the driver must maintain its own blacklist of failed devices. Similarly, when ejecting, calibre will call eject() and then assuming the next call to detect_managed_devices() returns None, it will call post_yank_cleanup().

SLOW_DRIVEINFO = False

If set the True, calibre will call the *get_driveinfo()* (die Titelseite 271) method after the books lists have been loaded to get the driveinfo.

ASK_TO_ALLOW_CONNECT = False

If set to True, calibre will ask the user if they want to manage the device with calibre, the first time it is detected. If you set this to True you must implement *get_device_uid()* (die Titelseite 274) and *ignore_connected_device()* (die Titelseite 274) and *get_user_blacklisted_devices()* (die Titelseite 274) and *set_user_blacklisted_devices()* (die Titelseite 274)

user_feedback_after_callback = None

Set this to a dictionary of the form {‚title‘:title, ‚msg‘:msg, ‚det_msg‘:detailed_msg} to have calibre popup a message to the user after some callbacks are run (currently only upload_books). Be careful to not spam the user with too many messages. This variable is checked after *every* callback, so only set it when you really need to.

classmethod get_open_popup_message()

GUI displays this as a non-modal popup. Should be an instance of OpenPopupMessage

classmethod model_metadata() → tuple[ModelMetadata, ...]

Metadata about all the actual device models this driver supports

is_usb_connected(*devices_on_system*, *debug=False*, *only_presence=False*)

Return True, device_info if a device handled by this plugin is currently connected.

Parameter

devices_on_system – Liste der derzeit verbundenen Geräte

detect_managed_devices(*devices_on_system*, *force_refresh=False*)

Wird nur aufgerufen falls MANAGES_DEVICE_PRESENCE True ist.

Scan for devices that this driver can handle. Should return a device object if a device is found. This object will be passed to the open() method as the connected_device. If no device is found, return None. The returned object can be anything, calibre does not use it, it is only passed to open().

This method is called periodically by the GUI, so make sure it is not too resource intensive. Use a cache to avoid repeatedly scanning the system.

Parameter

- **devices_on_system** – Set of USB devices found on the system.
- **force_refresh** – If True and the driver uses a cache to prevent repeated scanning, the cache must be flushed.

debug_managed_device_detection(*devices_on_system*, *output*)

Wird nur aufgerufen falls MANAGES_DEVICE_PRESENCE True ist.

Should write information about the devices detected on the system to output, which is a file like object.

Gibt wahr wieder, wenn ein Gerät gefunden und erfolgreich geöffnet wurde, sonst falsch.

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Parameter

- **key** – Der Schlüssel zum Entsperren des Geräts
- **log_packets** – If true the packet stream to/from the device is logged
- **report_progress** – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information
- **detected_device** – Device information from the device scanner

can_handle_windows(*usbdevice*, *debug=False*)

Optional method to perform further checks on a device to see if this driver is capable of handling it. If it is not it should return False. This method is only called after the vendor, product ids and the bcd have matched, so it can do some relatively time intensive checks. The default implementation returns True. This method is called only on Windows. See also *can_handle()* (die Titelseite 270).

Note that for devices based on USBMS this method by default delegates to *can_handle()* (die Titelseite 270). So you only need to override *can_handle()* (die Titelseite 270) in your subclass of USBMS.

Parameter

usbdevice – A usbdevice as returned by `calibre.devices.winusb.scan_usb_devices()`

can_handle(*device_info*, *debug=False*)

Unix-Version von *can_handle_windows()* (die Titelseite 270).

Parameter

device_info – Is a tuple of (vid, pid, bcd, manufacturer, product, serial number)

open(*connected_device*, *library_uuid*)

Perform any device specific initialization. Called after the device is detected but before any other functions that communicate with the device. For example: For devices that present themselves as USB Mass storage devices, this method would be responsible for mounting the device or if the device has been automounted, for finding out where it has been mounted. The method *calibre.devices.usbms.device.Device.open()* (die Titelseite 278) has an implementation of this function that should serve as a good example for USB Mass storage devices.

This method can raise an OpenFeedback exception to display a message to the user.

Parameter

- **connected_device** – The device that we are trying to open. It is a tuple of (vendor id, product id, bcd, manufacturer name, product name, device serial number). However, some devices have no serial number and on Windows only the first three fields are present, the rest are None.
- **library_uuid** – The UUID of the current calibre library. Can be None if there is no library (for example when used from the command line).

eject()

Un-mount / eject the device from the OS. This does not check if there are pending GUI jobs that need to communicate with the device.

NOTE: That this method may not be called on the same thread as the rest of the device methods.

post_yank_cleanup()

Called if the user yanks the device without ejecting it first.

set_progress_reporter(*report_progress*)

Set a function to report progress information.

Parameter

report_progress – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information

get_device_information(*end_session=True*)

Ask device for device information. See L{DeviceInfoQuery}.

Rückgabe

(device name, device version, software version on device, MIME type) The tuple can optionally have a fifth element, which is a drive information dictionary. See usbms.driver for an example.

get_driveinfo()

Return the driveinfo dictionary. Usually called from get_device_information(), but if loading the driveinfo is slow for this driver, then it should set SLOW_DRIVEINFO. In this case, this method will be called by calibre after the book lists have been loaded. Note that it is not called on the device thread, so the driver should cache the drive info in the books() method and this function should return the cached data.

card_prefix(*end_session=True*)

Return a 2 element list of the prefix to paths on the cards. If no card is present None is set for the card's prefix. E.G. (‚/place', ‚/place2') (None, ‚place2') (‚place', None) (None, None)

total_space(*end_session=True*)

Gibt den kompletten Speicherplatz der auf den folgenden Einhängepunkten verfügbar ist zurück:

1. Hauptspeicher
2. Speicherkarte A
3. Speicherkarte B

Rückgabe
A 3 element list with total space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return 0.

free_space(*end_session=True*)

Gibt den freien Speicherplatz der auf den folgenden Einhängepunkten verfügbar ist zurück:

1. Hauptspeicher
2. Karte A
3. Karte B

Rückgabe
A 3 element list with free space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return -1.

books(*oncard=None*, *end_session=True*)

Return a list of e-books on the device.

Parameter
oncard – If ‚carda' or ‚cardb' return a list of e-books on the specific storage card, otherwise return list of e-books in main memory of device. If a card is specified and no books are on the card return empty list.

Rückgabe
Eine BuchListe.

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)

Upload a list of books to the device. If a file already exists on the device, it should be replaced. This method should raise a `FreeSpaceError` if there is not enough free space on the device. The text of the FreeSpaceError must contain the word „card" if `on_card` is not None otherwise it must contain the word „memory".

Parameter

- **files** – Eine Liste mit Pfaden
- **names** – A list of file names that the books should have once uploaded to the device. len(names) == len(files)
- **metadata** – If not None, it is a list of `Metadata` objects. The idea is to use the metadata to determine where on the device to put the book. len(metadata) == len(files). Apart from the regular cover (path to cover), there may also be a thumbnail attribute, which should be used in preference. The thumbnail attribute is of the form (width, height, cover_data as jpeg).

Rückgabe
A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (die Titelseite 272).

classmethod add_books_to_metadata(*locations*, *metadata*, *booklists*)

Add locations to the booklists. This function must not communicate with the device.

Parameter

- **locations** – Ergebnis eines Aufrufs von L{upload_books}
- **metadata** – List of `Metadata` objects, same as for *upload_books()* (die Titelseite 272).
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

delete_books(*paths*, *end_session=True*)

Bücher unter Pfaden vom Gerät löschen.

classmethod remove_books_from_metadata(*paths*, *booklists*)

Remove books from the metadata list. This function must not communicate with the device.

Parameter

- **paths** – Pfade zu Büchern auf dem Gerät.
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

sync_booklists(*booklists*, *end_session=True*)

Metadaten auf dem Gerät aktualisieren.

Parameter
booklists – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

get_file(*path*, *outfile*, *end_session=True*)

Read the file at `path` on the device and write it to outfile.

Parameter
outfile – file object like `sys.stdout` or the result of an *open()* (die Titelseite 271) call.

classmethod config_widget()

Should return a QWidget. The QWidget contains the settings for the device interface

classmethod save_settings(*settings_widget*)

Should save settings to disk. Takes the widget created in *config_widget()* (die Titelseite 273) and saves all settings to disk.

classmethod settings()

Should return an opts object. The opts object should have at least one attribute *format_map* which is an ordered list of formats for the device.

set_plugboards(*plugboards*, *pb_func*)

provide the driver the current set of plugboards and a function to select a specific plugboard. This method is called immediately before add_books and sync_booklists.

pb_func is a callable with the following signature::
def pb_func(device_name, format, plugboards)

You give it the current device name (either the class name or DEVICE_PLUGBOARD_NAME), the format you are interested in (a ‚real' format or ‚device_db'), and the plugboards (you were given those by set_plugboards, the same place you got this method).

Rückgabe
None or a single plugboard instance.

set_driveinfo_name(*location_code*, *name*)

Set the device name in the driveinfo file to ‚name'. This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the get_device_information() method.

prepare_addable_books(*paths*)

Given a list of paths, returns another list of paths. These paths point to addable versions of the books.

If there is an error preparing a book, then instead of a path, the position in the returned list for that book should be a three tuple: (original_path, the exception instance, traceback)

startup()

Called when calibre is starting the device. Do any initialization required. Note that multiple instances of the class can be instantiated, and thus __init__ can be called multiple times, but only one instance will have this method called. This method is called on the device thread, not the GUI thread.

shutdown()

Called when calibre is shutting down, either for good or in preparation to restart. Do any cleanup required. This method is called on the device thread, not the GUI thread.

get_device_uid()

Must return a unique id for the currently connected device (this is called immediately after a successful call to open()). You must implement this method if you set ASK_TO_ALLOW_CONNECT = True

ignore_connected_device(*uid*)

Should ignore the device identified by uid (the result of a call to get_device_uid()) in the future. You must implement this method if you set ASK_TO_ALLOW_CONNECT = True. Note that this function is called immediately after open(), so if open() caches some state, the driver should reset that state.

get_user_blacklisted_devices()

Return map of device uid to friendly name for all devices that the user has asked to be ignored.

set_user_blacklisted_devices(*devices*)

Set the list of device uids that should be ignored by this driver.

specialize_global_preferences(*device_prefs*)

Implement this method if your device wants to override a particular preference. You must ensure that all call sites that want a preference that can be overridden use device_prefs[‚something'] instead of prefs[‚something']. Your method should call device_prefs.set_overrides(pref=val, pref=val, …). Currently used for: metadata management (prefs[‚manage_device_metadata'])

set_library_info(*library_name*, *library_uuid*, *field_metadata*)

Implement this method if you want information about the current calibre library. This method is called at startup and when the calibre library changes while connected.

is_dynamically_controllable()

Called by the device manager when starting plugins. If this method returns a string, then a) it supports the device manager's dynamic control interface, and b) that name is to be used when talking to the plugin.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

start_plugin()

This method is called to start the plugin. The plugin should begin to accept device connections however it does that. If the plugin is already accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

stop_plugin()

This method is called to stop the plugin. The plugin should no longer accept connections, and should cleanup behind itself. It is likely that this method should call shutdown. If the plugin is already not accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

get_option(*opt_string*, *default=None*)

Return the value of the option indicated by opt_string. This method can be called when the plugin is not started. Return None if the option does not exist.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

set_option(*opt_string*, *opt_value*)

Set the value of the option indicated by opt_string. This method can be called when the plugin is not started.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

is_running()

Return True if the plugin is started, otherwise false

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

synchronize_with_db(*db*, *book_id*, *book_metadata*, *first_call*)

Called during book matching when a book on the device is matched with a book in calibre's db. The method is responsible for synchronizing data from the device to calibre's db (if needed).

The method must return a two-value tuple. The first value is a set of calibre book ids changed if calibre's database was changed or None if the database was not changed. If the first value is an empty set then the metadata for the book on the device is updated with calibre's metadata and given back to the device, but no GUI refresh of that book is done. This is useful when the calibre data is correct but must be sent to the device.

The second value is itself a 2-value tuple. The first value in the tuple specifies whether a book format should be sent to the device. The intent is to permit verifying that the book on the device is the same as the book in calibre. This value must be None if no book is to be sent, otherwise return the base file name on the device (a string like foobar.epub). Be sure to include the extension in the name. The device subsystem will construct a send_books job for all books with not- None returned values. Note: other than to later retrieve the extension, the name is ignored in cases where the device uses a template to generate the file name, which most do. The second value in the returned tuple indicated whether the format is future-dated. Return True if it is, otherwise return False. calibre will display a dialog to the user listing all future dated books.

Extremely important: this method is called on the GUI thread. It must be threadsafe with respect to the device manager's thread.

book_id: the calibre id for the book in the database. book_metadata: the Metadata object for the book coming from the device. first_call: True if this is the first call during a sync, False otherwise

class calibre.devices.interface.**BookList**(*oncard*, *prefix*, *settings*)

Bases: list

A list of books. Each Book object must have the fields

1. Titel
2. Autoren
3. Größe (Dateigröße des Buches)
4. datetime (a UTC time tuple)
5. path (path on the device to the book)

6. thumbnail (can be None) thumbnail is either a str/bytes object with the image data or it should have an attribute image_path that stores an absolute (platform native) path to the image
7. tags (a list of strings, can be empty).

supports_collections()

Return True if the device supports collections for this book list.

add_book(*book*, *replace_metadata*)

Add the book to the booklist. Intent is to maintain any device-internal metadata. Return True if booklists must be sync'ed

remove_book(*book*)

Remove a book from the booklist. Correct any device metadata at the same time

get_collections(*collection_attributes*)

Return a dictionary of collections created from collection_attributes. Each entry in the dictionary is of the form collection name:[list of books]

The list of books is sorted by book title, except for collections created from series, in which case series_index is used.

Parameter

collection_attributes – Eine Liste der Attribute des Buchobjektes

USB-Massenspeicher-basierte Geräte

The base class for such devices is *calibre.devices.usbms.driver.USBMS* (die Titelseite 279). This class in turn inherits some of its functionality from its bases, documented below. A typical basic USBMS based driver looks like this:

```
from calibre.devices.usbms.driver import USBMS

class PDNOVEL(USBMS):
    name = 'Pandigital Novel device interface'
    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*)

Bases: DeviceConfig, *DevicePlugin* (die Titelseite 268)

This class provides logic common to all drivers for devices that export themselves as USB Mass Storage devices. Provides implementations for mounting/ejecting of USBMS devices on all platforms.

VENDOR_ID = 0

VENDOR_ID can be either an integer, a list of integers or a dictionary If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

An integer or a list of integers

BCD = None

BCD can be either None to not distinguish between devices based on BCD, or it can be a list of the BCD numbers of all devices supported by this driver.

WINDOWS_MAIN_MEM = None

String identifying the main memory of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_A_MEM = None

String identifying the first card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_B_MEM = None

String identifying the second card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

OSX_MAIN_MEM_VOL_PAT = None

Used by the new driver detection to disambiguate main memory from storage cards. Should be a regular expression that matches the main memory mount point assigned by macOS

MAX_PATH_LEN = 250

The maximum length of paths created on the device

NEWS_IN_FOLDER = True

Put news in its own folder

classmethod model_metadata() → tuple[ModelMetadata, ...]

Metadata about all the actual device models this driver supports

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Parameter

- **key** – Der Schlüssel zum Entsperren des Geräts
- **log_packets** – If true the packet stream to/from the device is logged
- **report_progress** – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information
- **detected_device** – Device information from the device scanner

set_progress_reporter(*report_progress*)

Set a function to report progress information.

Parameter

report_progress – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information

card_prefix(*end_session=True*)

Return a 2 element list of the prefix to paths on the cards. If no card is present None is set for the card's prefix. E.G. (‚/place', ‚/place2') (None, ‚place2') (‚place', None) (None, None)

total_space(*end_session=True*)

Gibt den kompletten Speicherplatz der auf den folgenden Einhängepunkten verfügbar ist zurück:

1. Hauptspeicher
2. Speicherkarte A
3. Speicherkarte B

Rückgabe

A 3 element list with total space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return 0.

free_space(*end_session=True*)

Gibt den freien Speicherplatz der auf den folgenden Einhängepunkten verfügbar ist zurück:

1. Hauptspeicher
2. Karte A
3. Karte B

Rückgabe

A 3 element list with free space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return -1.

windows_sort_drives(*drives*)

Called to disambiguate main memory and storage card for devices that do not distinguish between them on the basis of *WINDOWS_CARD_NAME*. For example: The EB600

can_handle_windows(*usbdevice*, *debug=False*)

Optional method to perform further checks on a device to see if this driver is capable of handling it. If it is not it should return False. This method is only called after the vendor, product ids and the bcd have matched, so it can do some relatively time intensive checks. The default implementation returns True. This method is called only on Windows. See also `can_handle()`.

Note that for devices based on USBMS this method by default delegates to `can_handle()`. So you only need to override `can_handle()` in your subclass of USBMS.

Parameter

usbdevice – A usbdevice as returned by `calibre.devices.winusb.scan_usb_devices()`

open(*connected_device*, *library_uuid*)

Perform any device specific initialization. Called after the device is detected but before any other functions that communicate with the device. For example: For devices that present themselves as USB Mass storage devices, this method would be responsible for mounting the device or if the device has been automounted, for finding out where it has been mounted. The method *calibre.devices.usbms.device.Device.open()* (die Titelseite 278) has an implementation of this function that should serve as a good example for USB Mass storage devices.

This method can raise an OpenFeedback exception to display a message to the user.

Parameter

- **connected_device** – The device that we are trying to open. It is a tuple of (vendor id, product id, bcd, manufacturer name, product name, device serial number). However, some devices have no serial number and on Windows only the first three fields are present, the rest are None.
- **library_uuid** – The UUID of the current calibre library. Can be None if there is no library (for example when used from the command line).

eject()

Un-mount / eject the device from the OS. This does not check if there are pending GUI jobs that need to communicate with the device.

NOTE: That this method may not be called on the same thread as the rest of the device methods.

post_yank_cleanup()

Called if the user yanks the device without ejecting it first.

sanitize_callback(*path*)

Callback to allow individual device drivers to override the path sanitization used by `create_upload_path()`.

filename_callback(*default*, *mi*)

Callback to allow drivers to change the default file name set by `create_upload_path()`.

sanitize_path_components(*components*)

Perform any device specific sanitization on the path components for files to be uploaded to the device

get_annotations(*path_map*)

Resolve path_map to annotation_map of files found on the device

add_annotation_to_library(*db*, *db_id*, *annotation*)

Add an annotation to the calibre library

class calibre.devices.usbms.cli.**CLI**

class calibre.devices.usbms.driver.**USBMS**(*plugin_path*)

Bases: *CLI* (die Titelseite 279), *Device* (die Titelseite 276)

The base class for all USBMS devices. Implements the logic for sending/getting/updating metadata/caching metadata/etc.

description = 'Kommunikation mit einem eBook-Reader.'

Eine kurze Beschreibung, was diese Erweiterung macht.

author = 'John Schember'

Der Autor dieser Erweiterung

supported_platforms = ['windows', 'osx', 'linux']

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

booklist_class

alias of `BookList`

book_class

alias of `Book`

FORMATS = []

Geordnete Liste unterstützter Formate

CAN_SET_METADATA = []

Whether the metadata on books can be set via the GUI.

get_device_information(*end_session=True*)

Ask device for device information. See L{DeviceInfoQuery}.

Rückgabe

(device name, device version, software version on device, MIME type) The tuple can optionally have a fifth element, which is a drive information dictionary. See usbms.driver for an example.

set_driveinfo_name(*location_code*, *name*)

Set the device name in the driveinfo file to ‚name'. This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the get_device_information() method.

books(*oncard=None*, *end_session=True*)

Return a list of e-books on the device.

Parameter

oncard – If ‚carda' or ‚cardb' return a list of e-books on the specific storage card, otherwise return list of e-books in main memory of device. If a card is specified and no books are on the card return empty list.

Rückgabe

Eine BuchListe.

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)

Upload a list of books to the device. If a file already exists on the device, it should be replaced. This method should raise a `FreeSpaceError` if there is not enough free space on the device. The text of the FreeSpaceError must contain the word „card" if `on_card` is not None otherwise it must contain the word „memory".

Parameter

- **files** – Eine Liste mit Pfaden
- **names** – A list of file names that the books should have once uploaded to the device. len(names) == len(files)
- **metadata** – If not None, it is a list of `Metadata` objects. The idea is to use the metadata to determine where on the device to put the book. len(metadata) == len(files). Apart from the regular cover (path to cover), there may also be a thumbnail attribute, which should be used in preference. The thumbnail attribute is of the form (width, height, cover_data as jpeg).

Rückgabe

A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (die Titelseite 281).

upload_cover(*path*, *filename*, *metadata*, *filepath*)

Upload book cover to the device. Default implementation does nothing.

Parameter

- **path** – The full path to the folder where the associated book is located.
- **filename** – Der Name der Buchdatei ohne die Endung.
- **metadata** – metadata belonging to the book. Use metadata.thumbnail for cover
- **filepath** – Der vollständige Pfad zur eBook-Datei

add_books_to_metadata(*locations*, *metadata*, *booklists*)

Add locations to the booklists. This function must not communicate with the device.

Parameter

- **locations** – Ergebnis eines Aufrufs von L{upload_books}
- **metadata** – List of `Metadata` objects, same as for *upload_books()* (die Titelseite 280).
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

delete_books(*paths*, *end_session=True*)

Bücher unter Pfaden vom Gerät löschen.

remove_books_from_metadata(*paths*, *booklists*)

Remove books from the metadata list. This function must not communicate with the device.

Parameter

- **paths** – Pfade zu Büchern auf dem Gerät.
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

sync_booklists(*booklists*, *end_session=True*)

Metadaten auf dem Gerät aktualisieren.

Parameter

booklists – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

classmethod normalize_path(*path*)

Return path with platform native path separators

12.1.8 User interface actions

If you are adding your own plugin in a ZIP file, you should subclass both InterfaceActionBase and InterfaceAction. The `load_actual_plugin()` method of your InterfaceActionBase subclass must return an instantiated object of your InterfaceBase subclass.

class `calibre.gui2.actions.`**InterfaceAction**(*parent*, *site_customization*)

Bases: `QObject`

A plugin representing an „action“ that can be taken in the graphical user interface. All the items in the toolbar and context menus are implemented by these plugins.

Note that this class is the base class for these plugins, however, to integrate the plugin with calibre's plugin system, you have to make a wrapper class that references the actual plugin. See the `calibre.customize.builtins` module for examples.

If two *InterfaceAction* (die Titelseite 281) objects have the same name, the one with higher priority takes precedence.

Sub-classes should implement the *genesis()* (die Titelseite 284), *library_changed()* (die Titelseite 284), *location_selected()* (die Titelseite 284), *shutting_down()* (die Titelseite 284), *initialization_complete()* (die Titelseite 284) and *tag_browser_context_action()* (die Titelseite 284) methods.

Once initialized, this plugin has access to the main calibre GUI via the `gui` member. You can access other plugins by name, for example:

```
self.gui.iactions['Save To Disk']
```

To access the actual plugin, use the `interface_action_base_plugin` attribute, this attribute only becomes available after the plugin has been initialized. Useful if you want to use methods from the plugin class like do_user_config().

The QAction specified by *action_spec* (die Titelseite 282) is automatically create and made available as `self.qaction`.

name = 'Implement me'

The plugin name. If two plugins with the same name are present, the one with higher priority takes precedence.

priority = 1

The plugin priority. If two plugins with the same name are present, the one with higher priority takes precedence.

popup_type = 1

The menu popup type for when this plugin is added to a toolbar

auto_repeat = False

Whether this action should be auto repeated when its shortcut key is held down.

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

If True, a menu is automatically created and added to self.qaction

action_menu_clone_qaction = False

If True, a clone of self.qaction is added to the menu of self.qaction If you want the text of this action to be different from that of self.qaction, set this variable to the new text

dont_add_to = frozenset({})

Set of locations to which this action must not be added. See `all_locations` for a list of possible locations

dont_remove_from = frozenset({})

Set of locations from which this action must not be removed. See `all_locations` for a list of possible locations

action_type = 'global'

Type of action ‚current' means acts on the current view ‚global' means an action that does not act on the current view, but rather on calibre as a whole

accepts_drops = False

If True, then this InterfaceAction will have the opportunity to interact with drag and drop events. See the methods, *accept_enter_event()* (die Titelseite 283), :meth`:accept_drag_move_event`, *drop_event()* (die Titelseite 283) for details.

accept_enter_event(*event*, *mime_data*)

This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

accept_drag_move_event(*event*, *mime_data*)

This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

drop_event(*event*, *mime_data*)

This method should perform some useful action and return True iff this interface action is capable of handling the drop event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI. You should not perform blocking/long operations in this function. Instead emit a signal or use QTimer.singleShot and return quickly. See the builtin actions for examples.

create_menu_action(*menu*, *unique_name*, *text*, *icon=None*, *shortcut=None*, *description=None*, *triggered=None*, *shortcut_name=None*, *persist_shortcut=False*)

Convenience method to easily add actions to a QMenu. Returns the created QAction. This action has one extra attribute calibre_shortcut_unique_name which if not None refers to the unique name under which this action is registered with the keyboard manager.

Parameter

- **menu** – The QMenu the newly created action will be added to
- **unique_name** – A unique name for this action, this must be globally unique, so make it as descriptive as possible. If in doubt, add an UUID to it.
- **text** – Der Text der Aktion.
- **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** – A string, a list of strings, None or False. If False, no keyboard shortcut is registered for this action. If None, a keyboard shortcut with no default keybinding is registered. String and list of strings register a shortcut with default keybinding as specified.
- **description** – A description for this action. Used to set tooltips.
- **triggered** – A callable which is connected to the triggered signal of the created action.
- **shortcut_name** – The text displayed to the user when customizing the keyboard shortcuts for this action. By default it is set to the value of `text`.
- **persist_shortcut** – Shortcuts for actions that don't always appear, or are library dependent, may disappear when other keyboard shortcuts are edited unless `persist_shortcut` is set True.

load_resources(*names*)

If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

Zum Beispiel, um ein Bild zu laden:

```
pixmap = QPixmap()
pixmap.loadFromData(tuple(self.load_resources(['images/icon.png']).
↪values())[0])
icon = QIcon(pixmap)
```

Parameter
names – List of paths to resources in the ZIP file using / as separator

Rückgabe
A dictionary of the form `{name : file_contents}`. Any names that were not found in the ZIP file will not be present in the dictionary.

genesis()

Setup this plugin. Only called once during initialization. self.gui is available. The action specified by *action_spec* (die Titelseite 282) is available as `self.qaction`.

location_selected(*loc*)

Called whenever the book list being displayed in calibre changes. Currently values for loc are: `library, main, card and cardb`.

This method should enable/disable this action and its sub actions as appropriate for the location.

library_about_to_change(*olddb*, *db*)

Called whenever the current library is changed.

Parameter

- **olddb** – The LibraryDatabase corresponding to the previous library.
- **db** – The LibraryDatabase corresponding to the new library.

library_changed(*db*)

Called whenever the current library is changed.

Parameter
db – The LibraryDatabase corresponding to the current library.

gui_layout_complete()

Called once per action when the layout of the main GUI is completed. If your action needs to make changes to the layout, they should be done here, rather than in *initialization_complete()* (die Titelseite 284).

initialization_complete()

Called once per action when the initialization of the main GUI is completed.

tag_browser_context_action(*index*)

Called when displaying the context menu in the Tag browser. `index` is the QModelIndex that points to the Tag browser item that was right clicked. Test it for validity with index.valid() and get the underlying TagTreeItem object with index.data(Qt.ItemDataRole.UserRole). Any action objects yielded by this method will be added to the context menu.

shutting_down()

Called once per plugin when the main GUI is in the process of shutting down. Release any used resources, but try not to block the shutdown for long periods of time.

class calibre.customize.**InterfaceActionBase**(**args*, ***kwargs*)

Bases: *Plugin* (die Titelseite 256)

supported_platforms = ['windows', 'osx', 'linux']

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

author = 'Kovid Goyal'

Der Autor dieser Erweiterung

type = 'Benutzeroberfläche'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

can_be_disabled = False

Falls *False*, wird der Benutzer keine Möglichkeit haben, die Erweiterung zu deaktivieren. Bitte mit Vorsicht verwenden.

load_actual_plugin(*gui*)

This method must return the actual interface action plugin object.

12.1.9 Preferences plugins

class calibre.customize.**PreferencesPlugin**(*plugin_path*)

Bases: *Plugin* (die Titelseite 256)

A plugin representing a widget displayed in the Preferences dialog.

This plugin has only one important method *create_widget()* (die Titelseite 286). The various fields of the plugin control how it is categorized in the UI.

supported_platforms = ['windows', 'osx', 'linux']

Liste von Plattformen auf denen diese Erweiterung funktioniert. Zum Beispiel: `['windows', 'osx', 'linux']`

author = 'Kovid Goyal'

Der Autor dieser Erweiterung

type = 'Einstellungen'

Der Typ dieser Erweiterung. Wird benutzt, um Erweiterungen in der GUI zu kategorisieren

can_be_disabled = False

Falls *False*, wird der Benutzer keine Möglichkeit haben, die Erweiterung zu deaktivieren. Bitte mit Vorsicht verwenden.

config_widget = None

Import path to module that contains a class named ConfigWidget which implements the ConfigWidgetInterface. Used by *create_widget()* (die Titelseite 286).

category_order = 100

Where in the list of categories the *category* (die Titelseite 285) of this plugin should be.

name_order = 100

Where in the list of names in a category, the *gui_name* (die Titelseite 285) of this plugin should be

category = None

Die Kategorie, in der diese Erweiterung enthalten sein soll

gui_category = None

Der Name der Kategorie, welcher dem Benutzer zu dieser Erweiterung angezeigt wird

gui_name = None

Der Name, welcher dem Benutzer zu dieser Erweiterung angezeigt wird

icon = None

The icon for this plugin, should be an absolute path

description = None

The description used for tooltips and the like

create_widget(*parent=None*)

Create and return the actual Qt widget used for setting this group of preferences. The widget must implement the *calibre.gui2.preferences.ConfigWidgetInterface* (die Titelseite 286).

The default implementation uses *config_widget* (die Titelseite 285) to instantiate the widget.

class calibre.gui2.preferences.**ConfigWidgetInterface**

This class defines the interface that all widgets displayed in the Preferences dialog must implement. See *ConfigWidgetBase* (die Titelseite 287) for a base class that implements this interface and defines various convenience methods as well.

changed_signal = None

This signal must be emitted whenever the user changes a value in this widget

supports_restoring_to_defaults = True

Set to True iff the `restore_to_defaults()` method is implemented.

restore_defaults_desc = 'Einstellungen auf Standardwerte zurücksetzen. Zum tatsächlichen Speichern der Standardeinstellungen muss auf "Anwenden" geklickt werden.'

The tooltip for the „Restore defaults“ button

restart_critical = False

If True the Preferences dialog will not allow the user to set any more preferences. Only has effect if *commit()* (die Titelseite 286) returns True.

genesis(*gui*)

Called once before the widget is displayed, should perform any necessary setup.

Parameter
gui – The main calibre graphical user interface

initialize()

Should set all config values to their initial values (the values stored in the config files). A „return“ statement is optional. Return False if the dialog is not to be shown.

restore_defaults()

Should set all config values to their defaults.

commit()

Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an `AbortCommit` exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

refresh_gui(*gui*)

Called once after this widget is committed. Responsible for causing the gui to reread any changed settings. Note that by default the GUI re-initializes various elements anyway, so most widgets won't need to use this method.

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*)

Base class that contains code to easily add standard config widgets like checkboxes, combo boxes, text fields and so on. See the *register()* (die Titelseite 287) method.

This class automatically handles change notification, resetting to default, translation between gui objects and config objects, etc. for registered settings.

If your config widget inherits from this class but includes setting that are not registered, you should override the *ConfigWidgetInterface* (die Titelseite 286) methods and call the base class methods inside the overrides.

changed_signal

This signal must be emitted whenever the user changes a value in this widget

supports_restoring_to_defaults = True

Set to True iff the restore_to_defaults() method is implemented.

restart_critical = False

If True the Preferences dialog will not allow the user to set any more preferences. Only has effect if *commit()* (die Titelseite 287) returns True.

register(*name*, *config_obj*, *gui_name=None*, *choices=None*, *restart_required=False*, *empty_string_is_None=True*, *setting=<class 'calibre.gui2.preferences.Setting'>*)

Register a setting.

Parameter

- **name** – Der Einstellungsname
- **config_obj** – The config object that reads/writes the setting
- **gui_name** – The name of the GUI object that presents an interface to change the setting. By default it is assumed to be 'opt_' + name.
- **choices** – If this setting is a multiple choice (combobox) based setting, the list of choices. The list is a list of two element tuples of the form: [(gui name, value), ...]
- **setting** – The class responsible for managing this setting. The default class handles almost all cases, so this param is rarely used.

initialize()

Should set all config values to their initial values (the values stored in the config files). A „return“ statement is optional. Return False if the dialog is not to be shown.

commit(**args*)

Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an AbortCommit exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

restore_defaults(**args*)

Should set all config values to their defaults.

12.2 Umgebungsvariablen

- CALIBRE_CONFIG_DIRECTORY - festlegen wo Calibre config - Dateien speichert.
- CALIBRE_TEMP_DIR - festlegen welchen Ordner Calibre als temporären Speicher nutzt

- `CALIBRE_CACHE_DIRECTORY` - festlegen wo Calibre Cache-Daten ablegt
- `CALIBRE_OVERRIDE_DATABASE_PATH` - erlaubt die Angabe des vollständigen Pfads zur metadata.db. Durch Benutzung dieser Variable kann sich die metadata.db an einem anderen Speicherort befinden als im normalerweise vorgesehenen Bibliotheks-Ordner. Dies ist nützlich, wenn Ihr Bibliotheksordner sich auf einem Netzlaufwerk befindet und das Sperren von Dateien nicht unterstützt wird.
- `CALIBRE_DEVELOP_FROM` - genutzt um Calibre aus einer Entwicklungsumgebung zu nutzen. Siehe *Einrichten einer Calibre-Entwicklungsumgebung* (die Titelseite 345).
- `CALIBRE_OVERRIDE_LANG` - genutzt um die Sprache der Oberfläche zu setzen (ISO 639 language code)
- `CALIBRE_TEST_TRANSLATION` - genutzt um Übersetzungsdateien (.po) zu testen. Den Pfad zu einer .po Datei angeben.
- `CALIBRE_NO_NATIVE_FILEDIALOGS` - Calibre nutzt die ‚nativen' Dialoge des Betriebssystems für Datei- und Ordnerdialoge.
- `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` - normaler Weise nutzt Calibre unter Linux seinen eigenen ‚Qt' Stil. Das verhindert Programmabbrüche oder -fehler auf Grund von Inkompatibilitäten zwischen der Calibre und der System Qt Library. Nachteil davon wäre, das Calibre ein etwas anderes ‚Look & Feel' als andere System Programme hat. Durch setzten dieser Variable, nutzt Calibre das Betriebssystem Theme, achten Sie aber auf Abbrüche und/oder Fehler.
- `CALIBRE_SHOW_DEPRECATION_WARNINGS` - Warnungen werden nach stdout geschrieben. Nützlich für Entwickler.
- `CALIBRE_NO_DEFAULT_PROGRAMS` - verhindert, das Calibre automatisch mit Dateitypen verknüpft wird. Das könnte bei der Nutzung unter Windows wichtig sein.
- `CALIBRE_USE_SYSTEM_CERTIFICATES` - make calibre use the system certificate store for SSL certificate verification instead of its own certificate store on Windows and macOS.
- `CALIBRE_NO_ICONS_IN_MENUS` - Disable icons in menus
- `QT_QPA_PLATFORM` - On Linux set this to `wayland` to force calibre to use Wayland and `xcb` to force use of X11.
- `SYSFS_PATH` - zu benutzen, wenn wenn sysfs irgendwo anders als unter /sys eingebunden ist.
- `http_proxy`, `https_proxy` - kann genutzt werden um unter Linux einen HTTP(S) proxy zu nutzen.

Schaue *Umgebungsvariablen in Windows <http://www.winfaq.de/faq_html/Content/tip0000/onlinefaq.php?h=tip0328.htm>* . Wenn Sie macOS nutzen, setzen Sie Environment Variabeln (Umgebungsvariabeln) durch erstellen der Datei `~/Library/Preferences/calibre/macos-env.txt` und schreiben die Variablen Zeile für Zeile dort hinein. Zum Beispiel:

```
CALIBRE_DEVELOP_FROM=$HOME/calibre-src/src
CALIBRE_NO_NATIVE_FILEDIALOGS=1
CALIBRE_CONFIG_DIRECTORY=~/.config/calibre
```

12.3 Optimierungen

Optimierungen oder auch „Tweaks" sind kleine Änderungen, die Sie angeben können, um verschiedene Aspekte von Calibres Verhalten zu steuern. Sie können diese ändern, indem Sie auf Einstellungen → Erweitert → Optimierungen gehen. Die Standardwerte für alle Optimierungen sind nachfolgend beschrieben.

```
#!/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
#   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'
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#  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".
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 and/or series_sort for names in Tag browser
# Set which author or series field is used to display as the item name in the
# Tag browser (the list of authors, series, publishers etc on the left hand
# side). The choices for author are 'author' and 'author_sort'. The choices for
# series are 'series' and 'series_sort'. This tweak affects only what is
# displayed under the category in the Tag browser and Content server. Please
# note that if you set this to …_sort, it is possible to see duplicate names in
# the list because although it is guaranteed that author and series names are
# unique, there is no such guarantee for their sort values. Showing duplicates
# won't break anything but it could lead to some confusion. The tooltip for an
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# item will show the item's name no matter how you set this tweak.
# Examples:
#   categories_use_field_for_author_name = 'author'
#   categories_use_field_for_author_name = 'author_sort'
#   categories_use_field_for_series_name = 'series'
#   categories_use_field_for_series_name = 'series_sort'
categories_use_field_for_author_name = 'author'
categories_use_field_for_series_name = 'series'

#: 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.
↪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)
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#  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␣
↪lowercase localized string for am or pm
#  AP    use a 12-hour clock instead of a 24-hour clock, with "AP" replaced by the␣
↪uppercase 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
#  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
# 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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# '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'", r'L´', r'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': (r'Lo\s+', r'Il\s+', r"L'", r'L´', r'La\s+', r'Gli\s+',
                r'I\s+', r'Le\s+', r'Uno\s+', r'Un\s+', r'Una\s+', "rUn'",
                r'Un´', r'Dei\s+', r'Degli\s+', r'Delle\s+', r'Del\s+',
                r'Della\s+', r'Dello\s+', r"Dell'", r'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
        '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+',
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                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.
#
# There are two related tweaks. The first determines the category name to use
# for a metadata field.  The second is a template, used to determines how the
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# 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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# [ ( [list of fields], sort field ) , ( [ list of fields ] , sort field ) ]
# Default: empty (no rules), so no collection attributes are named.
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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# 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
# 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
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#: 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.
# 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.
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# 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.
# 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'
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#   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
# 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'

#: East Asian language to use for transliteration
# Setting this tweak will make calibre use the specified language as the "base"
# language when transliterating East Asian languages to English. This might be
# useful if you run calibre in English but want text transliterated to
# Japanese instead of Chinese. The valid values are:
#   'ja' for Japanese
#   'kr' for Korean
#   'vn' for Vietnamese
#   'zh' for Chinese
# Any other value will use the language set in calibre preferences as the base
# language. A base language other than those in the above list causes transliteration
# with a base language of Chinese.
# Example: east_asian_base_language = 'ja'
east_asian_base_language = ''
```

12.4 Überschreiben von Symbolen, Vorlagen usw.

Bemerkung

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 ermöglicht es Ihnen statische Ressourcen, wie Icons, JavaScript, Templates, Metadata, Kataloge, usw. mit Ihren eigenen Verisonen zu überschreiben. All diese zu Calibre gehörenden Teile finden Sie in einem Unterordner. Unter Windows ist das normaler Weise `C:\Program Files\Calibre2\app\resources`. Unter macOS finden Sie das unter `/Applications/calibre.app/Contents/Resources/resources/`. Und unter Linux, wenn Sie den Installer von der Calibre Webseite genutzt haben im Ordner `/opt/calibre/resources`. Aber wie gesagt kann der Ort ein anderer sein, je nachdem wie Sie Calibre installiert haben.

Sie solten die Dateien nicht direkt in dem Ordner überschreiben oder ändern, das sie dann beim nächsten Update wieder hergestellt werden. Statt dessen sollten Sie in den Einstellungen bei *Preferences → Advanced → Miscellaneous* auf *Open calibre configuration folder* klicken und da einen neuen Ordner angeben. In diesem erstellen sie dann einen Unterordner mit dem Namen: ‚resources' und darin speichern Sie alle Dateien die Sie ändern möchten. Nutzen Sie entsprechende Unterordner z.B. `resources/images` für Bilder usw. . Nach dem nächsten Start nutzt Calibre automatisch Ihre Dateien anstatt der originalen.

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 Erstellen eines eigenen Symboldesigns für 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[116] 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.

12.6 Anpassen von Calibre durch Erweiterungen

Calibre ist Modular aufgebaut. Fast alle Funktionen in Calibre sind mit Plugin's erstellt. Sie werden zum Umwandeln, zum herunterladen von Newsfeeds (dann nennen wir sie Rezepte - recipes), als Teil der Benutzeroberfläche, zum Verbinden zwischen verschiedenen Komponenten, zum Bearbeiten von Dateien beim Import und so weiter genutzt. Eine komplette Liste aller eingebauten Plugin's sehen Sie unter: guilabel:*Preferences->Advanced->Plugins*.

Sie können Ihre eigenen Erweiterungen schreiben, um das Verhalten von Calibre anzupassen und zu erweitern. Die Architektur von Erweiterungen in Calibre ist sehr einfach, siehe hierzu das Tutorial *Schreibe Deine eigenen Erweiterungen zur Anpassung von Calibres Funktionalität* (die Titelseite 225).

Once you have written a plugin, you can upload that to the calibre plugins forum at Mobileread[117] and it will be made available via calibre's builtin plugin updater.

[116] https://www.mobileread.com/forums/forumdisplay.php?f=166
[117] https://www.mobileread.com/forums/forumdisplay.php?f=237

KAPITEL 13

Befehlszeilenschnittstelle

Bemerkung

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 Dokumentierte Befehle

13.1.1 `calibre`

```
calibre [options] [path_to_ebook or calibre url ...]
```

Launch the main **calibre** Graphical User Interface and optionally add the e-book at path_to_ebook to the database. You can also specify **calibre** URLs to perform various different actions, than just adding books. For example:

calibre://view-book/test_library/1842/epub

Will open the book with id 1842 in the EPUB format from the library „test_library“ in the **calibre** E-book viewer. Library names are the folder names of the libraries with spaces replaced by underscores. A full description of the various URL based actions is in the User Manual.

Wann immer Sie Argumente mit Leerzeichen an **calibre** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--detach

Calibre vom Terminalfenster ablösen, falls vorhanden (nur für Linux)

--help, -h

Diesen Hilfetext anzeigen und beenden.

--ignore-plugins

Benutzerdefinierte Erweiterungen ignorieren. Nützlich, wenn Sie eine Erweiterung installiert haben, die Calibre am Starten hindert

--no-update-check

Nicht nach neuen Versionen suchen

--shutdown-running-calibre, -s

Eine laufende Calibre-Instanz - falls vorhanden - beenden. Beachten Sie, dass dabei ggf. laufende Aufträge ohne Rückfrage abgebrochen werden. Also bitte mit Vorsicht verwenden.

--start-in-tray

Minimiert im Infobereich starten.

--verbose, -v

Ignoriert, bitte nicht verwenden. Nur wegen Altsystemen vorhanden

--version

Programmversion anzeigen und beenden

--with-library

Bibliothek unter dem angegebenen Pfad verwenden.

13.1.2 calibre-customize

```
calibre-customize Optionen
```

Calibre anpassen durch Laden externer Erweiterungen.

Wann immer Sie Argumente mit Leerzeichen an **calibre-customize** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--add-plugin, -a

Erweiterung hinzufügen durch Pfadangabe zu jener ZIP-Datei, die die Erweiterung enthält.

--build-plugin, -b

Für PlugIn-Entwickler: Pfad zum Ordner in dem das PlugIn entwickelt wird. Dieser Befehl wird das PlugIn automatisch packen und in Calibre aktualisieren.

--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

Genannte Erweiterung deaktivieren

--enable-plugin

Genannte Erweiterung aktivieren

--help, -h

Diesen Hilfetext anzeigen und beenden.

--list-plugins, -l

Alle installierten Erweiterungen auflisten

--remove-plugin, -r

Benutzererweiterung nach Namen entfernen. Dies hat keinen Einfluss auf integrierte Erweiterungen.

--version

Programmversion anzeigen und beenden

13.1.3 calibre-debug

```
calibre-debug [Optionen]
```

Various command line interfaces useful for debugging calibre. With no options, this command starts an embedded Python interpreter. You can also run the main calibre GUI, the calibre E-book viewer and the calibre editor in debug mode.

It also contains interfaces to various bits of calibre that do not have dedicated command line tools, such as font subsetting, the E-book diff tool and so on.

You can also use **calibre-debug** to run standalone scripts. To do that use it like this:

> **calibre-debug** -e myscript.py -- --option1 --option2 file1 file2 …

Everything after the -- is passed to the script. You can also use **calibre-debug** as a shebang in scripts, like this:

> #!/usr/bin/env -S **calibre-debug** -e -- –

Wann immer Sie Argumente mit Leerzeichen an **calibre-debug** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--add-simple-plugin

Eine einfache Erweiterung (d. h. eine nur aus einer *.py-Datei bestehende Erweiterung) hinzufügen durch Angabe des Pfads zur *.py-Datei, die den Erweiterungscode enthält.

--command, -c

Python-Code ausführen.

--debug-device-driver, -d

Fehlerdiagnose bei Geräteerkennung

--default-programs

Calibre in Windows als Standardprogramm festlegen bzw. entfernen. *--default-programs* (die Titelseite 305) = (register|unregister)

--diff

Calibre-Dateivergleichswerkzeug ausführen. Beispiel: calibre-debug *--diff* (die Titelseite 305) Datei1 Datei2

--edit-book

Bucheditor im Fehlerdiagnosemodus starten.

--exec-file, -e

Python-Code in der Datei ausführen.

--explode-book, -x

Explode the book into the specified folder. Usage: -x file.epub output_dir Exports the book as a collection of HTML files and metadata, which you can edit using standard HTML editing tools. Works with EPUB, AZW3, HTMLZ and DOCX files.

--export-all-calibre-data

Export all calibre data (books/settings/plugins). Normally, you will be asked for the export folder and the libraries to export. You can also specify them as command line arguments to skip the questions. Use absolute paths for the export folder and libraries. The special keyword "all" can be used to export all libraries. Examples: calibre-debug *--export-all-calibre-data* (die Titelseite 306) # for interactive use calibre-debug *--export-all-calibre-data* (die Titelseite 306) /path/to/empty/export/folder /path/to/library/folder1 /path/to/library2 calibre-debug *--export-all-calibre-data* (die Titelseite 306) /export/folder all # export all known libraries

--gui, -g

Ausführen der grafischen Benutzeroberfläche im Fehlerdiagnosemodus. Die Diagnoseausgabe wird an Stdout und Stderr ausgegeben.

--gui-debug

Grafische Benutzeroberfläche mit einer Fehlerdiagnosekonsole ausführen und im angegebenen Verzeichnis protokollieren. Nur zur internen Verwendung. Verwenden Sie die Option "-g", um die Benutzeroberfläche im Fehlerdiagnosemodus auszuführen

--help, -h

Diesen Hilfetext anzeigen und beenden.

--implode-book, -i

Packen Sie ein zuvor entpacktes Buch. Verwendung: -i output_dir file.epub Importiert das Buch aus den Dateien in output_dir, die durch einen vorherigen Aufruf von *--explode-book* (die Titelseite 306) erstellt wurden. Achten Sie darauf, dass Sie den gleichen Dateityp angeben, der beim Entpacken verwendet wurde.

--import-calibre-data

Zuvor exportierte Calibre-Daten importieren

--inspect-mobi, -m

MOBI-Datei(en) im/in den angegebenen Verzeichnis(sen) untersuchen

--kepubify

Convert the specified EPUB file to KEPUB without doing a full conversion. This is what the Kobo driver does when sending files to the device.

--paths

Ausgabe der für die Einrichtung von Calibres Umgebungsvariablen notwendigen Verzeichnispfade

--run-plugin, -r

Run a plugin that provides a command line interface. For example: calibre-debug -r "Plugin name" -- file1 --option1 Everything after the -- will be passed to the plugin as arguments.

--run-test, -t

Run the named test(s). Use the special value "all" to run all tests. If the test name starts with a period it is assumed to be a module name. If the test name starts with @ it is assumed to be a category name.

--run-without-debug

Don't run with the DEBUG flag set

--shutdown-running-calibre, -s

Eine laufende Calibre-Instanz - falls vorhanden - beenden. Beachten Sie, dass dabei ggf. laufende Aufträge ohne Rückfrage abgebrochen werden. Also bitte mit Vorsicht verwenden.

--subset-font, -f

Angegebene eingebettete Schriftart reduzieren. Verwenden Sie -- nach dieser Option, um Optionen an das Programm für die Schriftartreduzierung zu übergeben.

--test-build

Binäre Module des Builds überprüfen

--un-kepubify

Convert the specified KEPUB file to EPUB without doing a full conversion. This is what the Kobo driver does when importing files from the device.

--version

Programmversion anzeigen und beenden

--viewer, -w

eBook-Betrachter im Fehlerdiagnosemodus ausführen

13.1.4 calibre-server

```
calibre-server [Optionen] [Pfad zum Bibliotheksordner …]
```

Starten des Calibre-Inhalteservers. Der Calibre-Inhalteserver stellt Ihre Calibre-Bibliotheken über das Internet bereit. Sie können den Pfad zu den Bibliotheksordnern als Argumente von **calibre-server** angeben. Falls Sie keinen Pfad angeben, werden alle dem Calibre-Hauptprogramm bekannten Bibliotheken verwendet.

Wann immer Sie Argumente mit Leerzeichen an **calibre-server** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--access-log

Pfad zur Zugangsprotokolldatei. Dieses Protokoll enthält Informationen über Clients, die sich mit dem Server verbinden und Anfragen stellen. Standardmäßig wird keine Protokollierung durchgeführt.

--ajax-timeout

Zeit (in Sekunden), um auf eine Antwort des Servers zu warten, wenn Anfragen gestellt werden.

--auth-mode

Wählen Sie die verwendete Authentifizierungsart aus. Festlegen des vom Server verwendeten HTTP-Authentifizierungsmodus. Setzen Sie diesen auf "basic", falls Sie diesen Server hinter einem SSL-Proxy verwenden. Belassen Sie ihn ansonsten auf "Auto", was "Basic" verwendet, falls SSL konfiguriert ist, andernfalls wird "digest" verwendet.

--auto-reload

Den Server automatisch neu laden, wenn sich der Quellcode ändert. Nützlich für Entwickler. Sie sollten außerdem einen kleinen Wert für die Zeitüberschreitung zum Herunterfahren angeben.

--ban-after

Anzahl der fehlgeschlagenen Anmeldungen für einen Bann. Die Anzahl der Login-Fehler, nach denen eine IP-Adresse gesperrt wird.

--ban-for

Sperren von IP-Adressen mit wiederholten Login-Fehlern. Verbietet vorübergehend den Zugriff für IP-Adressen, die wiederholt fehlgeschlagene Anmeldungen für die angegebene Anzahl von Minuten haben. Nützlich, um Versuche zu verhindern, Passwörter zu erraten. Wenn auf Null gesetzt, wird kein Banning durchgeführt.

--book-list-mode

Standardmodus für Buchlisten. Festlegen des Standardmodus für Buchlisten, der für neue Benutzer verwendet wird. Einzelne Benutzer können den Standard in ihren eigenen Einstellungen übergehen. Standard ist die Verwendung eines Titelbildrasters.

--compress-min-size

Min. Größe, bei der Antworten Datenkompression verwenden (in Bytes).

--custom-list-template

Pfad zu einer JSON-Datei, die eine Vorlage für den benutzerdefinierten Buchlistenmodus enthält. Der einfachste Weg zur Erstellung einer solchen Vorlagendatei ist, in Calibre zu Einstellungen → Netzwerkserver → Buchlistenvorlage zu navigieren, die Vorlage zu erstellen und diese dann zu exportieren.

--daemonize

Prozess im Hintergrund als Dienst ausführen (nur Linux).

--displayed-fields

Angezeigte benutzerdefinierte Felder beschränken. Kommagetrennte Liste von benutzerdefinierten Metadatenfeldern, die vom Inhalteserver in den Ansichten "/opds" und "/mobile" angezeigt werden sollen. Falls Sie diese Option verwenden, werden sämtliche in der Liste nicht enthaltenen Felder nicht angezeigt. Zum Beispiel: my_rating,my_tags

--enable-allow-socket-preallocation, --disable-allow-socket-preallocation

Socket-Vorzuweisung, z. B. mit systemd-Socket-Aktivierung. Standardmäßig ist diese Optione aktiviert.

--enable-auth, --disable-auth

Passwortbasierte Authentifizierung für Serverzugriff. Normalerweise ist der Server nicht eingeschränkt, sodass jeder darauf Zugriff hat. Mit dier Option können Sie den Zugriff auf vordefinierte Benutzer beschränken. Standardmäßig ist diese Option deaktiviert.

--enable-fallback-to-detected-interface, --disable-fallback-to-detected-interface

Rückgriff auf autom. erkannte Schnittstelle. Falls es dem Server aus irgendeinem Grund nicht möglich ist, die in der Option "listen_on" angegebene Schnittstelle einzubinden, wird versucht, eine Schnittstelle zu erkennen, die mit der Außenwelt verbindet, und dann diese einzubinden. Standardmäßig ist diese Optione aktiviert.

--enable-local-write, --disable-local-write

Erlauben Sie nicht authentifizierten lokalen Verbindungen, Änderungen vorzunehmen. Wenn Sie die Authentifizierung nicht einschalten, arbeitet der Server normalerweise im Nur-Lese-Modus, damit anonyme Benutzer keine Änderungen an Ihren Calibre-Bibliotheken vornehmen können. Mit dieser Option kann jeder, der sich von demselben Computer aus verbindet, auf dem der Server läuft, Änderungen vornehmen. Dies ist nützlich, wenn Sie den Server ohne Authentifizierung betreiben möchten, aber dennoch calibredb verwenden möchten, um Änderungen an Ihren Calibre-Bibliotheken vorzunehmen. Beachten Sie, dass die Aktivierung dieser Option bedeutet, dass jedes Programm, das auf dem Computer läuft, Änderungen an Ihren Calibre-Bibliotheken vornehmen kann. Standardmäßig ist diese Option deaktiviert.

--enable-log-not-found, --disable-log-not-found

HTTP 404 (nicht gefunden)-Anfragen protokollieren. Normalerweise protokolliert der Server alle HTTP-Anfragen nach nicht gefundenen Ressourcen. Dies kann zu einer nicht unerheblichen Menge von Protokollmüll führen, falls Ihr Server das Ziel von Bots ist. Verwenden Sie diese Option, um die Protokollierung auszuschalten. Standardmäßig ist diese Optione aktiviert.

--enable-use-bonjour, --disable-use-bonjour

OPDS-Feeds via BonJour veröffentlichen. OPDS-Feeds via BonJour-Dienst veröffentlichen, sodass OPDS-basierte Lese-Apps den Server automatisch erkennen und sich mit diesem verbinden können. Standardmäßig ist diese Optione aktiviert.

--enable-use-sendfile, --disable-use-sendfile

Zero-Copy-Dateiübertragungen für verbesserte Leistung. Dies verwendet beim Senden von Dateien über das Netzwerk sogenannte Zero-Copy-, In-Kernel-Übertragungen, was die Leistung verbessert. Dies kann jedoch beschädigte Dateiübertragungen auf einigen fehlerhaften Dateisystemen verursachen. Falls Sie beschädigte Dateiübertragungen feststellen, schalten Sie diese Funktion aus. Standardmäßig ist diese Optione aktiviert.

--help, -h

Diesen Hilfetext anzeigen und beenden.

--ignored-fields

Benutzerdefinierte Metadatenfelder ignoriert. Kommagetrennte Liste von benutzerdefinierten Metadatenfeldern, die vom Inhalteserver in den Ansichten "/opds" und "/mobile" nicht angezeigt werden sollen. Zum Beispiel: my_rating,my_tags

--listen-on

Schnittstelle, die auf eingehende Verbindungen warten soll. The default is to listen on all available IPv6 and IPv4 interfaces. You can change this to, for example, "127.0.0.1" to only listen for IPv4 connections from the local machine, or to "0.0.0.0" to listen to all incoming IPv4 connections.

--log

Pfad zur Protokolldatei für das Serverprotokoll. Dieses Protokoll beinhaltet Serverinformationen und -fehler, keine Zugriffsprotokolle. Standardmäßig wird die Protokolldatei in stdout geschrieben.

--manage-users

Manage the database of users allowed to connect to this server. You can use it in automated mode by adding a –. See calibre-server *--manage-users* (die Titelseite 309) -- help for details. See also the *--userdb* (die Titelseite 310) option.

--max-header-line-size

Max. Größe eines einzelnen HTTP-Headers (in KB).

--max-job-time

Maximalzeit für Arbeiterprozesse. Maximale Zeit, die Arbeiterprozesse ausgeführt werden dürfen (in Minuten). Zum Deaktivieren der Zeitbeschränkung auf Null setzen.

--max-jobs

Maximale Anzahl von Arbeiterprozessen. Arbeiterprozesse werden nach Bedarf gestartet und für umfangreiche Aufgaben verwendet, wie das Vorbereiten eine Buches für die Anzeige, Hinzufügen von Büchern, Konvertieren usw. Normalerweise basiert die maximale Anzahl solcher Prozesse auf der Anzahl von CPU-Kernen. Sie können sie mit dieser Einstellung steuern.

--max-log-size

Max. Protokolldateigröße (in MB). Maximalgröße von vom Server erzeugten Protokolldateien. Sobald das Protokoll größer als die angegebene Größe wird, wird dieses automatisch rotiert. Zum Deaktivieren der Protokollrotation auf Null setzen.

--max-opds-items

Maximale Anzahl von Büchern in OPDS-Feeds. Maximale Anzahl von Büchern, die der Server in einem einzelnen OPDS-Bezugs-Feed zurückgibt.

--max-opds-ungrouped-items

Max. Anzahl nicht gruppierter Elemente in OPDS-Feeds. Kategorieelemente wie Autor/Schlagwörter nach dem ersten Buchstaben gruppieren, falls mehr als die angegebene Anzahl von Elementen existiert. Zum Deaktivieren auf Null setzen.

--max-request-body-size

Max. zulässige Größe für auf den Server übertragene Dateien (in MB).

--num-per-page

Anzahl der auf einer einzelnen Seite anzuzeigenden Bücher. Anzahl der auf einer einzelnen Seite im Browser anzuzeigenden Bücher.

--pidfile

Prozess-PID in die angegebene Datei schreiben

--port

Port, auf dem auf Verbindungen gewartet werden soll.

--search-the-net-urls

Path to a JSON file containing URLs for the "Search the internet" feature. The easiest way to create such a file is to go to Preferences-> Sharing over the net->Search the internet in calibre, create the URLs and export them.

--shutdown-timeout

Wartezeit in Sekunden bis zu einem sauberen Herunterfahren.

--ssl-certfile

Pfad zur SSL-Zertifikatsdatei.

--ssl-keyfile

Pfad zur SSL-Privatschlüsseldatei.

--timeout

Zeit (in Sekunden), nach der eine untätige Verbindung geschlossen wird.

--trusted-ips

Erlauben Sie nicht authentifizierten lokalen Verbindungen, Änderungen vorzunehmen. Normally, if you do not turn on authentication, the server operates in read-only mode, so as to not allow anonymous users to make changes to your calibre libraries. This option allows anybody connecting from the specified IP addresses to make changes. Must be a comma separated list of address or network specifications. This is useful if you want to run the server without authentication but still use calibredb to make changes to your calibre libraries. Note that turning on this option means anyone connecting from the specified IP addresses can make changes to your calibre libraries.

--url-prefix

Vorzustellendes Präfix für alle URLs. Hilfreich, wenn Sie diesen Server hinter einem Reverse-Proxy betreiben möchten. Verwenden Sie beispielsweise "/calibre" als das URL-Präfix.

--userdb

Pfad zur Benutzerdatenbank, die für die Authentifizierung verwendet werden soll. Die Datenbank ist eine SQLite-Datei. Um es zu erstellen, benutze *--manage-users* (die Titelseite 309). Mehr über die Verwaltung von Benutzern erfahren Sie unter: https://manual.calibre-ebook.com/de/server.html#managing-user-accounts-from-the-command-line-only

--version

Programmversion anzeigen und beenden

--worker-count

Anzahl von Arbeiterprozessen für die Verarbeitung von Anfragen.

13.1.5 calibre-smtp

```
calibre-smtp [Optionen] [Absender Empfänger Text]
```

Senden von E-Mail-Nachrichten über das SMTP-Protokoll. **calibre-smtp** kennt zwei Betriebsmodi. Im „Verfassen“-Modus geben Sie Absender, Empfänger und Text an, und diese Angaben werden dazu verwendet, eine E-Mail-Nachricht zu erstellen und zu versenden. Im „Filter“-Modus liest **calibre-smtp** eine vollständige E-Mail-Nachricht über die Standardeingabe STDIN ein und versendet diese.

„Text“ ist der Textkörper der E-Mail-Nachricht. Falls Text nicht angegeben ist, wird eine vollständige E-Mail-Nachricht von der Standardeingabe STDIN eingelesen. „Absender“ ist die E-Mail-Adresse des Absenders und „Empfänger“ ist die E-Mail-Adresse des Empfängers. Wurde eine vollständige E-Mail von STDIN eingelesen, dienen Absender und Empfänger nur zum Aushandeln des SMTP-Protokolls, die Nachrichtenkopfzeilen werden nicht verändert.

Wann immer Sie Argumente mit Leerzeichen an **calibre-smtp** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--fork, -f

Nachrichten im Hintergrund versenden. Falls Sie diese Option verwenden, sollten Sie zur Behandlung von Übermittlungsfehlern auch *--outbox* (die Titelseite 311) verwenden.

--help, -h

Diesen Hilfetext anzeigen und beenden.

--localhost, -l

Computername des localhost. Dieser wird bei der Verbindung zum SMTP-Server verwendet.

--outbox, -o

Pfad zum maildir-Ordner zum Speichern fehlgeschlagener E-Mails.

--timeout, -t

Zeitüberschreitung für die Verbindung

--verbose, -v

Mehr Informationen anzeigen

--version

Programmversion anzeigen und beenden

COMPOSE MAIL

Optionen zum Verfassen einer E-Mail. Wird ignoriert, wenn kein Text angegeben wird

--attachment, -a

An die E-Mail anzuhängende Datei

--subject, -s

Betreff der E-Mail

SMTP RELAY

Optionen für die Verwendung eines SMTP-Servers zum Versenden von E-Mails. Solange über –relay kein SMTP-Server angegeben ist, wird Calibre versuchen, eine E-Mail direkt zu versenden.

--cafile

Path to a file of concatenated CA certificates in PEM format, used to verify the server certificate when using TLS. By default, the system CA certificates are used.

--dont-verify-server-certificate

Do not verify the server certificate when connecting using TLS. This used to be the default behavior in calibre versions before 3.27. If you are using a relay with a self-signed or otherwise invalid certificate, you can use this option to restore the pre 3.27 behavior

--encryption-method, -e

Verwendete Verschlüsselungsmethode beim Herstellen der Verbindung mit dem SMTP-Server. Auswahlmöglichkeiten sind TLS, SSL und NONE. Standardwert ist TLS. Warnung: Die Auswahl NONE ist sehr unsicher.

--password, -p

Passwort für den Mail-Server

--port

Der Anschluss zum SMTP-Server, der auf die Verbindungen wartet. Standardmäßig wird bei SSL-Verschlüsslung Port 465 und andernfalls 25 verwendet.

--relay, -r

Der SMTP-Server-Name zum Versenden einer E-Mail Nachricht.

--username, -u

Benutzername auf dem Mail-Server

13.1.6 calibredb

```
calibredb Befehl [Optionen] [Argumente]
```

calibredb ist die Befehlszeilenschnittstelle zur Calibre-Datenbank. Es hat mehrere Unterbefehle, die nachfolgend dokumentiert sind.

calibredb kann verwendet werden, um entweder eine Calibre-Datenbank anzugeben, die durch den Pfad definiert ist, oder einen Calibre-*Content Server* - der entweder auf dem lokalen Rechner oder über das Internet läuft - zu manipulieren. Sie können einen Calibre- *Content Server* starten, indem Sie entweder **calibre-server** verwenden oder im Hauptprogramm von Calibre auf *Connect/share → Start Content Server* klicken. Da **calibredb** Änderungen an Ihren Calibre-Bibliotheken vornehmen kann, müssen Sie zuerst die Authentifizierung auf dem Server einrichten. Es gibt zwei Möglichkeiten, dies zu tun:

- Wenn Sie planen, nur eine Verbindung zu einem Server herzustellen, der auf demselben Computer läuft,

können Sie einfach die Option `--enable-local-write` des Content Servers verwenden, um jedem Programm, einschließlich calibredb, das auf dem lokalen Computer läuft, Änderungen an Ihren Calibre-Daten zu ermöglichen. Wenn Sie den Server im Hauptprogramm von Calibre ausführen, finden Sie diese Option unter *Einstellungen → Über das Netzwerk freigeben → Erweitert*.

- Wenn Sie den Zugriff über das Internet ermöglichen möchten, sollten Sie

Benutzerkonten auf dem Server einrichten und *--username* (die Titelseite 314) und *--password* (die Titelseite 314) von **calibredb** verwenden, um ihm den Zugriff zu ermöglichen. Sie können die Benutzerauthentifizierung für **calibre-server** aktivieren, indem Sie die Option `--enable-auth` verwenden und `--manage-users` verwenden, um die Benutzerkonten zu erstellen. Wenn Sie den Server im Hauptprogramm von Calibre ausführen, verwenden Sie *Einstellungen → Über das Netzwerk freigeben → Benutzername/Passwort erforderlich*.

Um eine Verbindung zu einem laufenden Content Server herzustellen, geben Sie die URL des Servers an die Option *--with-library* (die Titelseite 313) weiter. Weitere Informationen und Beispiele finden Sie in der Dokumentation dieser Option.

- *Globale Optionen* (die Titelseite 313)
- *list* (die Titelseite 314)
- *add* (die Titelseite 315)
  - *HINZUFÜGEN VON ORDNERN* (die Titelseite 316)
- *remove* (die Titelseite 316)
- *add_format* (die Titelseite 316)
- *remove_format* (die Titelseite 317)
- *show_metadata* (die Titelseite 317)
- *set_metadata* (die Titelseite 317)
- *export* (die Titelseite 318)
- *catalog* (die Titelseite 319)
  - *EPUB OPTIONEN* (die Titelseite 319)
- *saved_searches* (die Titelseite 321)
- *add_custom_column* (die Titelseite 321)
- *custom_columns* (die Titelseite 321)
- *remove_custom_column* (die Titelseite 322)
- *set_custom* (die Titelseite 322)
- *restore_database* (die Titelseite 322)
- *check_library* (die Titelseite 322)
- *list_categories* (die Titelseite 323)
- *backup_metadata* (die Titelseite 323)
- *clone* (die Titelseite 324)
- *embed_metadata* (die Titelseite 324)
- *search* (die Titelseite 324)
- *fts_index* (die Titelseite 325)
- *fts_search* (die Titelseite 325)

Globale Optionen

`--help, -h`

Diesen Hilfetext anzeigen und beenden.

`--library-path, --with-library`

Pfad zur Calibre-Bibliothek. Standardmäßig wird der in den Einstellungen gespeicherte Pfad verwendet. Sie können auch mit einem Calibre-Inhalteserver verbinden, um Aktionen auf Fernbibliotheken anzuwenden. Verwenden Sie hierfür eine URL der Form: http://hostname:port/#bibliothek_id, zum Beispiel http://localhost:8080/#meinebibliothek. bibliothek_id ist die Bibliothekskennung jener Bibliothek, zu der Sie auf dem Inhalteserver verbinden wollen. Um eine Liste aller auf dem Server verfügbaren Bibliothekskennungen zu erhalten, können Sie den Sonderwert `"-"` verwenden. Für Details zum Einrichten des Zugriffs mittels Inhalteserver siehe

https://manual.calibre-ebook.com/de/generated/de/calibredb.html.

--password

Passwort für die Verbindung zu einem Calibre-Inhalteserver. Um das Passwort von der Standardeingabe zu lesen, verwenden Sie den Sonderwert <stdin>. Um das Passwort aus einer Datei zu lesen, verwenden Sie <f:/path/to/file> (d. h. "<f:", gefolgt vom vollständigen Dateipfad und einem abschließenden ">"). Die spitzen Klammern sind erforderlich. Denken Sie daran, diese zu "escapen" oder verwenden Sie Anführungszeichen für Ihre Shell.

--timeout

Das Zeitlimit in Sekunden, wenn eine Verbindung zu einer Calibre Biblithek über das Netzwerk hergestellt wird. Der Standardwert beträgt zwei Minuten

--username

Benutzername zum Verbinden mit einem Calibre-Inhalteserver

--version

Programmversion anzeigen und beenden

list

```
calibredb list [Optionen]
```

Auflisten der in der Calibre-Datenbank verfügbaren Bücher.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--ascending

Ergebnisse in aufsteigender Reihenfolge sortieren

--fields, -f

Anzuzeigende Felder bei der Auflistung von Büchern in der Datenbank. Sollte eine kommagetrennte Liste von Feldern sein. Verfügbare Felder: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid Standard: % default. Das Sonderfeld "all" kann für die Auswahl aller Felder verwendet werden. Zusätzlich zu den oben genannten integrierten Feldern sind auch benutzerdefinierte Felder als "*Feldname" verfügbar. Verwenden Sie beispielsweise für ein benutzerdefiniertes Feld "#Bewertung" den Namen "*Bewertung".

--for-machine

Erzeugt die Ausgabe im JSON-Format, welches maschinell besser zu analysieren ist. Bewirkt, dass die Einstellungen für Zeilenbreite und Trennzeichen-Optionen ignoriert werden.

--limit

Maximale Anzahl anzuzeigender Ergebnisse. Standard: alle

--line-width, -w

Maximale Breite einer einzelnen Zeile in der Ausgabe. In der Voreinstellung wird die Bildschirmgröße erkannt.

--prefix

Der Präfix für alle Dateipfade. Voreinstellung ist der absolute Pfad zum Bibliotheksordner.

--search, -s

Die Ergebnisse durch die Suchanfrage filtern. Für das Format der Suchanfrage sehen Sie sich bitte die Dokumentation, die die Suche betrifft, im Benutzerhandbuch an. Voreinstellung ist, keine Filterung durchzuführen.

--separator

Zeichenfolge zur Unterscheidung der Felder. Die Voreinstellung ist ein Leerzeichen.

--sort-by

The field by which to sort the results. You can specify multiple fields by separating them with commas. Available fields: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid Default: id

--template

The template to run if "template" is in the field list. Note that templates are ignored while connecting to a calibre server. Default: None

--template_file, -t

Path to a file containing the template to run if "template" is in the field list. Default: None

--template_heading

Heading for the template column. Default: template. This option is ignored if the option *--for-machine* (die Titelseite 314) is set

add

```
calibredb add [Optionen] Datei1 Datei2 Datei3 …
```

Hinzufügen der angegebenen Dateien als Bücher zur Datenbank. Sie können auch Verzeichnisse angeben - vergleichen Sie hierfür die auf Verzeichnisse bezogenen Optionen unten.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--authors, -a

Autor der hinzugefügten Bücher festlegen

--automerge, -m

Wenn Bücher mit ähnlichen Titeln und Verfassern gefunden werden führe die neuen Formate (Dateien) automatisch mit den vorhandenen Bucheinträgen zusammen. Der Wert "Ignorieren" bedeutet, dass doppelte Formate verworfen werden. Der Wert "Überschreiben" bedeutet doppelte Formate in der Bibliothek werden von neu hinzugefügten Dateien überschrieben. Der Wert "Neuer Eintrag" bedeutet, dass doppelte Formate in einem neuen Bucheintrag gespeichert werden.

--cover, -c

Pfad zum Titelbild des hinzugefügten Buches

--duplicates, -d

Füge Bücher zur Datenbank selbst dann hinzu, wenn sie bereits vorhanden sind. Der Vergleich wird basierend auf Buchtiteln und Verfassern durchgeführt. Beachte, dass die `--automerge-Option` Vorrang hat.

--empty, -e

Leeres Buch hinzufügen (ein Buch ohne Formate)

--identifier, -I

Kennungen für dieses Buch festlegen, z. B. -I asin:XXX -I isbn:YYY

--isbn, -i

ISBN der hinzugefügten Bücher festlegen

--languages, -l

Eine kommagetrennte Liste der Sprachen (obwohl einige Sprachen auch am Namen erkannt werden, verwendet man besser die Sprachcodes der ISO639)

--series, -s

Serien der der hinzugefügten Bücher festlegen

--series-index, -S

Serien-Nummer der hinzugefügten Bücher festlegen

--tags, -T

Schlagwörter des/der hinzugefügten Buches/Bücher festlegen

--title, -t

Titel der hinzugefügten Bücher festlegen

HINZUFÜGEN VON ORDNERN

Optionen, um das Hinzufügen von Büchern aus Verzeichnissen zu steuern. Standardmäßig werden nur Dateien mit Dateinamenerweiterungen bekannter eBook-Dateitypen hinzugefügt.

--add

A filename (glob) pattern, files matching this pattern will be added when scanning folders for files, even if they are not of a known e-book file type. Can be specified multiple times for multiple patterns.

--ignore

A filename (glob) pattern, files matching this pattern will be ignored when scanning folders for files. Can be specified multiple times for multiple patterns. For example: *.pdf will ignore all PDF files

--one-book-per-directory, -1

Assume that each folder has only a single logical book and that all files in it are different e-book formats of that book

--recurse, -r

Verzeichnisse rekursiv verarbeiten

remove

```
calibredb remove IDs
```

Entfernen der durch IDs bezeichneten Bücher aus der Datenbank. Die IDs müssen in Form einer kommagetrennten Liste von ID-Nummern angegeben werden (Sie können ID-Nummern mithilfe des Suchbefehls erhalten). Beispielsweise 23,34,57-85 (bei Angabe eines Bereichs ist die letzte Nummer im Bereich nicht enthalten).

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--permanent

Papierkorb nicht verwenden

add_format

```
calibredb add_format [Optionen] ID eBook_Datei
```

Hinzufügen des eBooks in eBook_Datei zu den verfügbaren Formaten für das durch die ID bestimmte logische Buch. Sie können die ID durch die Verwendung des Suchbefehls erhalten. Falls das Format bereits existiert, wird es ersetzt, es sei denn, die „Nicht ersetzen“-Option ist ausgewählt.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--as-extra-data-file

Füge die Datei als zusätzliche Datendatei zum Buch hinzu, kein E-Book-Format

--dont-replace

Ersetzen Sie das Format nicht, falls es schon existiert

remove_format

```
calibredb remove_format [Optionen] ID Fmt
```

Entfernen des Formats Fmt des durch die ID bezeichneten logischen Buches. Sie erhalten die ID durch Verwendung des Suchbefehls. fmt sollte eine Dateinamenerweiterung wie LRF, TXT oder EPUB sein. Falls das logische Buch nicht im angegebenen Format verfügbar ist, passiert gar nichts.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

show_metadata

```
calibredb show_metadata [Optionen] ID
```

Anzeigen der in der Calibre-Datenbank gespeicherten Metadaten für das durch die ID bezeichnete Buch. ID ist eine ID-Nummer aus dem Suchbefehl.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--as-opf

Metadaten in OPF-Form (XML) drucken

set_metadata

```
calibredb set_metadata [options] book_id [/path/to/metadata.opf]
```

Set the metadata stored in the calibre database for the book identified by book_id from the OPF file metadata.opf. book_id is a book id number from the search command. You can get a quick feel for the OPF format by using the –as-opf switch to the show_metadata command. You can also set the metadata of individual fields with the –field option. If you use the –field option, there is no need to specify an OPF file.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--field, -f

Bereich festlegen. Das Format ist field_name:value, zum Beispiel: *--field* (die Titelseite 317) tags:tag1,tag2. Verwenden Sie *--list-fields* (die Titelseite 317) um eine Liste mit allen Feldnamen zu erhalten. Sie können diese Funktion mehrfach anwenden, um mehrere Felder festzulegen. Hinweis: Für die Sprache muss der ISO639 Sprachcode (z. B. en für Englisch, de für Deutsch, usw.) verwendet werden. Für Kennungen ist die Syntax *--field* (die Titelseite 317) identifiers:isbn:XXXX,doi:YYYYY. Für boolesche Felder (ja/nein) verwenden Sie true und false oder yes und no.

--list-fields, -l

Listet die Metadaten-Feldnamen auf, die mit der *--field* (die Titelseite 317) option (Feldoption) verwendet werden können

export

```
calibredb export [options] ids
```

Export the books specified by ids (a comma separated list) to the filesystem. The **export** operation saves all formats of the book, its cover and metadata (in an OPF file). Any extra data files associated with the book are also saved. You can get id numbers from the search command.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--all
: Alle Bücher der Datenbank exportieren, die Liste der IDs wird ignoriert.

--dont-asciiize
: Have calibre convert all non English characters into English equivalents for the file names. This is useful if saving to a legacy filesystem without full support for Unicode filenames. Die Betätigung dieses Schalters stellt das Verhalten aus.

--dont-save-cover
: Normalerweise speichert Calibre das Titelbild in einer separaten Datei zusammen mit den eigentlichen eBook-Dateien. Die Betätigung dieses Schalters stellt das Verhalten aus.

--dont-save-extra-files
: Save any data files associated with the book when saving the book Die Betätigung dieses Schalters stellt das Verhalten aus.

--dont-update-metadata
: Normalerweise aktualisiert Calibre die Metadaten in den gespeicherten Dateien mit den Metadaten in der Calibre-Bibliothek. Dies verlangsamt das Speichern auf Datenträger. Die Betätigung dieses Schalters stellt das Verhalten aus.

--dont-write-opf
: Normalerweise schreibt Calibre die Metadaten in eine separate OPF-Datei zusammen mit den eigentlichen eBook-Dateien. Die Betätigung dieses Schalters stellt das Verhalten aus.

--formats
: Kommagetrennte Liste der für jedes Buch zu speichernden Formate. Standardmäßig werden alle verfügbaren Formate gespeichert.

--progress
: Fortschritt melden

--replace-whitespace
: Leerzeichen mit Unterstrichen ersetzen.

--single-dir
: Alle Bücher in einen einzigen Ordner exportieren

--template
: The template to control the filename and folder structure of the saved files. Default is "{author_sort}/{title}/{title} - {authors}" which will save books into a per-author subfolder with filenames containing title and author. Available controls are: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt
: Das Format, in dem Kalenderdaten angezeigt werden sollen. %d - Tag, %b - Monat, %m - Monatsnummer, %Y - Jahr. Standard ist: %b, %Y

--to-dir

Bücher in den angegebenen Ordner exportieren. Vorgabe ist .

--to-lowercase

Pfade in Kleinbuchstaben umwandeln.

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.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--ids, -i

Kommagetrennte Liste der zu katalogisierenden Datenbank-IDs. Wenn angegeben, wird *--search* (die Titelseite 319) ignoriert. Standardeinstellung: alle

--search, -s

Die Ergebnisse durch die Suchanfrage filtern. Für das Format der Suchanfrage sehen Sie sich bitte die Dokumentation, die die Suche betrifft, im Benutzerhandbuch an. Standardeinstellung: keine Filterung

--verbose, -v

Detaillierte Ausgabeinformation anzeigen. Hilfreich zur Fehlerdiagnose

EPUB OPTIONEN

--catalog-title

Titel des erzeugten Katalogs, wie er als Titel in den Metadaten verwendet wird. Standard: "My Books" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--cross-reference-authors

Querverweise im Abschnitt "Autoren" für Bücher mit unterschiedlichen Autoren erstellen. Standard: "False" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--debug-pipeline

Save the output from different stages of the conversion pipeline to the specified folder. Useful if you are unsure at which stage of the conversion process a bug is occurring. Default: 'None' Applies to: AZW3, EPUB, MOBI output formats

--exclude-genre

Regulärer Ausdruck, der die Schlagwörter beschreibt, die von den Genres ausgeschlossen werden sollen. Standard: "[.+]|^+$" schließt mit "[]" geklammerte Schlagwörter aus, z. B. "[Projekt Gutenberg]", und "+", das Standardschlagwort für gelesene Bücher. Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--exclusion-rules

Geben Sie die Regeln an, die verwendet werden sollen, um Bücher aus dem generierten Katalog auszuschließen. Das Modell für eine Ausschlussregel ist entweder ('<Regelname>','Tags','<kommagetrennte Tag-Liste>') oder ('<Regelname>','<Benutzerdefinierte Spalte>','<Muster>'). Zum Beispiel: (('Archivierte Bücher','#status','Archiviert'),) wird ein Buch mit dem Wert 'Archived' in der benutzerdefinierten Spalte 'status' ausschließen. Wenn mehrere Regeln festgelegt sind, werden alle angewandt. Standard: "(('Catalogs','Tags','Catalog'),)" Anwendbar auf AZW3, ePub, MOBI-Zielformate

--generate-authors

Abschnitt "Autoren" im Katalog einschließen. Standard: "False" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--generate-descriptions

Abschnitt "Beschreibungen" im Katalog einschließen. Standard: "False" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--generate-genres

Abschnitt "Genres" im Katalog einschließen. Standard: "False" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--generate-recently-added

Abschnitt "Kürzlich hinzugefügt" im Katalog einschließen. Standard: "False" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--generate-series

Abschnitt "Serien" im Katalog einschließen. Standard: "False" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--generate-titles

Abschnitt "Titel" im Katalog einschließen. Standard: "False" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--genre-source-field

Quellfeld für "Genres"-Abschnitt. Standard: "Schlagwörter" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--header-note-source-field

Benutzerdefiniertes Feld, dessen Inhalt als Notiz in den Beschreibungskopf eingefügt wird. Standard: "" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--merge-comments-rule

#<custom field>:[before|after]:[True|False] specifying: <custom field> Custom field containing notes to merge with comments [before|after] Placement of notes with respect to comments [True|False] - A horizontal rule is inserted between notes and comments Default: '::' Applies to: AZW3, EPUB, MOBI output formats

--output-profile

Festlegen des Ausgabeprofils. In einigen Fällen ist ein Ausgabeprofil erforderlich, um den Katalog für ein Gerät zu optimieren. Zum Beispiel erstellt "kindle" oder "kindle_dx" ein strukturiertes Inhaltsverzeichnis mit Abschnitten und Beiträgen. Standard: "None" Wird angewendet auf: AZW3-, ePub- und MOBI-Zielformate

--prefix-rules

Definiert die Regeln um Präfixe zu verwendeten, welche gelesene Bücher, Wunschlisteneinträge und andere benutzerdefinierte Präfixe hervorheben. Das Modell einer Präfixregel ist ('<Regelname>','<Quellenfeld>','<Muster>','<Präfix>'). Wenn mehrere Regeln definiert wurden, wird der erste passende Regel verwendet. Voreinstellung: "(('Read books','tags','+','✓'),('Wishlist item','tags','Wishlist','×'))" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--preset

Verwenden einer benannten Voreinstellung, die mit der Katalogerstellung der Benutzeroberfläche erstellt wurde. Eine Voreinstellung legt alle Einstellungen für die Erstellung eines Katalogs fest. Standard: "None" Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--thumb-width

Größenhinweis (in Inch) für Titelbilder im Katalog. Wertebereich: 1.0 - 2.0 Standard: '1.0' Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

--use-existing-cover

Ersetzen existierender Titelbilder während der Katalogerzeugung. Standard: 'False' Anwendbar auf: AZW3-, EPUB- und MOBI-Zielformate

saved_searches

```
calibredb saved_searches [Optionen] (list|add|remove)
```

Verwalten der in der Calibre-Datenbank gespeicherten Suchen. Falls Sie eine Suche hinzuzufügen, deren Name bereits existiert, so wird die bisherige ersetzt.

Syntax für Hinzufügen („add“):

calibredb **saved_searches** add Suchenname Suchausdruck

Syntax für Entfernen („remove“):

calibredb **saved_searches** remove Suchenname

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

add_custom_column

```
calibredb add_custom_column [Optionen] Bezeichnung Name Datentyp
```

Erstellen einer benutzerdefinierten Spalte. Bezeichnung ist der maschinenfreundliche Name der Spalte, und sollte keine Leerzeichen oder Doppelpunkte enthalten. Name ist der anwenderfreundliche Name der Spalte. Datentyp ist einer von : bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--display

Ein Wörterbuch von Optionen zum Anpassen, wie die Daten in dieser Spalte interpretiert werden. Das ist eine JSON-Zeichenkette. Zum Aufzählen von Spalten verwenden Sie *--display* (die Titelseite 321)"{\ "enum_values\ ":[\ "val1\ ", \ "val2\ "]}" Es gibt einige Optionen, die in der Anzeigevariable enthalten sein können. Diese Optionen nach Art der Spalten sind: Zusammengesetzt – composite: composite_template, composite_sort, make_category,contains_html, use_decorations Datum und Zeit – datetime: date_format Aufzählung – enumeration: enum_values, enum_colors, use_decorations Init, Fluss – int, float: number_format Text – text: is_names, use_decorations Der beste Weg, um gültige Kombinationen zu finden, ist es, eine benutzerdefinierte Spalte des entsprechenden Typs im GUI zu erstellen und dann auf die Sicherung des OPF des Buches zu schauen. (Stellen Sie sicher, dass es ein neues OPF gibt, nachdem die Spalte hinzugefügt wurde.) Sie werden die JSON für die "Anzeige" in der neuen Spalte im OPF sehen.

--is-multiple

Diese Spalte speichert Daten ähnlich wie bei Schlagwörtern (z. B. mehrere kommagetrennte Werte). Gilt nur, wenn der Datentyp Text ist.

custom_columns

```
calibredb custom_columns [Optionen]
```

Auflisten aller verfügbaren benutzerdefinierten Spalten. Zeigt Spaltenbezeichnungen und IDs an.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--details, -d

Details jeder Spalte anzeigen.

remove_custom_column

```
calibredb remove_custom_column [Optionen] Bezeichnung
```

Entfernen der bezeichneten benutzerdefinierten Spalte. Verfügbare Spalten können durch den Befehl custom_columns angezeigt werden.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--force, -f

Nicht nach einer Bestätigung fragen

set_custom

```
calibredb set_custom [Optionen] Spalte ID Wert
```

Festlegen des Werts einer benutzerdefinierte Spalte für das durch die ID bezeichnete Buch. Sie erhalten eine Liste der IDs mithilfe des Suchbefehls. Sie erhalten eine Liste der Namen von benutzerdefinierten Spalten mithilfe des Befehls custom_columns.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--append, -a

Wenn die Spalte mehrere Werte speichert, sollen die angegebenen Werte zu den bestehenden hinzugefügt werden, anstatt sie zu ersetzen.

restore_database

```
calibredb restore_database [options]
```

Restore this database from the metadata stored in OPF files in each folder of the calibre library. This is useful if your metadata.db file has been corrupted.

WARNING: This command completely regenerates your database. You will lose all saved searches, user categories, plugboards, stored per-book conversion settings, and custom recipes. Restored metadata will only be as accurate as what is found in the OPF files.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--really-do-it, -r

Führt die Wiederherstellung tatsächlich durch. Der Befehl wird nicht ausgeführt, wenn diese Option nicht vorhanden ist.

check_library

```
calibredb check_library [Optionen]
```

Durchführen einiger Überprüfungen des Dateisystems einer Bibliothek. Berichte sind invalid_titles, extra_titles, invalid_authors, extra_authors, missing_formats, extra_formats, extra_files, missing_covers, extra_covers, failed_folders

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--csv, -c

Ausgabe als CSV

--ignore_extensions, -e

Kommagetrennte Liste zu ignorierender Erweiterungen. Standardeinstellung: alle

--ignore_names, -n

Kommagetrennte Liste zu ignorierender Namen. Standardeinstellung: alle

--report, -r

Kommagetrenne Liste von Berichten. Standardeinstellung: alle

--vacuum-fts-db

Die Volltextsuchdatenbank aufräumen. Dies kann sehr langsam und arbeitsspeicherlastig sein, je nach Größe der Datenbank.

list_categories

```
calibredb list_categories [options]
```

Produce a report of the category information in the database. The information is the equivalent of what is shown in the Tag browser.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--categories, -r

Kommagetrennte Liste von Kategorie-Nachschlagenamen. Standardeinstellung: alle

--csv, -c

Ausgabe als CSV

--dialect

Typ der zu produzierenden CSV-Datei. Auwahl: excel, excel-tab, unix

--item_count, -i

Nur Anzahl der Einträge in einer Kategorie ausgeben anstelle der Anzahl pro Eintrag in der Kategorie

--width, -w

Maximale Breite einer einzelnen Zeile in der Ausgabe. In der Voreinstellung wird die Bildschirmgröße erkannt.

backup_metadata

```
calibredb backup_metadata [options]
```

Backup the metadata stored in the database into individual OPF files in each books folder. This normally happens automatically, but you can run this command to force re-generation of the OPF files, with the –all option.

Note that there is normally no need to do this, as the OPF files are backed up automatically, every time metadata is changed.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--all

Normalerweise beachtet dieser Befehl nur Bücher, deren OPF-Dateien nicht mehr aktuell sind. Diese Option führt den Befehl für alle Bücher aus.

clone

```
calibredb dupliziere Pfad/zur/neuen/Bibliothek
```

Erstellt eine Kopie der aktuellen Bibliothek. Dies erstellt eine neue, leere Bibliothek mit allen eigenen Spalten, virtulellen Bibliothelen und anderen Einstellungen der aktuellen Bibliothek.

Die geklonte Bibliothek enthält keine Bücher. Wenn Sie eine Kopie mit allen Büchern erstellen möchten, nutzen Sie einfach die Werkzeuge Ihres Dateisystems, um den Bibliotheksordner zu kopieren.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

embed_metadata

```
calibredb embed_metadata [Optionen] Buch_ID
```

Aktualisieren der in den eigentlichen Buchdateien enthaltenen Metadaten auf Basis der in der Calibre-Bibliothek gespeicherten. Die Metadaten werden normalerweise nur aktualisiert, wenn Dateien aus Calibre exportiert werden. Dieser Befehl ist hilfreich, wenn Dateien an Ort und Stelle aktualisiert werden sollen. Beachten Sie, dass unterschiedliche Formate unterschiedliche Metadaten unterstützen. Sie können den besonderen Wert „all“ als Buch_ID verwenden, um die Metadaten aller Bücher zu aktualisieren. Mehrere Bücher können ebenfalls mittels Leerzeichen voneinander getrennter Buch-IDs und mittels Bindestrichen voneinander getrennter ID-Bereiche angegeben werden. Beispiel: calibredb **embed_metadata** 1 2 10-15 23

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--only-formats, -f

Aktualisiert nur Metadaten in Dateien des angegebenen Formats. Wählen Sie die gewünschten Formate durch Anklicken aus. Standardmäßig werden alle Formate aktualisiert.

search

```
calibredb search [options] search expression
```

Search the library for the specified **search** term, returning a comma separated list of book ids matching the **search** expression. The output format is useful to feed into other commands that accept a list of ids as input.

The **search** expression can be anything from calibre's powerful **search** query language, for example: calibredb **search** author:asimov ,title:“i robot“‘

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--limit, -l

Maximale Anzahl der anzuzeigenden Ergebnisse. Standard ist alle Ergebnisse.

fts_index

```
calibredb fts_index [options] enable/disable/status/reindex
```

Control the Full text search indexing process.

enable
Turns on FTS indexing for this library

disable
Turns off FTS indexing for this library

status
Shows the current indexing status

reindex
Can be used to re-index either particular books or the entire library. To re-index particular books specify the book ids as additional arguments after the reindex command. If no book ids are specified the entire library is re-indexed.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--indexing-speed
Die Geschwindigkeit der Indizierung. Verwende schnell für schnelles Indizieren unter Verwendung der gesamten Rechnerkapazität und langsam für weniger rechenlastiges Indizieren. Beachte, dass die Geschwindigkeit nach jedem Aufruf auf langsam zurückgesetzt wird.

--wait-for-completion
Warte bis alle Bücher indiziert sind, zeige Indizierungsprozess periodisch

fts_search

```
calibredb fts_search [options] search expression
```

Do a full text search on the entire library or a subset of it.

Wann immer Sie Argumente mit Leerzeichen an calibredb weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

--do-not-match-on-related-words
Only match on exact words not related words. So correction will not match correcting.

--include-snippets
Include snippets of the text surrounding each match. Note that this makes searching much slower.

--indexing-threshold
How much of the library must be indexed before searching is allowed, as a percentage. Defaults to 90

--match-end-marker
The marker used to indicate the end of a matched word inside a snippet

--match-start-marker
The marker used to indicate the start of a matched word inside a snippet

--output-format
The format to output the search results in. Either `"text"` for plain text or `"json"` for JSON output.

--restrict-to
Restrict the searched books, either using a search expression or ids. For example: ids:1,2,3 to restrict by ids or search:tag:foo to restrict to books having the tag foo.

13.1.7 ebook-convert

```
ebook-convert input_file output_file [options]
```

Convert an e-book from one format to another.

input_file is the input and output_file is the output. Both must be specified as the first two arguments to the command.

The output e-book format is guessed from the file extension of output_file. output_file can also be of the special format .EXT where EXT is the output file extension. In this case, the name of the output file is derived from the name of the input file. Note that the filenames must not start with a hyphen. Finally, if output_file has no extension, then it is treated as a folder and an „open e-book“ (OEB) consisting of HTML files is written to that folder. These files are the files that would normally have been passed to the output plugin.

After specifying the input and output file you can customize the conversion by specifying various options. The available options depend on the input and output file types. To get help on them specify the input and output file and then use the -h option.

For full documentation of the conversion system see *eBook Konvertierung* (die Titelseite 61)

Wann immer Sie Argumente mit Leerzeichen an **ebook-convert** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

Die Optionen und Standardwerte für die Optionen ändern sich je nach dem gewählten Quell- und Zielformat. Überprüfen Sie deshalb diese immer mit nachfolgendem Befehl:

ebook-convert myfile.input_format myfile.output_format -h

Nachfolgend sind die Optionen beschrieben, die für alle Konvertierungen gleich sind, gefolgt von den spezifischen Optionen zu jedem Quell- und Zielformat.

- *EINGABE-OPTIONEN* (die Titelseite 327)
- *AUSGABE-OPTIONEN* (die Titelseite 327)
- *ERSCHEINUNGSBILD* (die Titelseite 328)
- *HEURISTISCHE VERARBEITUNG* (die Titelseite 331)
- *SUCHEN UND ERSETZEN* (die Titelseite 332)
- *STRUKTURERKENNUNG* (die Titelseite 332)
- *INHALTSVERZEICHNIS* (die Titelseite 333)
- *METADATEN* (die Titelseite 334)
- *DIAGNOSE* (die Titelseite 335)

--help, -h
: Diesen Hilfetext anzeigen und beenden.

--input-profile
: Geben Sie das Eingabeprofil an. Das Eingabeprofil stellt dem Konvertierungssystem Informationen zur Verfügung, wie diverse Informationen des Eingabedokuments zu interpretieren sind. Zum Beispiel auflösungsabhängige Längen (z. B. Längen in Pixeln). Auswahlmöglichkeiten sind: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes
: Anzeigen der Namen der integrierten Nachrichtenquellen. Sie können aus einer integrierten Quelle ein eBook wie folgt erstellen: ebook-convert "Recipe Name.recipe" output.epub

--output-profile

Geben sie das Ausgabeprofil an. Das Ausgabeprofil bestimmt, wie das Konvertierungssystem das erstellte Dokument für das angegebene Gerät optimiert. In einigen Fällen kann ein Ausgabeprofil verwendet werden, um die Ausgabe für ein bestimmtes Gerät zu optimieren, aber dies ist selten notwendig. Auswahlmöglichkeiten sind:cybookg3, cybook_opus, default, generic_eink, generic_eink_large, generic_eink_hd, 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, pocketbook_inkpad3, pocketbook_lux, pocketbook_hd, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet

--version

Programmversion anzeigen und beenden

EINGABE-OPTIONEN

Optionen zur Verarbeitungssteuerung der Quelldatei mobi

--input-encoding

Geben Sie die Zeichencodierung des Eingabedokuments an. Falls verwendet, werden alle im Dokument angegebenen Codierungen aufgehoben. Besonders hilfreich bei Dokumenten, die keine oder fehlerhafte Codierungen enthalten.

AUSGABE-OPTIONEN

Optionen zur Verarbeitungssteuerung der Ausgabe epub

--dont-split-on-page-breaks

Das Aufteilen bei Seitenwechseln ausschalten. Normalerweise werden Eingabe-Dateien automatisch bei jedem Seitenwechsel in zwei Dateien aufgeteilt. Damit erhält man in der Ausgabe ein eBook, das schneller mit weniger Resourcen analysiert werden kann. Das Aufteilen ist jedoch langsam und falls die Ausgangsdatei sehr viele Seitenwechsel enthält, sollten Sie das Aufteilen bei Seitenwechseln ausschalten.

--epub-flatten

Diese Option ist nur nötig, wenn Sie beabsichtigen das EPUB mit einem FBReaderJ zu benutzen. Diese Option wird die Ordnerstruktur des EPUB's vereinfachen, indem alle Dateien in der obersten Ebene abgelegt werden.

--epub-inline-toc

Inline-Inhaltsverzeichnis einfügen, das als Teil des Buchhauptinhalts erscheinen wird.

--epub-max-image-size

The maximum image size (width x height). A value of none means use the screen size from the output profile. A value of profile means no maximum size is specified. For example, a value of 100x200 will cause all images to be resized so that their width is no more than 100 pixels and their height is no more than 200 pixels. Note that this only affects the size of the actual image files themselves. Any given image may be rendered at a different size depending on the styling applied to it in the document.

--epub-toc-at-end

Eingefügtes Inline-Inhaltsverzeichnis am Buchende anstatt am -anfang positionieren.

--epub-version

Die zu generierende Version der EPUB-Datei. EPUB 2 ist am besten kompatibel. Verwenden Sie EPUB 3 nur, wenn Sie wissen, dass Sie es wirklich benötigen.

--extract-to

Extract the contents of the generated book to the specified folder. The contents of the folder are first deleted, so be careful.

--flow-size

Teile alle HTML Dateien größer als die angegebene Größe (in KB). Das ist notwendig, da nicht alle EPUB Lesegeräte große Dateien verarbeiten können. Der Standard von 260KB ist die Größe, die für Adobe Digital Editions verarbeitet wird. Auf 0 setzen, um das Teilen anhand der Dateigröße zu deaktivieren.

--no-default-epub-cover

Falls die Quelldatei keine Titelbild enthält und Sie eine solche auch nicht angeben, wird normalerweise eine standardisiertes Titelbild mit Titel, Autoren usw. erstellt. Diese Einstellung deaktiviert dessen Erstellung.

--no-svg-cover

SVG nicht als Titelbild benutzen. Aktivieren Sie diese Option, falls Ihre EPUB-Datei auf einem Gerät benutzt werden soll, welches SVG nicht unterstützt (z. B. iPhone oder JetBook Lite). Ohne diese Option werden solche Geräte eine leere Seite als Titelbild anzeigen.

--preserve-cover-aspect-ratio

Bei Verwendung eines SVG-Titelbildes führt diese Einstellung dazu, dass das Titelbild auf die verfügbare Bildschirmgröße skaliert wird, aber dennoch dessen Seitenverhältnis (Verhältnis von Breite zu Höhe) erhalten bleibt. Das heißt, es können weiße Ränder an den Seiten oder oben und unten auf dem Bild enthalten sein, aber das Bild wird nie verzerrt werden. Ohne diese Einstellung ist das Bild möglicherweise leicht verzerrt, jedoch ohne Ränder.

--pretty-print

Falls angegeben, versucht die Ausgabeerweiterung ein für Menschen möglichst verständliches Ergebnis zu erzeugen. Führt bei einigen Ausgabeerweiterungen eventuell zu keinem Effekt.

--toc-title

Title for any generated inline table of contents.

ERSCHEINUNGSBILD

Optionen zur Steuerung des Erscheinungsbilds der Ausgabe

--asciiize

Transliteriert Unicode-Zeichen in eine ASCII-Darstellung. Seien Sie vorsichtig, denn dadurch werden Unicode-Zeichen durch ASCII-Zeichen ersetzt. Zum Beispiel wird "Pelé" durch "Pele" ersetzt. Beachten Sie außerdem, dass in Fällen, in denen es mehrere Darstellungen eines Zeichens gibt (z. B. Zeichen, die von Chinesisch und Japanisch gemeinsam verwendet werden), die Darstellung verwendet wird, die auf der aktuellen Sprache der Calibre-Schnittstelle basiert.

--base-font-size

Die Bezugsschriftgröße in pts. Alle Schriftgrößen im erstellten E-Book werden basierend auf dieser Größe skaliert. Mit einer größeren Schriftgröße erhalten Sie eine größere Schrift in der Ausgabe und umgekehrt. Laut Voreinstellung basiert die Bezugsschriftgröße auf dem gewählten Ausgabe-Profil.

--change-justification

Ändern der Textausrichtung. Die Angabe "Links" konvertiert den gesamten Blocksatz der Quelle zu Links ausgerichtet (also ohne Blocksatz). Die Angabe "Blocksatz" konvertiert die Ausrichtung zu Blocksatz. Und die Angabe "Original" (Standardeinstellung) ändert die Ausrichtung gegenüber der Quelldatei nicht. Beachten Sie, dass nicht alle Zielformate die Textausrichtung unterstützen.

--disable-font-rescaling

Jegliche Neuskalierung von Schriftgrößen deaktivieren.

--embed-all-fonts

Alle Schriftarten einbetten, die im Eingabedokument referenziert, jedoch noch nicht eingebettet sind. Ihr System wird dabei nach den Schriftarten durchsucht und - falls gefunden - diese eingebettet. Einbettungen werden nur funktionieren, wenn das Zielformat das Einbetten von Schriftarten unterstützt, wie z. B. bei EPUB, AZW3, DOCX

oder PDF. Bitte stellen Sie sicher, dass Sie über die richtige Lizenz für das Einbetten der verwendeten Schriftarten verfügen.

--embed-font-family

Die angegebene Schriftfamilie in das Buch einbetten. Dadurch wird die "Basisschriftart" festgelegt, die für das Buch verwendet wird. Falls das Quelldokument seine eigenen Schriftarten bestimmt, können diese die Basisschriftart außer Kraft setzen. Sie können Stilangaben filtern, um Schriftarten aus dem Quelldokument zu entfernen. Beachten Sie, dass das Einbetten von Schriftarten nur mit einigen Zielformaten funktioniert; prinzipiell sind das die Formate EPUB, AZW3 und DOCX.

--expand-css

Standardmäßig wird Calibre die Kurzform bei verschiedenen CSS Eigenschaften wie margin, padding, border, usw. verwenden.

Diese Option wird stattdessen die Verwendung der ungekürzte Form aktivieren. Beachten Sie, dass CSS immer erweitert wird, wenn EPUB Dateien generiert werden, bei denen das Ausgabeprofil auf eines der Nook Profile eingestellt ist, da der Nook verkürztes CSS nicht verarbeiten kann.

--extra-css

Entweder der Pfad zu einem CSS-Stylesheet oder Roh-CSS. Dieses CSS wird an die Style-Regeln der Quelldatei angehängt, sodass es zum Außerkraftsetzen dieser Regeln verwendet werden kann.

--filter-css

Eine kommagetrennte Liste aller CSS-Eigenschaften, die aus allen CSS-Style-Regeln entfernt werden. Dies ist hilfreich, falls die Anwesenheit einiger Style-Informationen deren Außerkraftsetzen auf Ihrem Gerät verhindert. Zum Beispiel: font-family (Schriftfamilie), color (Farbe), margin-left (Randbreite rechts), margin-right (Randbreite links)

--font-size-mapping

Zuordnung von CSS-Schriftnamen zu Schriftgrößen in Punkten. Eine Beispieleinstellung ist 12,12,14,16,18,20,22,24. Dies sind die Zuordnungen für die Größen xx-small bis xx-large, mit der letzten Größe für riesige Schriftarten. Der Algorithmus für die Schriftartneuskalierung verwendet diese Größen für die intelligente Neuskalierung von Schriftarten. Voreinstellung ist die Verwendung einer Zuordnung auf Basis des gewählten Ausgabeprofils.

--insert-blank-line

Leerzeile zwischen Absätzen einfügen. Funktioniert nur, wenn die Quelldatei Absätze verwendet (<p>- oder -HTML-Tags).

--insert-blank-line-size

Höhe der eingefügten Leerzeilen (in em) festlegen. Die Höhe der Zeilen zwischen Absätzen wird das Doppelte des hier festgelegten Wertes betragen.

--keep-ligatures

Preserve ligatures present in the input document. A ligature is a combined character of a pair of characters like ff, fi, fl et cetera. Most readers do not have support for ligatures in their default fonts, so they are unlikely to render correctly. By default, calibre will turn a ligature into the corresponding pair of normal characters. Note that ligatures here mean only unicode ligatures not ligatures created via CSS or font styles. This option will preserve them instead.

--line-height

Zeilenhöhe in Punkten. Steuert den Abstand zwischen aufeinanderfolgenden Zeilen von Text. Gilt nur für Elemente, die ihre eigene Zeilenhöhe nicht selbst festlegen. In den meisten Fällen ist die Option zum Festlegen der minimalen Zeilenhöhe vorzuziehen. Standardmäßig wird keine Änderung der Zeilenhöhe durchgeführt.

--linearize-tables

Einige schlecht gestaltete Dokumente verwenden Tabellen für das Layout von Text auf der Seite. Bei der Konvertierung führt dies häufig zu Text außerhalb der Seite und anderen Artefakten. Diese Einstellung extrahiert den Tabelleninhalt und gibt ihn linear wieder.

--margin-bottom

Festlegen des unteren Randes in pts. Standard ist 5.0. Bei einem Wert von weniger als null wird kein Rand erzeugt (die Randeinstellung im Originaldokument bleibt erhalten). Hinweis: Blattorientierte Formate wie PDF oder DOCX besitzen ihre eigenen Randeinstellungen, die Vorrang haben.

--margin-left

Festlegen des linken Randes in pts. Standard ist 5.0. Bei einem Wert von weniger als null wird kein Rand erzeugt (die Randeinstellung im Originaldokument bleibt erhalten). Hinweis: Blattorientierte Formate wie PDF oder DOCX besitzen ihre eigenen Randeinstellungen, die Vorrang haben.

--margin-right

Festlegen des rechten Randes in pts. Standard ist 5.0. Bei einem Wert von weniger als null wird kein Rand erzeugt (die Randeinstellung im Originaldokument bleibt erhalten). Hinweis: Blattorientierte Formate wie PDF oder DOCX besitzen ihre eigenen Randeinstellungen, die Vorrang haben.

--margin-top

Festlegen des oberen Randes in pts. Standard ist 5.0. Bei einem Wert von weniger als null wird kein Rand erzeugt (die Randeinstellung im Originaldokument bleibt erhalten). Hinweis: Blattorientierte Formate wie PDF oder DOCX besitzen ihre eigenen Randeinstellungen, die Vorrang haben.

--minimum-line-height

Minimale Zeilenhöhe, ausgedrückt als Prozentsatz der berechneten Schriftgröße des Elementes. Calibre wird sicherstellen, dass jedes Element diese eingestellte Mindesthöhe hat, unabhängig von der Vorgabe des Eingabedokuments. Zum Deaktivieren auf Null setzen. Standardeinstellung ist 120%. Verwenden Sie diese Einstellung anstatt der direkten Angabe der Zeilenhöhe, es sei denn, Sie wissen was Sie tun. Beispielsweise können Sie einen doppelzeiligen Text erreichen, indem Sie den Wert auf 240 setzten.

--remove-paragraph-spacing

Abstand zwischen Absätzen entfernen. Fügt einen Zeileneinzug bei Absätzen von 1,5 em ein. Die Entfernung des Abstands funktioniert nur bei Quelldateien, die Absätze verwenden (<p>- oder -HTML-Tags).

--remove-paragraph-spacing-indent-size

Wenn Calibre Leerzeilen zwischen Absätzen entfernt, wird automatisch ein Absatzeinzug hinzugefügt, um Absätze einfach unterscheidbar darzustellen. Diese Option legt die Größe des Einzugs (in em) fest. Bei einem negativen Wert wird der im Eingabedokument angegebene Absatzeinzug verwendet, d. h. Calibre ändert dann Einzüge nicht.

--smarten-punctuation

Convert plain quotes, dashes and ellipsis to their typographically correct equivalents. For details, see https://daringfireball.net/projects/smartypants.

--subset-embedded-fonts

Alle eingebetteten Schriftarten reduzieren. Jede eingebettete Schriftart wird so reduziert, dass Sie nur Glyphen enthält, die in diesem Dokument verwendet werden. Dadurch verringert sich die Größe der Schriftdateien. Hilfreich, falls Sie eine besonders umfangreiche Schriftart mit vielen unbenutzten Glyphen einbetten.

--transform-css-rules

Pfad zu einer Datei, die Regeln zum Umwandeln der CSS-Styles in diesem Buch enthält. Der einfachste Weg, um eine solche Datei zu erstellen, ist die Verwendung des Assistenten zum Erstellen von Regeln in Calibres Benutzeroberfläche. Sie erreichen diesen über den Bereich "Layout" → "Styles umwandeln" des Dialogs "Bücher konvertieren". Sobald Sie Regeln erstellen, können Sie die Schaltfläche "Exportieren" verwenden, um diese in eine Datei zu speichern.

--transform-html-rules

Path to a file containing rules to transform the HTML in this book. The easiest way to create such a file is to use the wizard for creating rules in the calibre GUI. Access it in the "Look & feel->Transform HTML" section of the conversion dialog. Once you create the rules, you can use the "Export" button to save them to a file.

--unsmarten-punctuation

Typografische Anführungszeichen, Gedankenstriche und Auslassungspunkte in ihre gewöhnlichen Entsprechungen umwandeln.

HEURISTISCHE VERARBEITUNG

Dokumenttext und -struktur durch Anwendung häufiger Muster verändern. Standardmäßig deaktiviert. Zum Aktivieren –enable-heuristics verwenden. Individuelle Aktionen können mit den –disable-*-Optionen deaktiviert werden.

--disable-dehyphenate

Wörter mit Bindestrichen im gesamten Dokument analysieren. Das Dokument selbst wird als Wörterbuch zur Feststellung verwendet, ob Bindestriche beibehalten oder entfernt werden sollen.

--disable-delete-blank-paragraphs

Leere Absätze aus dem Dokument entfernen, falls diese zwischen allen anderen Absätzen liegen.

--disable-fix-indents

Einrückungen, die durch mehrere geschützte Leerzeichen erstellt wurden, in CSS-Einrückungen umwandeln.

--disable-format-scene-breaks

Linksbündige Markierungen für Abschnittswechsel werden zentriert. Weiche Abschnittswechsel, die mehrere Leerzeilen verwenden, werden mit horizontalen Linien ersetzt.

--disable-italicize-common-cases

Nach gängigen Wörtern und Mustern suchen, die auf Kursivschrift hindeuten und diese entsprechend formatieren.

--disable-markup-chapter-headings

Unformatierte Kapitelüberschriften und Unterüberschriften erkennen und in h2- und h3-HTML-Tags ändern. Diese Einstellung erstellt kein Inhaltsverzeichnis, kann aber in Kombination mit der Strukturerkennung verwendet werden, um eines zu erstellen.

--disable-renumber-headings

Sucht aufeinanderfolgende <h1>- oder <h2>-HTML-Tags. Um Trennungen innerhalb von Kapitelüberschriften zu verhinden, werden die HTML-Tags neu nummeriert.

--disable-unwrap-lines

Zeilenumbrüche basierend auf Satzzeichen und anderen Hinweisen entfernen.

--enable-heuristics

Heuristische Verarbeitung aktivieren. Falls deaktiviert, wird keinerlei heuristische Verarbeitung durchgeführt.

--html-unwrap-factor

Skala zur Festlegung der Zeilenlänge, bei der Zeilenumbrüche entfernt werden sollen. Gültige Werte sind Dezimalzahlen zwischen 0 und 1. Die Standardeinstellung ist 0,4, gerade unter der mittleren Zeilenlänge. Reduzieren Sie diesen Wert, falls nur einige wenige Zeilen des Dokuments ein Entfernen von Umbrüchen benötigen.

--replace-scene-breaks

Abschnittswechsel mit angegebenem Text ersetzen. Standardmäßig wird der Text des Eingabedokuments verwendet.

SUCHEN UND ERSETZEN

Text und Struktur des Dokument ändern durch Nutzung von benutzerdefinierten Mustern.

--search-replace

Pfad zu einer Datei mit regulären Ausdrücken für das Suchen und Ersetzen. Die Datei muss abwechselnd Zeilen mit einem regulären Ausdruck, gefolgt von einem Austauschmuster (das eine leere Zeile sein kann) enthalten. Der reguläre Ausdruck muss der Python-Regex-Syntax entsprechen und die Datei muss in UTF-8 codiert sein.

--sr1-replace

Ersatz zum Ersetzen des mit "sr1-search" gefundenen Textes.

--sr1-search

Suchmuster (regulärer Ausdruck), das durch "sr1-replace" ersetzt werden soll.

--sr2-replace

Ersatz zum Ersetzen des mit "sr2-search" gefundenen Textes.

--sr2-search

Suchmuster (regulärer Ausdruck), das durch "sr2-replace" ersetzt werden soll.

--sr3-replace

Ersatz zum Ersetzen des mit "sr3-search" gefundenen Textes.

--sr3-search

Suchmuster (regulärer Ausdruck), das durch "sr3-replace" ersetzt werden soll.

STRUKTURERKENNUNG

Kontrolle der automatischen Erkennung der Dokumentstruktur.

--add-alt-text-to-img

When an <img> tag has no alt attribute, check the associated image file for metadata that specifies alternate text, and use it to fill in the alt attribute. The alt attribute improves accessibility by providing text descriptions for users who cannot see or fully interpret visual content.

--chapter

XPath-Ausdruck zum Erkennen von Kapitelüberschriften. Standardmässig werden <h1>- oder <h2>-Tags, welche die Wörter "chapter", "book", "section", "prologue", "epilogue" oder "part" enthalten, als Kapitelüberschrift behandelt, ebenso wie alle Tags, die class="chapter" enthalten. Der verwendete Ausdruck muss eine Liste von Elementen bewerten. Um die Kapitelerkennung zu deaktivieren, verwenden Sie den Ausdruck "/". Siehe das XPath-Tutorial im Calibre-Handbuch für weitere Hilfe zu diesem Feature.

--chapter-mark

Geben Sie an, wie erkannte Kapitel gekennzeichnet werden sollen. Der Wert "pagebreak" fügt Seitenumbrüche vor Kapiteln ein. Der Wert "rule" fügt eine Linie vor Kapiteln ein. Der Wert "none" schaltet die Kapitelmarkierung aus und der Wert "both" verwendet sowohl Seitenumbrüche als auch Linien zur Kapitelmarkierung.

--disable-remove-fake-margins

Einige Dokumente definieren Blattränder durch die Angabe eines rechten und linken Randes für jeden Absatz. Calibre wird versuchen, diese Ränder zu erkennen und zu entfernen. Manchmal kann dies das Entfernen von Rändern verursachen, die hätten beibehalten werden sollen. In diesem Fall können Sie die Entfernung deaktiveren.

--insert-metadata

Metadaten des Buchs am Buchanfang einfügen. Hilfreich, falls Ihr eBook-Reader die direkte Anzeige/Suche von Metadaten nicht unterstützt.

--page-breaks-before

XPath-Ausdruck. Seitenumbrüche werden vor den angegebenen Elementen eingefügt. Verwenden Sie zur Deaktivierung den Ausdruck: `"/"`.

--prefer-metadata-cover

Erkanntes Titelbild der Quelldatei bevorzugt verwenden anstatt des angegebenen.

--remove-first-image

Erstes Bild des Eingabe-eBooks entfernen. Hilfreich, falls das Eingabedokument ein Titelbild besitzt, das nicht als solches erkannt wurde. Ohne diese Option würde in diesem Fall das Hinzufügen eines Titelbildes mit Calibre zu zwei Titelbildern im Ausgabedokument führen.

--start-reading-at

XPath-Ausdruck zur Erkennung der Stelle im Dokument, an der mit dem Lesen begonnen wird. Einige eBook-Leseprogramme (allem voran der Kindle) verwenden diese Stelle als die Position, an der das Buch geöffnet wird. Siehe das XPath-Tutorial im Calibre-Benutzerhandbuch für weitere Hilfe zur Benutzung dieser Funktion.

INHALTSVERZEICHNIS

Kontrolle der automatischen Erstellung eines Inhaltsverzeichnisses. Laut Voreinstellung wird, falls die Ursprungsdatei ein Inhaltsverzeichnis hat, dieses verwendet anstatt des automatisch erstellten.

--duplicate-links-in-toc

Beim Erstellen eines Inhaltsverzeichnisses aus Verweisen des Eingabedokuments doppelte Einträge erlauben. D. h. mehr als einen Eintrag mit demselben Text erlauben, solange diese auf unterschiedliche Orte verweisen.

--level1-toc

XPath-Ausdruck, der alle Tags bestimmt, die zur ersten Ebene des Inhaltsverzeichnisses hinzugefügt werden sollen. Falls angegeben, hat der Ausdruck Vorrang vor anderen Formen der automatischen Erkennung. Siehe das XPath-Tutorial im Calibre-Benutzerhandbuch für Beispiele.

--level2-toc

XPath-Ausduck, der alle Tags bestimmt, die zur zweiten Ebene des Inhaltverzeichnisses hinzugefügt werden sollen. Jeder Eintrag wird unter dem vorherigen Eintrag der ersten Ebene hinzugefügt. Siehe das XPath-Tutorial im Calibre-Benutzerhandbuch für Beispiele.

--level3-toc

XPath-Ausdruck, der alle Tags bestimmt, die zur dritten Ebene des Inhaltverzeichnisses hinzugefügt werden sollen. Jeder Eintrag wird unter dem vorherigen Eintrag der zweiten Ebene hinzugefügt. Siehe das XPath-Tutorial im Calibre-Benutzerhandbuch für Beispiele.

--max-toc-links

Höchstzahl an Verweisen, die in das Inhaltsverzeichnis eingefügt werden. Zum Deaktivieren auf 0 setzen. Voreinstellung ist: 50. Verweise werden nur dann zum Inhaltsverzeichnis hinzugefügt, wenn weniger Kapitel als im Schwellenwert angegeben erkannt wurden.

--no-chapters-in-toc

Automatisch erkannte Kapitel nicht zum Inhaltsverzeichnis hinzufügen.

--toc-filter

Einträge aus dem Inhaltsverzeichnis entfernen, deren Titel dem angegebenen regulären Ausdruck entsprechen. Übereinstimmende Einträge und deren untergeordnete Einträge werden entfernt.

--toc-threshold

Falls weniger Kapitel als die hier angegebe Anzahl erkannt wurden, werden Verweise zum Inhaltsverzeichnis hinzugefügt. Voreinstellung: 6

--use-auto-toc

Normalerweise wird, falls die Quelldatei bereits ein Inhaltsverzeichnis besitzt, dieses vorzugsweise verwendet anstatt des automatisch erstellten. Mit dieser Option wird immer das automatisch erstellte verwendet.

METADATEN

Optionen zum Festlegen von Metadaten in der Ausgabe

--author-sort

Für die Sortierung nach Autor zu verwendende Zeichenfolge.

--authors

Geben Sie die Autoren an. Mehrere Autoren sollten durch &-Zeichen voneinander getrennt werden.

--book-producer

Geben Sie den Hersteller des Buches an.

--comments

Geben Sie die Beschreibung des eBooks an.

--cover

Geben Sie eine Datei oder URL mit dem Titelbild an.

--isbn

Geben Sie die ISBN des Buches an.

--language

Geben Sie die Sprache an.

--pubdate

Veröffentlichungsdatum festlegen (wird in der lokalen Zeitzone angenommen, es sei denn, die Zeitzone wird explizit angegeben)

--publisher

Geben Sie den Verlag des eBooks an.

--rating

Geben Sie die Bewertung an. Dies sollte eine Zahl zwischen 1 und 5 sein.

--read-metadata-from-opf, --from-opf, -m

Metadaten aus angegebener OPF-Datei lesen. Die aus dieser Datei gelesenen Metadaten haben Vorrang vor jeglichen Metadaten in der Quelldatei.

--series

Geben Sie die Serie an, zu der dieses eBook gehört.

--series-index

Geben Sie den Index des Buches in dieser Serie an.

--tags

Geben Sie die Schlagwörter für das Buch an. Dies sollte eine kommagetrennte Liste sein.

--timestamp

Geben Sie den Buchzeitstempel an (wird nirgends mehr verwendet).

--title

Geben Sie den Titel an.

--title-sort

Für die Sortierung zu verwendende Version des Titels.

DIAGNOSE

Optionen zur Hilfe bei der Konvertierungs-Fehlerdiagnose

--debug-pipeline, -d

Save the output from different stages of the conversion pipeline to the specified folder. Useful if you are unsure at which stage of the conversion process a bug is occurring.

--verbose, -v

Ausführlichkeitsgrad. Mehrfach angeben für größere Ausführlichkeit. Zweifache Angabe führt zu vollem, einfache Angabe zu einfachem und überhaupt keine Angabe zu geringstem Detailgrad. Beispiel für Level 2: ebook-convert a.epub b.epub *--verbose* (die Titelseite 335) *--verbose* (die Titelseite 335) oder ebook-convert a.epub b.epub -vv

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.

Wann immer Sie Argumente mit Leerzeichen an **ebook-edit** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--detach

Calibre vom Terminalfenster ablösen, falls vorhanden (nur für Linux)

--help, -h

Diesen Hilfetext anzeigen und beenden.

--select-text

The text to select in the book when it is opened for editing

--version

Programmversion anzeigen und beenden

13.1.9 ebook-meta

```
ebook-meta eBook_Datei [Optionen]
```

Metadaten lesen/schreiben aus/in eBook-Dateien

Unterstützte Formate zum Lesen von Metadaten: azw, azw1, azw3, azw4, cb7, cbc, cbr, cbz, chm, docx, epub, fb2, fbz, html, htmlz, imp, kepub, lit, lrf, lrx, mobi, odt, oebzip, opf, pdb, pdf, pml, pmlz, pobi, prc, rar, rb, rtf, snb, tpz, txt, txtz, updb, zip

Unterstützte Formate zum Schreiben von Metadaten: azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, kepub, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

Unterschiedliche Dateitypen unterstützen unterschiedliche Arten von Metadaten. Falls Sie versuchen, einige Metadaten für einen Dateityp festzulegen, der dies nicht unterstützt, werden die Metadaten ohne Rückmeldung ignoriert.

Wann immer Sie Argumente mit Leerzeichen an **ebook-meta** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--author-sort

Zeichenfolge, die für die Sortierung nach Autor verwendet wird. Falls hier nichts, jedoch der Autor angegeben ist, wird sie aus dem Autorenfeld automatisch erstellt.

--authors, -a

Legen Sie die Autoren fest. Mehrere Autoren sollten durch ein &-Zeichen getrennt werden. Autorennamen sollten in der Reihenfolge Vorname Nachname angegeben werden.

--book-producer, -k

Geben Sie den Hersteller des Buches an.

--category

Geben Sie die Kategorie des Buches an.

--comments, -c

Geben Sie die Beschreibung des eBooks an.

--cover

Titelbild auf die angegebene Datei festlegen.

--date, -d

Geben Sie das Erscheinungsdatum an.

--disallow-rendered-cover

For formats like EPUB that use a "default cover" of the first page rendered, disallow such default covers

--from-opf

Metadaten aus der angegebenen OPF-Datei lesen und als Metadaten im eBook verwenden. In der Befehlszeile angegebene Metadaten übergehen die aus der OPF-Datei gelesenen Metadaten.

--get-cover

Titelbild des eBooks als die angegebene Datei speichern.

--help, -h

Diesen Hilfetext anzeigen und beenden.

--identifier

Festlegen der Kennungen des Buches. Kann mehrmals angegeben werden. Zum Beispiel: *--identifier* (die Titelseite 336) uri:https://acme.com *--identifier* (die Titelseite 336) isbn:12345. Um eine Kennung zu entfernen, geben Sie keinen Wert an: *--identifier* (die Titelseite 336) isbn. Beachten Sie, dass bei EPUB-Dateien Kennungen, die als Paketkennung markiert sind, nicht entfernt werden können.

--index, -i

Geben Sie den Index des Buches in dieser Serie an.

--isbn

Geben Sie die ISBN des Buches an.

--language, -l

Geben Sie die Sprache an.

--lrf-bookid

Buch-ID in LRF-Dateien festlegen

--publisher, -p

Geben Sie den Verlag des eBooks an.

--rating, -r

Geben Sie die Bewertung an. Dies sollte eine Zahl zwischen 1 und 5 sein.

--series, -s

Geben Sie die Serie an, zu der dieses eBook gehört.

--tags

Geben Sie die Schlagwörter für das Buch an. Dies sollte eine kommagetrennte Liste sein.

--title, -t

Geben Sie den Titel an.

--title-sort

Version des Titels, die für die Sortierung verwendet wird. Falls hier nichts, jedoch der Titel angegeben ist, wird die Version automatisch aus dem Titel erstellt.

--to-opf

Geben Sie den Namen einer OPF-Datei an. Die Metadaten werden in die OPF-Datei geschrieben.

--version

Programmversion anzeigen und beenden

13.1.10 ebook-polish

```
ebook-polish [Optionen] Quelldatei [Ausgabedatei]
```

Bei Bücher perfektionieren dreht sich alles um den Feinschliff und die Perfektionierung des sorgfältig gestalteten Buches.

Perfektionieren verändert den Code im Buch nur minimal. Im Gegensatz zur Konvertierung findet hierbei keine Änderung von CSS-Struktur, Dateinamen, Schriftgrößen, Rändern usw. statt. Jede Aktion führt nur ein Mindestmaß an Korrekturen für den gewünschten Effekt aus

Dieses Werkzeug sollte als letzter Schritt bei der Erstellung Ihres eBooks eingesetzt werden.

„Bücher perfektionieren“ funktioniert nur bei Büchern in den Formaten AZW3 oder EPUB oder KEPUB.

Wann immer Sie Argumente mit Leerzeichen an **ebook-polish** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--add-soft-hyphens, -H

Weiche Trennzeichen zu allen Wörtern des Buchs hinzufügen. Bei bündig ausgerichtetem Text ermöglicht dies eine bessere Darstellung, falls das Lesegerät keine Silbentrennung unterstützt.

--compress-images, -i

Komprimieren Sie verlustfrei die Bilder im Buch, um die Dateigröße zu reduzieren, ohne die Bildqualität zu beeinflussen.

--cover, -c

Pfad zu einem Titelbild. Ändert das im eBook festgelegte Titelbild. Falls kein Titelbild vorhanden oder dieses nicht korrekt identifiziert ist, wird ein neues Titelbild eingefügt.

--download-external-resources, -d

Download external resources such as images, stylesheets, etc. that point to URLs instead of files in the book. All such resources will be downloaded and added to the book so that the book no longer references any external resources.

--embed-fonts, -e

Alle Schriftarten einbetten, die im Dokument referenziert, aber bisher noch nicht eingebettet sind. Hierbei werden Schriftarten auf Ihrem System gesucht und im Dokument eingebettet, wenn sie gefunden werden. Bitte stellen Sie sicher, dass Sie über die richtige Lizenz für die Einbettung dieser Schriftarten verfügen.

--help, -h

Diesen Hilfetext anzeigen und beenden.

--jacket, -j

"Buchumschlag" als erste Seite des Buches einfügen, welche alle Metadaten des Buches wie Titel, Schlagwörter, Autoren, Serie, Kommentare usw. beinhaltet. Jeder bisherige Buchumschlag wird ersetzt.

--opf, -o

Pfad zu einer OPF-Datei. Die Metadaten in dem Buch werden aus der OPF-Datei aktualisiert.

--remove-jacket

Zuvor eingefügte Buchumschlagseite entfernen.

--remove-soft-hyphens

Weiche Trennzeichen von sämtlichem Text des Buchs entfernen.

--remove-unused-css, -u

Alle nicht genutzten CSS Regeln vom Stylesheets und <Style> tags entfernt. Einige Bücher die von Templates abhängen könnten eine Menge extra CSS Regeln enthalten die mit im aktuellen Inhalt nicht verwendet werden. Diese extra Regeln könnten Betrachter verlangsamen weil sie das alles verarbeiten müssen.

--smarten-punctuation, -p

Gewöhnliche Gedankenstriche, Auslassungspunkte, Anführungszeichen, mehrere Bindestriche in Folge usw. in ihre typografisch korrekten Entsprechungen umwandeln. Beachten Sie, dass der Algorithmus manchmal zu falschen Ergebnissen führen kann, insbesondere dann, wenn einfache Anführungszeichen am Beginn von Kontraktionen verwendet werden.

--subset-fonts, -f

Eine "Schrift reduzieren" heißt, eine eingebettete Schrift auf die Buchstaben zu reduzieren, welche im Buch tatsächlich verwendet werden. Diese Option vermindert die Größe der Schriftartendatei enorm (die Dateigröße zu halbieren ist fast immer möglich). Wenn ein Buch zum Beispiel eine spezielle Schrift für Überschriften benutzt, dann wird "Schrift reduzieren" nur die Buchstaben in die eingebettete Schrift übernehmen, die tatsächlich in den Überschriften verwendet werden. Genau so wird, wenn die Schrift auch kursive oder fette Buchstaben enthält, diese aber selten oder gar nicht verwendet werden, diese Schrift auf die wirklich benutzten Zeichen reduziert oder sogar komplett entfernt werden. Der einzige Nachteil von "Schrift reduzieren" ist, dass ein nachträglich hinzugefügter Text möglicherweise nicht ganz durch die reduzierte Schrift abgedeckt wird.

--upgrade-book, -U

Interne Buchstrukturen upgraden, falls möglich. Wandelt beispielsweise EPUB-2-Bücher in EPUB-3-Bücher um.

--verbose

Umfangreichere Ausgabe erzeugen. Nützlich für die Fehlerdiagnose.

--version

Programmversion anzeigen und beenden

13.1.11 ebook-viewer

```
ebook-viewer [Optionen] Datei
```

Anzeigen eines eBooks.

Wann immer Sie Argumente mit Leerzeichen an **ebook-viewer** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--continue

Zuletzt geöffnetes Buch weiterlesen

--detach

Calibre vom Terminalfenster ablösen, falls vorhanden (nur für Linux)

--force-reload

Force reload of all opened books

--full-screen, --fullscreen, -f

If specified, the E-book viewer window will try to open full screen when started.

--help, -h

Diesen Hilfetext anzeigen und beenden.

--new-instance

Open a new viewer window even when the option to use only a single viewer window is set

--open-at

The position at which to open the specified book. The position is a location or position you can get by using the Go to->Location action in the viewer controls. Alternately, you can use the form toc:something and it will open at the location of the first Table of Contents entry that contains the string "something". The form toc-href:something will match the href (internal link destination) of toc nodes. The matching is exact. If you want to match a substring, use the form toc-href-contains:something. The form ref:something will use Reference mode references. The form search:something will search for something after opening the book. The form regex:something will search for the regular expression something after opening the book.

--raise-window

If specified, the E-book viewer window will try to come to the front when started.

--version

Programmversion anzeigen und beenden

13.1.12 fetch-ebook-metadata

```
fetch-ebook-metadata [Optionen]
```

Abrufen von Buchmetadaten aus Onlinequellen. Sie müssen zumindest Titel, Autor oder ISBN angeben.

Wann immer Sie Argumente mit Leerzeichen an **fetch-ebook-metadata** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--allowed-plugin, -p

Geben Sie den Namen einer Erweiterung zum Herunterladen von Metadaten an. Standardmäßig werden alle Metadaten-Erweiterungen verwendet. Kann für mehrere Erweiterungen mehrfach angegeben werden. Alle Erweiterungsnamen: Google, Google Images, Amazon.com, Edelweiss, Open Library, Big Book Search

--authors, -a

Buchautor(en)

--cover, -c

Geben Sie einen Dateinamen an. Das Titelbild wird, falls verfügbar, darunter gespeichert werden. Ohne diese Option wird kein Titelbild heruntergeladen.

--help, -h

Diesen Hilfetext anzeigen und beenden.

--identifier, -I

Identifiers such as ASIN/Goodreads id etc. Can be specified multiple times for multiple identifiers. For example: *--identifier* (die Titelseite 340) asin:B0082BAJA0

--isbn, -i

Buch-ISBN

--opf, -o

Metadaten im OPF-Format ausgeben anstatt in lesbarem Klartext.

--timeout, -d

Zeitüberschreitung in Sekunden. Standardeinstellung ist 30

--title, -t

Buchtitel

--verbose, -v

Protokoll in der Konsole (stderr) ausgeben

--version

Programmversion anzeigen und beenden

13.1.13 lrf2lrs

```
lrf2lrs Buch.lrf
```

Konvertieren einer LRF-Datei in eine LRS- (XML-UTF-8-codierte) Datei

Wann immer Sie Argumente mit Leerzeichen an **lrf2lrs** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--dont-output-resources

Eingebettete Bilder und Schriftartdateien nicht auf Datenträger speichern

--help, -h

Diesen Hilfetext anzeigen und beenden.

--output, -o
Ausgabe-LRS-Datei

--verbose
Mehr Informationen anzeigen

--version
Programmversion anzeigen und beenden

13.1.14 lrfviewer

```
lrfviewer [Optionen] Buch.lrf
```

Öffnen des LRF-eBooks Buch.lrf

Wann immer Sie Argumente mit Leerzeichen an **lrfviewer** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--disable-hyphenation
Silbentrennung deaktivieren. Das Rendering sollte deutlich beschleunigt werden.

--help, -h
Diesen Hilfetext anzeigen und beenden.

--profile
LRF-Renderer profilieren

--verbose
Mehr Informationen über den Renderingprozess ausgeben

--version
Programmversion anzeigen und beenden

--visual-debug
Visuelle Hilfen für die Fehlerdiagnose der Rendering-Engine aktivieren.

--white-background
Standardmäßig ist der Hintergrund naturweiß, weil ich dies angenehmer für die Augen finde. Verwenden Sie diese Option, um den Hintergrund auf reinweiß einzustellen.

13.1.15 lrs2lrf

```
lrs2lrf [Optionen] Datei.lrs
```

Erzeugen einer LRF-Datei aus einer LRS-Datei.

Wann immer Sie Argumente mit Leerzeichen an **lrs2lrf** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--help, -h
Diesen Hilfetext anzeigen und beenden.

--lrs

LRS zu LRS konvertieren, hilfreich für die Fehlerdiagnose.

--output, -o

Pfad zur Ausgabedatei

--verbose

Ausführliche Verarbeitung

--version

Programmversion anzeigen und beenden

13.1.16 web2disk

```
web2disk URL
```

wobei URL z. B. https://google.com ist

Wann immer Sie Argumente mit Leerzeichen an **web2disk** weitergeben, müssen diese Argumente in Anführungsstriche gesetzt werden. Zum Beispiel: „/some path/with spaces“

[Optionen]

--base-dir, -d

Base folder into which URL is saved. Default is .

--delay

Kleinstes Intervall in Sekunden zwischen aufeinander folgenden Abrufen. Standardeinstellung ist 0 s

--dont-download-stylesheets

CSS-Stylesheets nicht herunterladen.

--encoding

Die Zeichencodierung für Websites, die Sie zu laden versuchen. In der Voreinstellung wird versucht, die Codierung zu ermitteln.

--filter-regexp

Jeder Verweis, der durch diesen regulären Ausdruck beschrieben wird, wird ignoriert. Diese Option kann mehrfach angegeben werden, dann wird jeder Verweis, der durch irgendeinen der regulären Ausdrücke beschrieben wird, ignoriert. Standardmäßig wird kein Verweis ignoriert. Werden sowohl Filter- als auch Suchausdrücke verwendet, werden die Filterausdrücke zuerst angewendet.

--help, -h

Diesen Hilfetext anzeigen und beenden.

--match-regexp

Nur Verweise, die diesem regulären Ausdruck entsprechen, werden verfolgt. Diese Option kann mehrmals angegeben werden, somit werden Verweise verfolgt, solange sie einem regulären Ausdruck entsprechen. In der Voreinstellung werden alle Verweise verfolgt.

--max-files, -n

Maximale Anzahl herunterzuladender Dateien. Dies betrifft nur auf Dateien aus <a href>-Schlagwörtern. Standardeinstellung ist 9223372036854775807

--max-recursions, -r

Maximale Anzahl von einbezogenen Ebenen, z. B. Tiefe der Verweise, die verfolgt werden. Standardeinstellung 1

--timeout, -t

Zeitüberschreitung in Sekunden beim Warten auf eine Antwort des Servers. Standardeinstellung: 10.0 s

--verbose

Detaillierte Ausgabeinformation anzeigen. Hilfreich zur Fehlerdiagnose

--version

Programmversion anzeigen und beenden

13.2 Undokumentierte Befehle

- ebook-device
- markdown-calibre

Sie können die Verwendung von undokumentierten Befehlen durch Ausführen der Befehle ohne Angabe von Argumenten in einem Terminalfenster sehen.

KAPITEL 14

Einrichten einer Calibre-Entwicklungsumgebung

calibre is completely open source, licensed under the GNU GPL v3[118]. 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[119] with some C/C++ code for speed and system interfacing. Note that calibre requires at least Python 3.8.

Inhalt

- *Designphilosophie* (die Titelseite 346)
  - *Code-Layout* (die Titelseite 346)
- *Beziehen des Codes* (die Titelseite 347)
  - *Submitting your changes to be included* (die Titelseite 347)
- *Windows-Entwicklungsumgebung* (die Titelseite 348)
- *macOS Entwicklungsumgebung* (die Titelseite 349)
- *Linux-Entwicklungsumgebung* (die Titelseite 350)
- *Verwenden separater „Normal"- und „Entwicklungs"-Installationen von Calibre auf demselben Computer* (die Titelseite 350)
- *Tipps zur Fehlerdiagnose* (die Titelseite 351)
  - *Using print statements* (die Titelseite 351)
  - *Using an interactive Python interpreter* (die Titelseite 351)
  - *Using the Python debugger as a remote debugger* (die Titelseite 351)
  - *Using the debugger in your favorite Python IDE* (die Titelseite 352)

[118] https://www.gnu.org/licenses/gpl.html
[119] https://www.python.org

  - *Executing arbitrary scripts in the calibre Python environment* (die Titelseite 352)
- *Verwenden von Calibre in Ihren Projekten* (die Titelseite 352)
  - *Binäre Installation von Calibre* (die Titelseite 353)
  - *Source install on Linux* (die Titelseite 353)
- *API-Dokumentation für diverse Teile von Calibre* (die Titelseite 353)

14.1 Designphilosophie

Calibre hat seine Wurzeln in der Welt von Unix, was bedeutet, dass es sehr modular aufgebaut ist. Die verschiedenen Module interagieren miteinander über klar definierte Schnittstellen. Das macht es einfach, neue Funktionen hinzuzufügen oder Fehler zu beheben, woraus das hohe Entwicklungstempo von Calibre resultiert. Auf Grund dieser Wurzeln besitzt Calibre eine umfassende Befehlszeilenschnittstelle für alle seiner Funktionen, dokumentiert in *Befehlszeilenschnittstelle* (die Titelseite 303).

The modular design of calibre is expressed via `Plugins`. There is a *tutorial* (die Titelseite 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[120]. Similarly, adding support for new conversion formats involves writing input/output format plugins. Another example of the modular design is the *recipe system* (die Titelseite 33) for fetching news. For more examples of plugins designed to add features to calibre, see the Index of plugins[121].

14.1.1 Code-Layout

All the calibre Python code is in the `calibre` package. This package contains the following main sub-packages

- devices - Alle Gerätetreiber. Schauen Sie sich einfach einige der integrierten Treiber an um eine Vorstellung für ihre Funktionsweise zu bekommen.
  - 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`.
  - Alles zu Metadaten lesen, schreiben und herunterladen befindet sich unter `ebooks.metadata`
  - Conversion happens in a pipeline, for the structure of the pipeline, see *Einführung* (die Titelseite 63). 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 *API documentation for the e-book editing tools* (die Titelseite 363).
- db - The database back-end. See *API documentation for the database interface* (die Titelseite 353) for the interface to the calibre library.

[120] https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices
[121] https://www.mobileread.com/forums/showthread.php?p=1362767#post1362767

- 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`.

If you want to locate the entry points for all the various calibre executables, look at the `entry_points` structure in linux.py[122].

If you need help understanding the code, post in the development forum[123] and you will most likely get help from one of calibre's many developers.

14.2 Beziehen des Codes

You can get the calibre source code in two ways, using a version control system or directly downloading a tarball[124].

calibre uses Git[125], 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[126].

Um einen Zweig auf den neusten Stand des Codes zu aktualisieren, verwenden Sie den Befehl:

```
git pull --no-edit
```

You can also browse the code at GitHub[127].

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[128]. 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[129]. 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
```

Um die letzten *n* Commits zu senden, ersetzen Sie *1* mit *n*, zum Beispiel für die letzten 3 Commits:

[122] https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.py
[123] https://www.mobileread.com/forums/forumdisplay.php?f=240
[124] https://calibre-ebook.com/dist/src
[125] https://www.git-scm.com/
[126] https://calibre-ebook.com/dist/src
[127] https://github.com/kovidgoyal/calibre
[128] https://bugs.launchpad.net/calibre
[129] https://bugs.launchpad.net/calibre

```
git format-patch HEAD~3 --stdout > my-changes
```

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[130] 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[131]
- Setup ssh keys for authentication to GitHub, as described here: Generating SSH keys[132]
- Gehen Sie auf https://github.com/kovidgoyal/calibre und klicken Sie auf die Schaltfläche *Fork*.
- Führen Sie in einem Terminal aus:

  ```
  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[133]. 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-Entwicklungsumgebung

Bemerkung

You must also get the calibre source code separately as described above.

Install calibre normally, using the Windows installer[134]. 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[135] to setting environment variables on Windows.

[130] https://github.com
[131] https://help.github.com/articles/set-up-git
[132] https://help.github.com/articles/generating-ssh-keys
[133] https://www.mobileread.com/forums/forumdisplay.php?f=240
[134] https://calibre-ebook.com/download_windows
[135] https://docs.python.org/using/windows.html#excursus-setting-environment-variables

Once you have set the environment variable, open a new command prompt 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!`.

You can also setup a calibre development environment inside the free Microsoft Visual Studio, if you like, following the instructions here[136].

14.4 macOS Entwicklungsumgebung

> **Bemerkung**
>
> You must also get the calibre source code separately as described above.

Install calibre normally using the provided .dmg[137]. 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.

Erstellen Sie eine Klartextdatei:

```
#!/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
```

Once you have done this, run:

```
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.

[136] https://www.mobileread.com/forums/showthread.php?t=251201
[137] https://calibre-ebook.com/download_osx

14.5 Linux-Entwicklungsumgebung

> **Bemerkung**
>
> 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[138]. 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.

> **Bemerkung**
>
> 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. You can get these by running `calibre-debug --paths`. But note that distribution calibre packages are often terminally broken and are completely unsupported.

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 Verwenden separater „Normal"- und „Entwicklungs"-Installationen von Calibre auf demselben 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)

[138] https://calibre-ebook.com/download_linux

To launch the release version of calibre with your everyday library:

calibre-normal.bat:

```
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 Tipps zur Fehlerdiagnose

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[139].

Bemerkung

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)`.

Bemerkung

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.

14.8 Verwenden von Calibre in Ihren Projekten

It is possible to directly use calibre functions/code in your Python project. Two ways exist to do this:

[139] https://docs.python.org/library/pdb.html#debugger-commands

14.8.1 Binäre Installation von 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-Dokumentation für diverse Teile von Calibre

14.9.1 API documentation for the database interface

Die API ist thread sicher (sie benutzt ein „multiple reader, single writer“ locking Schema). Sie können diese API so benützen:

```
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

Wenn Sie in einer Calibre-Erweiterung sind, die Teil der Calibre-Hauptbenutzeroberfläche ist, bekommen Sie so Zugriff darauf:

```
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 wird benutzt um robust von der metadata.db zu lesen und in sie zu schreiben. Die ganze Tabellen les-/sortier-/such-/cachelogik wurde neu implementiert. Dies war notwendig um maximale Performance und Flexibilität zu erreichen.

class EventType(**values*)

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)

links_changed = 11

When the links associated with items(s) are changed, with arguments: (field_name, item_ids)

metadata_changed = 1

When some metadata is changed for some books, with arguments: (name of changed field, set of affected book ids)

notes_changed = 10

When the notes associated with item(s) are changed, with arguments: (field_name, item_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'}`.

Liefert ein Paar von lists:code:*ids, duplicates*. `ids` enthält die Bücher IDs für alle neu erstellten Bücher in der Datenbank. `` `duplicates `` enthält die `(mi, format_map)` für alle Bücher die bereits in der Datenbank exisitieren laut der simplen Duplikatdetektionsheuristik welche von *has_book()* (die Titelseite 359). verwendet wird.

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.

Parameter

- **replace** – Wenn True ersetze das existierende Format, andernfalls, wenn das Format bereits existiert, liefere False zurück.
- **run_hooks** – Falls *True* werden Dateityperweiterungen ausgeführt bevor und nachdem ein Format hinzugefügt wurde.
- **dbapi** – Nur zur internen Verwendung.

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* (die Titelseite 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_annotation_types()

Return a tuple of all annotation types in the database.

all_annotation_users()

Return a tuple of all (user_type, user name) that have annotations.

all_annotations(*restrict_to_user=None*, *limit=None*, *annotation_type=None*, *ignore_removed=False*, *restrict_to_book_ids=None*)

Return a tuple of all annotations matching the specified criteria. *ignore_removed* controls whether removed (deleted) annotations are also returned. Removed annotations are just a skeleton used for merging of annotations.

all_annotations_for_book(*book_id*)

Return a tuple containing all annotations for the specified book_id as a dict with keys: *format*, *user_type*, *user*, *annotation*. Here, annotation is the annotation data.

all_book_ids(*type=<class 'frozenset'>*)

Statisches Set von IDs für alle Werte im Feld „Name“

all_field_for(*field*, *book_ids*, *default_value=None*)

Gleich wie field_for, ausser dass es auf mehreren Büchern zugleich operiert.

all_field_ids(*name*)

Eingefrorene ID-Sets für alle Werte im Feld `name`.

all_field_names(*field*)

Eingefrorene Sets aller Feldnamen (sollte nur für die Felder „many-one“ und „many-many“ verwendet werden)

annotation_count_for_book(*book_id*)

Return the number of annotations for the specified book available in the database.

annotations_map_for_book(*book_id*, *fmt*, *user_type='local'*, *user='viewer'*)

Return a map of annotation type -> annotation data for the specified book_id, format, user and user_type.

author_data(*author_ids=None*)

Liefere die Autorendaten als ein dictionary mit den Schlüsseln: name, sort, link

Wenn keine Autoren mit den spezifizierten IDs gefunden wurde, wird ein leeres dictionary zurückgeliefert. Wenn author_ids None ist, werden die Daten für alle Autoren zurückgeliefert.

author_sort_from_authors(*authors*, *key_func=<function make_change_case_func.<locals>.change_case>*)

Gibt die Zeichenkette author_sort für die Autoren einer gegebenen Liste von Autoren zurück, wobei die dem Autor zugehörige Autorensortierung gegenüber der berechneten Zeichenkette bevorzugt wird.

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`.

Der zurückgelieferte Wert ist ein Set von Bücher IDs, oder ein leeres Set wenn das Element oder das Feld nicht existiert.

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.

Parameter

- **as_file** – If True return the image as an open file object (a SpooledTemporaryFile)
- **as_image** – If True return the image as a QImage object
- **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()* (die Titelseite 357) 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()* (die Titelseite 359). This can be used for an implementation of *has_book()* (die Titelseite 359) in a worker process without access to the db.

delete_annotations(*annot_ids*)

Delete annotations with the specified ids.

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()* (die Titelseite 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.

Parameter

- **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()* (die Titelseite 356) 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()* (die Titelseite 356).

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()* (die Titelseite 356) method.

Parameter

- **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 {}.

Parameter
book_id – the book id in question.

Rückgabe
{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*, *uncollapsed_categories=None*)

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*, *case_sensitive=False*)

Return the item id for item_name or None if not found. This function is very slow if doing lookups for multiple names use either get_item_ids() or get_item_name_map(). Similarly, case sensitive lookups are faster than case insensitive ones.

get_item_ids(*field*, *item_names*, *case_sensitive=False*)

Return a dict mapping item_name to the item id or None

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()* (die Titelseite 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.

Parameter
for_field – the lookup name of the field for which the link map is desired

Rückgabe
{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* (die Titelseite 210) 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.

Parameter

- **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()* (die Titelseite 359) 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()* (die Titelseite 356).

has_format(*book_id*, *fmt*)

Return True iff the format exists on disk

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.

Parameter

- **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.

Rückgabe

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_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

Merge the specified annotations into the existing annotations for book_id, fm, user_type, and user.

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*)

Buch aus Papierkorb wiederherstellen

move_format_from_trash(*book_id*, *fmt*)

Format aus Papierkorb wiederherstellen

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_extra_files(*book_id: int*, *relpaths: Iterable[str]*, *permanent=False*) → dict[str, Exception | None]

Delete the specified extra files, either to Recycle Bin or permanently.

remove_formats(*formats_map*, *db_only=False*)

Remove the specified formats from the specified books.

Parameter

- **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.

Rückgabe
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.

Parameter

- **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.

Parameter

- **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_annotations(*fts_engine_query*, *use_stemming=True*, *highlight_start=None*, *highlight_end=None*, *snippet_size=None*, *annotation_type=None*, *restrict_to_book_ids=None*, *restrict_to_user=None*, *ignore_removed=False*)

Return of a tuple of annotations matching the specified Full-text query.

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_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

Set all annotations for the specified book_id, fmt, user_type and user.

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.

Parameter

- **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** – Used internally, you should never change it.

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

Parameter

- **field** – the lookup name
- **value_to_link_map** – dict(field_value:link, …). Note that these are values, not field ids.

Rückgabe

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()* (die Titelseite 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.

Parameter

- **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

update_annotations(*annot_id_map*)

Anmerkungen aktualisieren

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 API documentation for the e-book editing tools

The e-book editing tools consist of a *calibre.ebooks.oeb.polish.container.Container* (die Titelseite 364) 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. Sie erhalten ein Container Objekt für ein Buch mit einem Pfad wie diesem:

```
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
```

Das Containerobjekt

class calibre.ebooks.oeb.polish.container.**Container**(*rootpath=None*, *opfpath=None*, *log=<calibre.utils.logging.Log object>*, *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()* (die Titelseite 366) or *open()* (die Titelseite 366) 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*)

Einen absoluten Pfad in ein canonical name relative zu `root` konvertieren

Parameter

root – The base folder. By default the root for this container object is used.

add_file(*name*, *data=b''*, *media_type=None*, *spine_index=None*, *modify_name_if_needed=False*, *process_manifest_item=None*, *suggested_id=''*)

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*, *suggested_id=''*)

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'

Der Typ des Buches (epub für EPUB-Dateien und azw3 für AZW3-Dateien)

commit(*outpath=None*, *keep_parsed=False*)

Commit all dirtied parsed objects to the filesystem and write out the e-book file at outpath.

Parameter

- **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()* (die Titelseite 366)

dirty(*name*)

Mark the parsed object corresponding to name as dirty. See also: *parsed()* (die Titelseite 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()* (die Titelseite 365)

filesize(*name*)

Return the size in bytes of the file represented by the specified canonical name. Automatically handles dirtied parsed objects. See also: *parsed()* (die Titelseite 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()* (die Titelseite 365) 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

Die geparste OPF-Datei

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

Die Version die im OPF-<package>-Element festgelegt wurde

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()* (die Titelseite 364) so that the container knows to update the cache. See also *replace()* (die Titelseite 367).

raw_data(*name*, *decode=True*, *normalize_to_nfc=True*)

Return the raw data corresponding to the file specified by name

Parameter

- **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()* (die Titelseite 364) 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()* (die Titelseite 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

Ein Iterator, der den Pfad für jeden Eintrag im Spine des Buchs liefert. Siehe auch: *spine_iter* (die Titelseite 367) and *spine_items* (die Titelseite 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* (die Titelseite 367) and *spine_items* (die Titelseite 367).

property spine_names

An iterator yielding name and is_linear for every item in the books' spine. See also: *spine_iter* (die Titelseite 367) and *spine_items* (die Titelseite 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.

Parameter

- **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** – If False, links are not replaced in the OPF file.

calibre.ebooks.oeb.polish.replace.**rename_files**(*container*, *file_map*)

Rename files in the container, automatically updating all links to them.

Parameter

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.

Aufteilen und Zusammenführen von Dateien

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.

Parameter

- **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** – Intern verwendet

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()* (die Titelseite 368). Splitting automatically migrates all links and references to the affected files.

Parameter

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.

Parameter

- **category** – Must be either `'text'` for HTML files or `'styles'` for CSS files
- **names** – Die Liste der zu zusammenführenden Dateien
- **master** – Which of the merged files is the *master* file, that is, the file that will remain after merging.

Verwalten von Titelbildern

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.

Parameter

- **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.

Parameter

move_to_start – If True the HTML file is moved to the start of the spine

Arbeiten mit 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.

Parameter

- **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.

Parameter

- **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.

Arbeiten mit dem Inhaltsverzeichnis

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.

Parameter

title – The title for this table of contents.

Buchbearbeitungswerkzeug

class calibre.gui2.tweak_book.plugin.**Tool**

Bases: object

The base class for individual tools in an Edit Book plugin. Useful members include:

- self.plugin: A reference to the *calibre.customize.Plugin* (die Titelseite 256) object to which this tool belongs.
- self. *boss* (die Titelseite 371)
- self. *gui* (die Titelseite 371)

Methoden die in Unterklassen überschrieben werden müssen:

- *create_action()* (die Titelseite 372)
- *register_shortcut()* (die Titelseite 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* (die Titelseite 372) object. Used to control the user interface.

property gui

Das Hauptfenster der Benutzeroberfläche

property current_container

Return the current *calibre.ebooks.oeb.polish.container.Container* (die Titelseite 364) 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.

Parameter

- **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')

        # 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
```

Siehe auch

Method *register_shortcut()* (die Titelseite 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.

Parameter

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

Parameter

- **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.

Parameter

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()

Speichern Sie das Buch. Der Speichervorgang wird im Hintergrund durchgeführt

set_modified()

Mark the book as having been modified

show_current_diff(*allow_revert=True*, *to_container=None*)

Show the changes to the book from its last checkpointed state

Parameter

- **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

KAPITEL 15

Digitale Rechteverwaltung (DRM)

Digital Rights Management (DRM) ist ein Oberbegriff für Zugangskontrolltechnologien, die von Hardware-Herstellern, Verlagen, Urheberrechtsinhabern und Einzelpersonen eingesetzt werden können, um die Nutzung digitaler Inhalte und Geräte einzuschränken. Manchmal wird es auch abschätzig als Digital Restrictions Management bezeichnet. Der Begriff wird verwendet, um jede Technologie zu beschreiben, die die (rechtmäßige oder anderweitige) Nutzung digitaler Inhalte verhindert, die vom Inhaltsanbieter nicht erwünscht oder vorhergesehen wurde. Der Begriff bezieht sich im Allgemeinen nicht auf andere Formen des Kopierschutzes, die umgangen werden können, ohne die Datei oder das Gerät zu verändern, wie Seriennummern oder Schlüsseldateien. Er kann sich auch auf Beschränkungen beziehen, die mit bestimmten Instanzen digitaler Werke oder Geräte verbunden sind. DRM-Technologien versuchen, die Nutzung digitaler Medien zu kontrollieren, indem sie den Zugriff, das Kopieren oder die Umwandlung in andere Formate durch die Endnutzer verhindern. Siehe Wikipedia[140].

15.1 Was impliziert DRM für mich persönlich?

Wenn du ein e-Book mit DRM kaufst, besitzt du es nicht tatsächlich, sondern zu hast eine Erlaubnis gekauft, es in der Weise zu benutzen, wie es vom Verkäufer bestimmt wird. DRM beschränkt, was du mit den e-Books tun kannst, die du „gekauft“ hast. Häufig sind sich Menschen, die Bücher mit DRM kaufen nicht über das Ausmaß der Beschränkungen im Klaren. Diese Einschränkungen hindern dich das e-Book nach deinem Geschmack zu reformatieren, einschließlich dem Anwenden von stylistischen Änderungen wie Anpassen der Schriftgröße. Obwohl es Software gibt, die dir dies für Nicht-DRM-Bücher ermöglicht. Die Menschen sind häufig überrascht, dass ein e-Book, dass sie in einem bestimmten Format gekauft haben, nicht in ein anderes Format konvertiert werden kann, wenn dieses e-Book mit DRM versehen ist. Wenn du also einen Amazon Kindle besitzt und ein Buch über Barnes and Nobles kaufst, solltest du wissen, dass, wenn dieses e-Book mit DRM versehen ist, es dir nicht möglich sein wird, es auf deinem Kindle zu lesen. Beachte, dass ich hier von einem Buch spreche, dass du gekauft hast - nicht gestohlen, nicht raubkopiert, sondern GEKAUFT hast.

15.2 Was bewirkt DRM für Autoren?

Publishers of DRMed e-books argue that the DRM is all for the sake of authors and to protect their artistic integrity and prevent piracy. But DRM does NOT prevent piracy. People who want to pirate content or use pirated content still do it and succeed. The three major DRM schemes for e-books today are run by Amazon, Adobe and Barnes and Noble and all

[140] https://en.wikipedia.org/wiki/Digital_rights_management

three DRM schemes have been cracked. All DRM does is inconvenience legitimate users. It can be argued that it actually harms authors as people who would have bought the book choose to find a pirated version as they are not willing to put up with DRM. Those that would pirate in the absence of DRM do so in its presence as well. To reiterate, the key point is that DRM *does not prevent piracy*. So DRM is not only pointless and harmful to buyers of e-books but also a waste of money.

15.3 DRM und Freiheit

Although digital content can be used to make information as well as creative works easily available to everyone and empower humanity, this is not in the interests of some publishers who want to steer people away from this possibility of freedom simply to maintain their relevance in world developing so fast that they can't keep up.

15.4 Weshalb unterstützt Calibre kein DRM?

calibre is open source software while DRM by its very nature is closed. If calibre were to support opening or viewing DRM files it could be trivially modified to be used as a tool for DRM removal which is illegal under today's laws. Open source software and DRM are a clash of principles. While DRM is all about controlling the user, open source software is about empowering the user. The two simply can not coexist.

15.5 Wie ist Calibres Ansicht zu Inhalte-Anbietern?

We firmly believe that authors and other content providers should be compensated for their efforts, but DRM is not the way to go about it. We are developing this database of DRM-free e-books from various sources to help you find DRM-free alternatives and to help independent authors and publishers of DRM-free e-books publicize their content. We hope you will find this useful and we request that you do not pirate the content made available to you here.

15.6 Wie kann ich helfen DRM zu bekämpfen?

As somebody who reads and buys e-books you can help fight DRM. Do not buy e-books with DRM. There are some publishers who publish DRM-free e-books. Make an effort to see if they carry the e-book you are looking for. If you like books by certain independent authors that sell DRM-free e-books and you can afford it make donations to them. This is money well spent as their e-books tend to be cheaper (there may be exceptions) than the ones you would buy from publishers of DRMed books and would probably work on all devices you own in the future saving you the cost of buying the e-book again. Do not discourage publishers and authors of DRM-free e-books by pirating their content. Content providers deserve compensation for their efforts. Do not punish them for trying to make your reading experience better by making available DRM-free e-books. In the long run this is detrimental to you. If you have bought books from sellers that carry both DRMed as well as DRM-free books, not knowing if they carry DRM or not make it a point to leave a comment or review on the website informing future buyers of its DRM status. Many sellers do not think it important to clearly indicate to their buyers if an e-book carries DRM or not. Here[141] you will find a guide to DRM-free living.

[141] https://www.defectivebydesign.org/guide/ebooks

KAPITEL 16

Glossar

RSS

RSS *(Really Simple Syndication)* ist ein Web-Feed-Format. Es wird verwendet, um sich häufig aktualisierende Inhalte, wie Nachrichten, Blog-Einträge, usw. zu veröffentlichen. Es ist ein Format, das sich besonders eignet, um von Computern gelesen werden und daher der bevorzugte Weg, um Inhalte aus dem Web in ein eBook zu bekommen. Im Internet sind viele andere Feed-Formate im Einsatz und Calibre versteht die meisten. Es hat besonders gute Unterstützung für das *ATOM*-Format, welches im Allgemeinen für Blogs verwendet wird.

Schema

Ein Schema (Recipe) ist ein Satz von Anweisungen, der Calibre zeigt, wie eine Online-Nachrichtenquelle, z. B. eine Zeitschrift oder ein Blog in ein eBook konvertiert wird. Ein Schema ist im Wesentlichen Python[142] Code. Als solcher kann es beliebig komplexe Nachrichtenquellen in eBooks umwandeln. Auf der einfachsten Ebene ist es nur ein Satz von Variablen, wie z. B. URLs, die Calibre genügend Informationen geben, um ins Internet zu gehen und die Nachrichten herunterzuladen.

HTML

HTML *(Hyper Text Mark-Up Language)*, eine Teilmenge der Standard Generalized Mark-Up Language (SGML) für elektronische Publikationen, ein spezifischer Standard, der im World Wide Web benutzt wird.

CSS

CSS *(Cascading Style Sheets, dt. Mehrstufige Formatvorlagen)* ist eine Sprache, die beschreibt, wie ein *HTML*-Dokument dargestellt werden soll (visuelle Gestaltung).

API

API *(Application Programming Interface)* ist eine offenen Anwendungsprogrammschnittstelle, welche für andere Programme Service-Bibliotheken mit benutzbaren Programmteilen bereitstellt.

LRF

LRF Das eBook-Format, das von den SONY-eBook-Readern gelesen wird.

URL

URL *(Uniform Resource Locator)* zum Beispiel: `http://beispiel.de`

Regexp

Regular expressions provide a concise and flexible means for identifying strings of text of interest, such as parti-

[142] https://www.python.org

cular characters, words, or patterns of characters. See *the tutorial* (die Titelseite 214) for an introduction to regular expressions.

Python-Modulindex

C

calibre.customize, 256
calibre.customize.conversion, 265
calibre.db.cache, 353
calibre.devices.interface, 268
calibre.ebooks.metadata.book.base, 210
calibre.ebooks.metadata.sources.base, 262
calibre.ebooks.oeb.polish.container, 363
calibre.ebooks.oeb.polish.cover, 369
calibre.ebooks.oeb.polish.css, 370
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, 184
calibre.web.feeds.news, 41

Sonderzeichen

-1
    calibredb-add command line option, 316
-H
    ebook-polish command line option, 337
-I
    calibredb-add command line option, 315
    fetch-ebook-metadata command line option, 340
-S
    calibredb-add command line option, 316
-T
    calibredb-add command line option, 316
-U
    ebook-polish command line option, 338
--access-log
    calibre-server command line option, 307
--add
    calibredb-add command line option, 316
--add-alt-text-to-img
    ebook-convert command line option, 332
--add-plugin
    calibre-customize command line option, 304
--add-simple-plugin
    calibre-debug command line option, 305
--add-soft-hyphens
    ebook-polish command line option, 337
--ajax-timeout
    calibre-server command line option, 307
--all
    calibredb-backup_metadata command line option, 323
    calibredb-export command line option, 318
--allowed-plugin
    fetch-ebook-metadata command line option, 340
--append
    calibredb-set_custom command line option, 322
--as-extra-data-file
    calibredb-add_format command line option, 316
--as-opf
    calibredb-show_metadata command line option, 317
--ascending
    calibredb-list command line option, 314
--asciiize
    ebook-convert command line option, 328
--attachment
    calibre-smtp command line option, 311
--auth-mode
    calibre-server command line option, 307
--author-sort
    ebook-convert command line option, 334
    ebook-meta command line option, 336
--authors
    calibredb-add command line option, 315
    ebook-convert command line option, 334
    ebook-meta command line option, 336
    fetch-ebook-metadata command line option, 340
--auto-reload
    calibre-server command line option, 307
--automerge
    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, 342
--base-font-size
    ebook-convert command line option, 328
--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, 311
--catalog-title
calibredb-catalog command line option, 319
--categories
calibredb-list_categories command line option, 323
--category
ebook-meta command line option, 336
--change-justification
ebook-convert command line option, 328
--chapter
ebook-convert command line option, 332
--chapter-mark
ebook-convert command line option, 332
--command
calibre-debug command line option, 305
--comments
ebook-convert command line option, 334
ebook-meta command line option, 336
--compress-images
ebook-polish command line option, 337
--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, 334
ebook-meta command line option, 336
ebook-polish command line option, 337
fetch-ebook-metadata command line option, 340
--cross-reference-authors
calibredb-catalog command line option, 319
--csv
calibredb-check_library command line option, 323
calibredb-list_categories command line option, 323
--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, 336
--debug-device-driver
calibre-debug command line option, 305
--debug-pipeline
calibredb-catalog command line option, 319
ebook-convert command line option, 335
--default-programs
calibre-debug command line option, 305
--delay
web2disk command line option, 342
--detach
calibre command line option, 304
ebook-edit command line option, 335
ebook-viewer command line option, 339
--details
calibredb-custom_columns command line option, 321
--dialect
calibredb-list_categories command line option, 323
--diff
calibre-debug command line option, 305
--disable-allow-socket-preallocation
calibre-server command line option, 308
--disable-auth
calibre-server command line option, 308
--disable-dehyphenate
ebook-convert command line option, 331
--disable-delete-blank-paragraphs
ebook-convert command line option, 331
--disable-fallback-to-detected-interface
calibre-server command line option, 308
--disable-fix-indents
ebook-convert command line option, 331
--disable-font-rescaling
ebook-convert command line option, 328
--disable-format-scene-breaks
ebook-convert command line option, 331
--disable-hyphenation
lrfviewer command line option, 341
--disable-italicize-common-cases
ebook-convert command line option, 331
--disable-local-write
calibre-server command line option, 308
--disable-log-not-found
calibre-server command line option, 308
--disable-markup-chapter-headings
ebook-convert command line option, 331
--disable-plugin
calibre-customize command line option, 304
--disable-remove-fake-margins
ebook-convert command line option, 332

--disable-renumber-headings
    ebook-convert command line option, 331
--disable-unwrap-lines
    ebook-convert command line option, 331
--disable-use-bonjour
    calibre-server command line option, 308
--disable-use-sendfile
    calibre-server command line option, 309
--disallow-rendered-cover
    ebook-meta command line option, 336
--display
    calibredb-add_custom_column command line option, 321
--displayed-fields
    calibre-server command line option, 308
--do-not-match-on-related-words
    calibredb-fts_search command line option, 325
--dont-asciiize
    calibredb-export command line option, 318
--dont-download-stylesheets
    web2disk command line option, 342
--dont-output-resources
    lrf2lrs command line option, 340
--dont-replace
    calibredb-add_format command line option, 317
--dont-save-cover
    calibredb-export command line option, 318
--dont-save-extra-files
    calibredb-export command line option, 318
--dont-split-on-page-breaks
    ebook-convert command line option, 327
--dont-update-metadata
    calibredb-export command line option, 318
--dont-verify-server-certificate
    calibre-smtp command line option, 312
--dont-write-opf
    calibredb-export command line option, 318
--download-external-resources
    ebook-polish command line option, 337
--duplicate-links-in-toc
    ebook-convert command line option, 333
--duplicates
    calibredb-add command line option, 315
--edit-book
    calibre-debug command line option, 305
--embed-all-fonts
    ebook-convert command line option, 328
--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, 331
--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, 342
--encryption-method
    calibre-smtp command line option, 312
--epub-flatten
    ebook-convert command line option, 327
--epub-inline-toc
    ebook-convert command line option, 327
--epub-max-image-size
    ebook-convert command line option, 327
--epub-toc-at-end
    ebook-convert command line option, 327
--epub-version
    ebook-convert command line option, 327
--exclude-genre
    calibredb-catalog command line option, 319
--exclusion-rules
    calibredb-catalog command line option, 319
--exec-file
    calibre-debug command line option, 305
--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, 329
--extract-to
    ebook-convert command line option, 327
--field

calibredb-set_metadata command line option, 317
--fields
calibredb-list command line option, 314
--filter-css
ebook-convert command line option, 329
--filter-regexp
web2disk command line option, 342
--flow-size
ebook-convert command line option, 327
--font-size-mapping
ebook-convert command line option, 329
--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, 318
--from-opf
ebook-convert command line option, 334
ebook-meta command line option, 336
--full-screen
ebook-viewer command line option, 339
--fullscreen
ebook-viewer command line option, 339
--generate-authors
calibredb-catalog command line option, 319
--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, 336
--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, 313
ebook-convert command line option, 326
ebook-edit command line option, 335
ebook-meta command line option, 336
ebook-polish command line option, 338
ebook-viewer command line option, 339
fetch-ebook-metadata command line option, 340
lrf2lrs command line option, 340
lrfviewer command line option, 341
lrs2lrf command line option, 341
web2disk command line option, 342
--html-unwrap-factor
ebook-convert command line option, 331
--identifier
calibredb-add command line option, 315
ebook-meta command line option, 336
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, 325

--index
    ebook-meta command line option, 336
--indexing-speed
    calibredb-fts_index command line option, 325
--indexing-threshold
    calibredb-fts_search command line option, 325
--input-encoding
    ebook-convert command line option, 327
--input-profile
    ebook-convert command line option, 326
--insert-blank-line
    ebook-convert command line option, 329
--insert-blank-line-size
    ebook-convert command line option, 329
--insert-metadata
    ebook-convert command line option, 332
--inspect-mobi
    calibre-debug command line option, 306
--is-multiple
    calibredb-add_custom_column command line option, 321
--isbn
    calibredb-add command line option, 315
    ebook-convert command line option, 334
    ebook-meta command line option, 336
    fetch-ebook-metadata command line option, 340
--item_count
    calibredb-list_categories command line option, 323
--jacket
    ebook-polish command line option, 338
--keep-ligatures
    ebook-convert command line option, 329
--kepubify
    calibre-debug command line option, 306
--language
    ebook-convert command line option, 334
    ebook-meta command line option, 336
--languages
    calibredb-add command line option, 315
--level1-toc
    ebook-convert command line option, 333
--level2-toc
    ebook-convert command line option, 333
--level3-toc
    ebook-convert command line option, 333
--library-path
    command line option, 313
--limit
    calibredb-list command line option, 314
    calibredb-search command line option, 324
--line-height
    ebook-convert command line option, 329
--line-width
    calibredb-list command line option, 314
--linearize-tables
    ebook-convert command line option, 329
--list-fields
    calibredb-set_metadata command line option, 317
--list-plugins
    calibre-customize command line option, 305
--list-recipes
    ebook-convert command line option, 326
--listen-on
    calibre-server command line option, 309
--localhost
    calibre-smtp command line option, 311
--log
    calibre-server command line option, 309
--lrf-bookid
    ebook-meta command line option, 336
--lrs
    lrs2lrf command line option, 341
--manage-users
    calibre-server command line option, 309
--margin-bottom
    ebook-convert command line option, 330
--margin-left
    ebook-convert command line option, 330
--margin-right
    ebook-convert command line option, 330
--margin-top
    ebook-convert command line option, 330
--match-end-marker
    calibredb-fts_search command line option, 325
--match-regexp
    web2disk command line option, 342
--match-start-marker
    calibredb-fts_search command line option, 325
--max-files
    web2disk command line option, 342
--max-header-line-size
    calibre-server command line option, 309
--max-job-time
    calibre-server command line option, 309
--max-jobs
    calibre-server command line option, 309
--max-log-size
    calibre-server command line option, 309
--max-opds-items
    calibre-server command line option, 309

--max-opds-ungrouped-items
    calibre-server command line option, 309
--max-recursions
    web2disk command line option, 342
--max-request-body-size
    calibre-server command line option, 310
--max-toc-links
    ebook-convert command line option, 333
--merge-comments-rule
    calibredb-catalog command line option, 320
--minimum-line-height
    ebook-convert command line option, 330
--new-instance
    ebook-viewer command line option, 339
--no-chapters-in-toc
    ebook-convert command line option, 333
--no-default-epub-cover
    ebook-convert command line option, 328
--no-svg-cover
    ebook-convert command line option, 328
--no-update-check
    calibre command line option, 304
--num-per-page
    calibre-server command line option, 310
--one-book-per-directory
    calibredb-add command line option, 316
--only-formats
    calibredb-embed_metadata command line option, 324
--open-at
    ebook-viewer command line option, 339
--opf
    ebook-polish command line option, 338
    fetch-ebook-metadata command line option, 340
--outbox
    calibre-smtp command line option, 311
--output
    lrf2lrs command line option, 340
    lrs2lrf command line option, 342
--output-format
    calibredb-fts_search command line option, 325
--output-profile
    calibredb-catalog command line option, 320
    ebook-convert command line option, 327
--page-breaks-before
    ebook-convert command line option, 332
--password
    calibre-smtp command line option, 312
    command line option, 314
--paths
    calibre-debug command line option, 306
--permanent
    calibredb-remove command line option, 316
--pidfile
    calibre-server command line option, 310
--port
    calibre-server command line option, 310
    calibre-smtp command line option, 312
--prefer-metadata-cover
    ebook-convert command line option, 333
--prefix
    calibredb-list command line option, 314
--prefix-rules
    calibredb-catalog command line option, 320
--preserve-cover-aspect-ratio
    ebook-convert command line option, 328
--preset
    calibredb-catalog command line option, 320
--pretty-print
    ebook-convert command line option, 328
--profile
    lrfviewer command line option, 341
--progress
    calibredb-export command line option, 318
--pubdate
    ebook-convert command line option, 334
--publisher
    ebook-convert command line option, 334
    ebook-meta command line option, 336
--raise-window
    ebook-viewer command line option, 339
--rating
    ebook-convert command line option, 334
    ebook-meta command line option, 337
--read-metadata-from-opf
    ebook-convert command line option, 334
--really-do-it
    calibredb-restore_database command line option, 322
--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, 330
--remove-paragraph-spacing-indent-size
    ebook-convert command line option, 330
--remove-plugin

calibre-customize command line option, 305
--remove-soft-hyphens
ebook-polish command line option, 338
--remove-unused-css
ebook-polish command line option, 338
--replace-scene-breaks
ebook-convert command line option, 331
--replace-whitespace
calibredb-export command line option, 318
--report
calibredb-check_library command line option, 323
--restrict-to
calibredb-fts_search command line option, 325
--run-plugin
calibre-debug command line option, 306
--run-test
calibre-debug command line option, 306
--run-without-debug
calibre-debug command line option, 306
--search
calibredb-catalog command line option, 319
calibredb-list command line option, 314
--search-replace
ebook-convert command line option, 332
--search-the-net-urls
calibre-server command line option, 310
--select-text
ebook-edit command line option, 335
--separator
calibredb-list command line option, 314
--series
calibredb-add command line option, 315
ebook-convert command line option, 334
ebook-meta command line option, 337
--series-index
calibredb-add command line option, 316
ebook-convert command line option, 334
--shutdown-running-calibre
calibre command line option, 304
calibre-debug command line option, 306
--shutdown-timeout
calibre-server command line option, 310
--single-dir
calibredb-export command line option, 318
--smarten-punctuation
ebook-convert command line option, 330
ebook-polish command line option, 338
--sort-by
calibredb-list command line option, 314
--sr1-replace
ebook-convert command line option, 332
--sr1-search
ebook-convert command line option, 332
--sr2-replace
ebook-convert command line option, 332
--sr2-search
ebook-convert command line option, 332
--sr3-replace
ebook-convert command line option, 332
--sr3-search
ebook-convert command line option, 332
--ssl-certfile
calibre-server command line option, 310
--ssl-keyfile
calibre-server command line option, 310
--start-in-tray
calibre command line option, 304
--start-reading-at
ebook-convert command line option, 333
--subject
calibre-smtp command line option, 311
--subset-embedded-fonts
ebook-convert command line option, 330
--subset-font
calibre-debug command line option, 307
--subset-fonts
ebook-polish command line option, 338
--tags
calibredb-add command line option, 316
ebook-convert command line option, 334
ebook-meta command line option, 337
--template
calibredb-export command line option, 318
calibredb-list command line option, 315
--template_file
calibredb-list command line option, 315
--template_heading
calibredb-list command line option, 315
--test-build
calibre-debug command line option, 307
--thumb-width
calibredb-catalog command line option, 320
--timefmt
calibredb-export command line option, 318
--timeout
calibre-server command line option, 310
calibre-smtp command line option, 311
command line option, 314
fetch-ebook-metadata command line option, 340
web2disk command line option, 342
--timestamp
ebook-convert command line option, 334

--title
    calibredb-add command line option, 316
    ebook-convert command line option, 334
    ebook-meta command line option, 337
    fetch-ebook-metadata command line option, 340
--title-sort
    ebook-convert command line option, 334
    ebook-meta command line option, 337
--to-dir
    calibredb-export command line option, 318
--to-lowercase
    calibredb-export command line option, 319
--to-opf
    ebook-meta command line option, 337
--toc-filter
    ebook-convert command line option, 333
--toc-threshold
    ebook-convert command line option, 333
--toc-title
    ebook-convert command line option, 328
--transform-css-rules
    ebook-convert command line option, 330
--transform-html-rules
    ebook-convert command line option, 330
--trusted-ips
    calibre-server command line option, 310
--un-kepubify
    calibre-debug command line option, 307
--unsmarten-punctuation
    ebook-convert command line option, 331
--upgrade-book
    ebook-polish command line option, 338
--url-prefix
    calibre-server command line option, 310
--use-auto-toc
    ebook-convert command line option, 333
--use-existing-cover
    calibredb-catalog command line option, 320
--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, 338
    fetch-ebook-metadata command line option, 340
    lrf2lrs command line option, 341
    lrfviewer command line option, 341
    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, 335
    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, 341
    lrs2lrf command line option, 342
    web2disk command line option, 343
--viewer
    calibre-debug command line option, 307
--visual-debug
    lrfviewer command line option, 341
--wait-for-completion
    calibredb-fts_index command line option, 325
--white-background
    lrfviewer command line option, 341
--width
    calibredb-list_categories command line option, 323
--with-library
    calibre command line option, 304
    command line option, 313
--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, 322
    ebook-meta command line option, 336
    fetch-ebook-metadata command line option, 340
-b

calibre-customize command line option, 304
-c
calibre-debug command line option, 305
calibredb-add command line option, 315
calibredb-check_library command line option, 323
calibredb-list_categories command line option, 323
ebook-meta command line option, 336
ebook-polish command line option, 337
fetch-ebook-metadata command line option, 340
-d
calibre-debug command line option, 305
calibredb-add command line option, 315
calibredb-custom_columns command line option, 321
ebook-convert command line option, 335
ebook-meta command line option, 336
ebook-polish command line option, 337
fetch-ebook-metadata command line option, 340
web2disk command line option, 342
-e
calibre-debug command line option, 305
calibre-smtp command line option, 312
calibredb-add command line option, 315
calibredb-check_library command line option, 323
ebook-polish command line option, 338
-f
calibre-debug command line option, 307
calibre-smtp command line option, 311
calibredb-embed_metadata command line option, 324
calibredb-list command line option, 314
calibredb-remove_custom_column command line option, 322
calibredb-set_metadata command line option, 317
ebook-polish command line option, 338
ebook-viewer command line option, 339
-g
calibre-debug command line option, 306
-h
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, 313
ebook-convert command line option, 326
ebook-edit command line option, 335
ebook-meta command line option, 336
ebook-polish command line option, 338
ebook-viewer command line option, 339
fetch-ebook-metadata command line option, 340
lrf2lrs command line option, 340
lrfviewer command line option, 341
lrs2lrf command line option, 341
web2disk command line option, 342
-i
calibre-debug command line option, 306
calibredb-add command line option, 315
calibredb-catalog command line option, 319
calibredb-list_categories command line option, 323
ebook-meta command line option, 336
ebook-polish command line option, 337
fetch-ebook-metadata command line option, 340
-j
ebook-polish command line option, 338
-k
ebook-meta command line option, 336
-l
calibre-customize command line option, 305
calibre-smtp command line option, 311
calibredb-add command line option, 315
calibredb-search command line option, 324
calibredb-set_metadata command line option, 317
ebook-meta command line option, 336
-m
calibre-debug command line option, 306
calibredb-add command line option, 315
ebook-convert command line option, 334
-n
calibredb-check_library command line option, 323
web2disk command line option, 342
-o
calibre-smtp command line option, 311
ebook-polish command line option, 338
fetch-ebook-metadata command line option, 340
lrf2lrs command line option, 340
lrs2lrf command line option, 342
-p
calibre-smtp command line option, 312
ebook-meta command line option, 336
ebook-polish command line option, 338

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
calibredb-add command line option, 316
calibredb-check_library command line option, 323
calibredb-list_categories command line option, 323
calibredb-restore_database command line option, 322
ebook-meta command line option, 337
web2disk command line option, 342
-s
calibre command line option, 304
calibre-debug command line option, 306
calibre-smtp command line option, 311
calibredb-add command line option, 315
calibredb-catalog command line option, 319
calibredb-list command line option, 314
ebook-meta command line option, 337
-t
calibre-debug command line option, 306
calibre-smtp command line option, 311
calibredb-add command line option, 316
calibredb-list command line option, 315
ebook-meta command line option, 337
fetch-ebook-metadata command line option, 340
web2disk command line option, 342
-u
calibre-smtp command line option, 312
ebook-polish command line option, 338
-v
calibre command line option, 304
calibre-smtp command line option, 311
calibredb-catalog command line option, 319
ebook-convert command line option, 335
fetch-ebook-metadata command line option, 340
-w
calibre-debug command line option, 307
calibredb-list command line option, 314
calibredb-list_categories command line option, 323
-x
calibre-debug command line option, 306

A

abort_article() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
abort_recipe_processing() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
abspath_to_name() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
accept_drag_move_event() (*Methode von calibre.gui2.actions.InterfaceAction*), 283
accept_enter_event() (*Methode von calibre.gui2.actions.InterfaceAction*), 283
accepts_drops (*Attribut von calibre.gui2.actions.InterfaceAction*), 283
action_add_menu (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
action_menu_clone_qaction (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
action_shortcut_name (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
action_spec (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
action_type (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
add_annotation_to_library() (*Methode von calibre.devices.usbms.device.Device*), 279
add_book() (*Methode von calibre.devices.interface.BookList*), 276
add_books() (*Methode von calibre.db.cache.Cache*), 354
add_books_to_metadata() (*Klassenmethode von calibre.devices.interface.DevicePlugin*), 272
add_books_to_metadata() (*Methode von calibre.devices.usbms.driver.USBMS*), 281
add_custom_book_data() (*Methode von calibre.db.cache.Cache*), 354
add_extra_files() (*Methode von calibre.db.cache.Cache*), 354
add_file() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
add_format() (*Methode von calibre.db.cache.Cache*), 354
add_listener() (*Methode von calibre.db.cache.Cache*), 354
add_name_to_manifest() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
add_notes_resource() (*Methode von calibre.db.cache.Cache*), 354
add_or_replace_jacket() (*im Modul calibre.ebooks.oeb.polish.jacket*), 368
add_properties() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
add_savepoint() (*Methode von calib-*

re.gui2.tweak_book.boss.Boss), 372
add_toc_thumbnail() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
adeify_images() (*Klassenmethode von calibre.web.feeds.news.BasicNewsRecipe*), 41
all_annotation_types() (*Methode von calibre.db.cache.Cache*), 355
all_annotation_users() (*Methode von calibre.db.cache.Cache*), 355
all_annotations() (*Methode von calibre.db.cache.Cache*), 355
all_annotations_for_book() (*Methode von calibre.db.cache.Cache*), 355
all_book_ids() (*Methode von calibre.db.cache.Cache*), 355
all_field_for() (*Methode von calibre.db.cache.Cache*), 355
all_field_ids() (*Methode von calibre.db.cache.Cache*), 355
all_field_keys() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
all_field_names() (*Methode von calibre.db.cache.Cache*), 355
all_non_none_fields() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
allowed_in_menu (*Attribut von calibre.gui2.tweak_book.plugin.Tool*), 371
allowed_in_toolbar (*Attribut von calibre.gui2.tweak_book.plugin.Tool*), 371
annotation_count_for_book() (*Methode von calibre.db.cache.Cache*), 355
annotations_map_for_book() (*Methode von calibre.db.cache.Cache*), 355
API, **377**
apply_container_update_to_gui() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 372
apply_unique_properties() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
articles_are_obfuscated (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 46
ASK_TO_ALLOW_CONNECT (*Attribut von calibre.devices.interface.DevicePlugin*), 269
author (*Attribut von calibre.customize.InterfaceActionBase*), 285
author (*Attribut von calibre.customize.MetadataReaderPlugin*), 260
author (*Attribut von calibre.customize.MetadataWriterPlugin*), 261
author (*Attribut von calibre.customize.Plugin*), 257
author (*Attribut von calibre.customize.PreferencesPlugin*), 285
author (*Attribut von calibre.devices.usbms.driver.USBMS*), 279
author (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
author_data() (*Methode von calibre.db.cache.Cache*), 355
author_sort_from_authors() (*Methode von calibre.db.cache.Cache*), 355
auto_cleanup (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 46
auto_cleanup_keep (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 46
auto_repeat (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
auto_trim_covers (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262

B

BasicNewsRecipe (*Klasse in calibre.web.feeds.news*), 41
BCD (*Attribut von calibre.devices.interface.DevicePlugin*), 268
BCD (*Attribut von calibre.devices.usbms.device.Device*), 277
book_class (*Attribut von calibre.devices.usbms.driver.USBMS*), 280
book_created (*Attribut von calibre.db.cache.Cache.EventType*), 353
book_edited (*Attribut von calibre.db.cache.Cache.EventType*), 353
book_type (*Attribut von calibre.ebooks.oeb.polish.container.Container*), 364
BookList (*Klasse in calibre.devices.interface*), 275
booklist_class (*Attribut von calibre.devices.usbms.driver.USBMS*), 280
books() (*Methode von calibre.devices.interface.DevicePlugin*), 272
books() (*Methode von calibre.devices.usbms.driver.USBMS*), 280
books_for_field() (*Methode von calibre.db.cache.Cache*), 355
books_in_virtual_library() (*Methode von calibre.db.cache.Cache*), 355
books_removed (*Attribut von calibre.db.cache.Cache.EventType*), 353
boss (*calibre.gui2.tweak_book.plugin.Tool property*), 371
Boss (*Klasse in calibre.gui2.tweak_book.boss*), 372
browser_type (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 46
BuiltinAdd (*Klasse in calibre.utils.formatter_functions*), 185
BuiltinAnd (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinAnnotationCount (*Klasse in calibre.utils.formatter_functions*), 194

BuiltinApproximateFormats (*Klasse in calibre.utils.formatter_functions*), 194
BuiltinArguments (*Klasse in calibre.utils.formatter_functions*), 184
BuiltinAssign (*Klasse in calibre.utils.formatter_functions*), 184
BuiltinAuthorLinks (*Klasse in calibre.utils.formatter_functions*), 194
BuiltinAuthorSorts (*Klasse in calibre.utils.formatter_functions*), 195
BuiltinBookCount (*Klasse in calibre.utils.formatter_functions*), 187
BuiltinBooksize (*Klasse in calibre.utils.formatter_functions*), 195
BuiltinBookValues (*Klasse in calibre.utils.formatter_functions*), 188
BuiltinCapitalize (*Klasse in calibre.utils.formatter_functions*), 187
BuiltinCeiling (*Klasse in calibre.utils.formatter_functions*), 185
BuiltinCharacter (*Klasse in calibre.utils.formatter_functions*), 205
BuiltinCheckYesNo (*Klasse in calibre.utils.formatter_functions*), 205
BuiltinCmp (*Klasse in calibre.utils.formatter_functions*), 204
BuiltinConnectedDeviceName (*Klasse in calibre.utils.formatter_functions*), 195
BuiltinConnectedDeviceUUID (*Klasse in calibre.utils.formatter_functions*), 195
BuiltinContains (*Klasse in calibre.utils.formatter_functions*), 205
BuiltinCount (*Klasse in calibre.utils.formatter_functions*), 200
BuiltinCurrentLibraryName (*Klasse in calibre.utils.formatter_functions*), 195
BuiltinCurrentLibraryPath (*Klasse in calibre.utils.formatter_functions*), 195
BuiltinCurrentVirtualLibraryName (*Klasse in calibre.utils.formatter_functions*), 196
BuiltinDateArithmetic (*Klasse in calibre.utils.formatter_functions*), 190
BuiltinDaysBetween (*Klasse in calibre.utils.formatter_functions*), 191
BuiltinDivide (*Klasse in calibre.utils.formatter_functions*), 185
BuiltinEncodeForURL (*Klasse in calibre.utils.formatter_functions*), 208
BuiltinEval (*Klasse in calibre.utils.formatter_functions*), 204
BuiltinExtraFileModtime (*Klasse in calibre.utils.formatter_functions*), 188
BuiltinExtraFileNames (*Klasse in calibre.utils.formatter_functions*), 188
BuiltinExtraFileSize (*Klasse in calibre.utils.formatter_functions*), 189
BuiltinField (*Klasse in calibre.utils.formatter_functions*), 196
BuiltinFieldExists (*Klasse in calibre.utils.formatter_functions*), 206
BuiltinFieldListCount (*Klasse in calibre.utils.formatter_functions*), 200
BuiltinFinishFormatting (*Klasse in calibre.utils.formatter_functions*), 191
BuiltinFirstMatchingCmp (*Klasse in calibre.utils.formatter_functions*), 204
BuiltinFirstNonEmpty (*Klasse in calibre.utils.formatter_functions*), 198
BuiltinFloor (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinFormatDate (*Klasse in calibre.utils.formatter_functions*), 191
BuiltinFormatDateField (*Klasse in calibre.utils.formatter_functions*), 192
BuiltinFormatDuration (*Klasse in calibre.utils.formatter_functions*), 193
BuiltinFormatNumber (*Klasse in calibre.utils.formatter_functions*), 194
BuiltinFormatsModtimes (*Klasse in calibre.utils.formatter_functions*), 196
BuiltinFormatsPaths (*Klasse in calibre.utils.formatter_functions*), 197
BuiltinFormatsPathSegments (*Klasse in calibre.utils.formatter_functions*), 196
BuiltinFormatsSizes (*Klasse in calibre.utils.formatter_functions*), 197
BuiltinFractionalPart (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinGetLink (*Klasse in calibre.utils.formatter_functions*), 189
BuiltinGetNote (*Klasse in calibre.utils.formatter_functions*), 189
BuiltinGlobals (*Klasse in calibre.utils.formatter_functions*), 185
BuiltinHasCover (*Klasse in calibre.utils.formatter_functions*), 197
BuiltinHasExtraFiles (*Klasse in calibre.utils.formatter_functions*), 190
BuiltinHasNote (*Klasse in calibre.utils.formatter_functions*), 190
BuiltinHumanReadable (*Klasse in calibre.utils.formatter_functions*), 194
BuiltinIdentifierInList (*Klasse in calibre.utils.formatter_functions*), 199
BuiltinIfempty (*Klasse in calibre.utils.formatter_functions*), 206
BuiltinInList (*Klasse in calibre.utils.formatter_functions*), 199

BuiltinIsDarkMode (*Klasse in calibre.utils.formatter_functions*), 185
BuiltinIsMarked (*Klasse in calibre.utils.formatter_functions*), 197
BuiltinLanguageCodes (*Klasse in calibre.utils.formatter_functions*), 197
BuiltinLanguageStrings (*Klasse in calibre.utils.formatter_functions*), 197
BuiltinListCountMatching (*Klasse in calibre.utils.formatter_functions*), 200
BuiltinListDifference (*Klasse in calibre.utils.formatter_functions*), 201
BuiltinListEquals (*Klasse in calibre.utils.formatter_functions*), 201
BuiltinListIntersection (*Klasse in calibre.utils.formatter_functions*), 201
BuiltinListitem (*Klasse in calibre.utils.formatter_functions*), 199
BuiltinListJoin (*Klasse in calibre.utils.formatter_functions*), 201
BuiltinListRe (*Klasse in calibre.utils.formatter_functions*), 202
BuiltinListReGroup (*Klasse in calibre.utils.formatter_functions*), 202
BuiltinListRemoveDuplicates (*Klasse in calibre.utils.formatter_functions*), 202
BuiltinListSort (*Klasse in calibre.utils.formatter_functions*), 202
BuiltinListSplit (*Klasse in calibre.utils.formatter_functions*), 202
BuiltinListUnion (*Klasse in calibre.utils.formatter_functions*), 202
BuiltinLookup (*Klasse in calibre.utils.formatter_functions*), 199
BuiltinLowercase (*Klasse in calibre.utils.formatter_functions*), 187
BuiltinMakeUrl (*Klasse in calibre.utils.formatter_functions*), 208
BuiltinMakeUrlExtended (*Klasse in calibre.utils.formatter_functions*), 208
BuiltinMod (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinMultiply (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinNot (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinOndevice (*Klasse in calibre.utils.formatter_functions*), 198
BuiltinOr (*Klasse in calibre.utils.formatter_functions*), 187
BuiltinPrint (*Klasse in calibre.utils.formatter_functions*), 185
BuiltinQueryString (*Klasse in calibre.utils.formatter_functions*), 209
BuiltinRange (*Klasse in calibre.utils.formatter_functions*), 203
BuiltinRatingToStars (*Klasse in calibre.utils.formatter_functions*), 194
BuiltinRawField (*Klasse in calibre.utils.formatter_functions*), 198
BuiltinRawList (*Klasse in calibre.utils.formatter_functions*), 198
BuiltinRe (*Klasse in calibre.utils.formatter_functions*), 206
BuiltinReGroup (*Klasse in calibre.utils.formatter_functions*), 206
BuiltinRound (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinSelect (*Klasse in calibre.utils.formatter_functions*), 200
BuiltinSeriesSort (*Klasse in calibre.utils.formatter_functions*), 198
BuiltinSetGlobals (*Klasse in calibre.utils.formatter_functions*), 185
BuiltinShorten (*Klasse in calibre.utils.formatter_functions*), 206
BuiltinStrcat (*Klasse in calibre.utils.formatter_functions*), 206
BuiltinStrcatMax (*Klasse in calibre.utils.formatter_functions*), 207
BuiltinStrcmp (*Klasse in calibre.utils.formatter_functions*), 205
BuiltinStrcmpcase (*Klasse in calibre.utils.formatter_functions*), 205
BuiltinStrInList (*Klasse in calibre.utils.formatter_functions*), 200
BuiltinStrlen (*Klasse in calibre.utils.formatter_functions*), 207
BuiltinSubitems (*Klasse in calibre.utils.formatter_functions*), 203
BuiltinSublist (*Klasse in calibre.utils.formatter_functions*), 204
BuiltinSubstr (*Klasse in calibre.utils.formatter_functions*), 207
BuiltinSubtract (*Klasse in calibre.utils.formatter_functions*), 186
BuiltinSwapAroundArticles (*Klasse in calibre.utils.formatter_functions*), 207
BuiltinSwapAroundComma (*Klasse in calibre.utils.formatter_functions*), 207
BuiltinSwitch (*Klasse in calibre.utils.formatter_functions*), 199
BuiltinSwitchIf (*Klasse in calibre.utils.formatter_functions*), 199
BuiltinTemplate (*Klasse in calibre.utils.formatter_functions*), 204
BuiltinTest (*Klasse in calibre.utils.formatter_functions*), 207

BuiltinTitlecase (*Klasse in calibre.utils.formatter_functions*), 187
BuiltinToday (*Klasse in calibre.utils.formatter_functions*), 191
BuiltinToHex (*Klasse in calibre.utils.formatter_functions*), 210
BuiltinTransliterate (*Klasse in calibre.utils.formatter_functions*), 207
BuiltinUppercase (*Klasse in calibre.utils.formatter_functions*), 187
BuiltinUrlsFromIdentifiers (*Klasse in calibre.utils.formatter_functions*), 210
BuiltinUserCategories (*Klasse in calibre.utils.formatter_functions*), 198
BuiltinVirtualLibraries (*Klasse in calibre.utils.formatter_functions*), 198

C

Cache (*Klasse in calibre.db.cache*), 353
Cache.EventType (*Klasse in calibre.db.cache*), 353
cached_cover_url_is_reliable (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
calibre command line option
    --detach, 304
    --help, 304
    --ignore-plugins, 304
    --no-update-check, 304
    --shutdown-running-calibre, 304
    --start-in-tray, 304
    --verbose, 304
    --version, 304
    --with-library, 304
    -h, 304
    -s, 304
    -v, 304
calibre.customize
    module, 256
calibre.customize.conversion
    module, 265
calibre.db.cache
    module, 353
calibre.devices.interface
    module, 268
calibre.ebooks.metadata.book.base
    module, 210
calibre.ebooks.metadata.sources.base
    module, 262
calibre.ebooks.oeb.polish.container
    module, 363
calibre.ebooks.oeb.polish.cover
    module, 369
calibre.ebooks.oeb.polish.css
    module, 370
calibre.ebooks.oeb.polish.jacket
    module, 368
calibre.ebooks.oeb.polish.pretty
    module, 368
calibre.ebooks.oeb.polish.replace
    module, 367
calibre.ebooks.oeb.polish.split
    module, 368
calibre.ebooks.oeb.polish.toc
    module, 370
calibre.gui2.tweak_book.boss
    module, 372
calibre.utils.formatter_functions
    module, 184
calibre.web.feeds.news
    module, 41
calibre-customize command line option
    --add-plugin, 304
    --build-plugin, 304
    --customize-plugin, 304
    --disable-plugin, 304
    --enable-plugin, 304
    --help, 305
    --list-plugins, 305
    --remove-plugin, 305
    --version, 305
    -a, 304
    -b, 304
    -h, 305
    -l, 305
    -r, 305
calibre-debug command line option
    --add-simple-plugin, 305
    --command, 305
    --debug-device-driver, 305
    --default-programs, 305
    --diff, 305
    --edit-book, 305
    --exec-file, 305
    --explode-book, 306
    --export-all-calibre-data, 306
    --gui, 306
    --gui-debug, 306
    --help, 306
    --implode-book, 306
    --import-calibre-data, 306
    --inspect-mobi, 306
    --kepubify, 306
    --paths, 306
    --run-plugin, 306
    --run-test, 306
    --run-without-debug, 306
    --shutdown-running-calibre, 306
    --subset-font, 307
    --test-build, 307

--un-kepubify, 307
--version, 307
--viewer, 307
-c, 305
-d, 305
-e, 305
-f, 307
-g, 306
-h, 306
-i, 306
-m, 306
-r, 306
-s, 306
-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, 310
--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, 311
--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
calibredb-add command line option
-1, 316
-I, 315
-S, 316
-T, 316
--add, 316
--authors, 315
--automerge, 315

--cover, 315
--duplicates, 315
--empty, 315
--identifier, 315
--ignore, 316
--isbn, 315
--languages, 315
--one-book-per-directory, 316
--recurse, 316
--series, 315
--series-index, 316
--tags, 316
--title, 316
-a, 315
-c, 315
-d, 315
-e, 315
-i, 315
-l, 315
-m, 315
-r, 316
-s, 315
-t, 316
calibredb-add_custom_column command line option
--display, 321
--is-multiple, 321
calibredb-add_format command line option
--as-extra-data-file, 316
--dont-replace, 317
calibredb-backup_metadata command line option
--all, 323
calibredb-catalog command line option
--catalog-title, 319
--cross-reference-authors, 319
--debug-pipeline, 319
--exclude-genre, 319
--exclusion-rules, 319
--generate-authors, 319
--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, 320
--prefix-rules, 320
--preset, 320
--search, 319
--thumb-width, 320
--use-existing-cover, 320
--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, 321
-d, 321
calibredb-embed_metadata command line option
--only-formats, 324
-f, 324
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, 318
--progress, 318
--replace-whitespace, 318
--single-dir, 318
--template, 318
--timefmt, 318
--to-dir, 318
--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, 325
--include-snippets, 325
--indexing-threshold, 325
--match-end-marker, 325
--match-start-marker, 325
--output-format, 325
--restrict-to, 325
calibredb-list command line option
--ascending, 314
--fields, 314
--for-machine, 314
--limit, 314

--line-width, 314
--prefix, 314
--search, 314
--separator, 314
--sort-by, 314
--template, 315
--template_file, 315
--template_heading, 315
-f, 314
-s, 314
-t, 315
-w, 314
calibredb-list_categories command line option
--categories, 323
--csv, 323
--dialect, 323
--item_count, 323
--width, 323
-c, 323
-i, 323
-r, 323
-w, 323
calibredb-remove command line option
--permanent, 316
calibredb-remove_custom_column command line option
--force, 322
-f, 322
calibredb-restore_database command line option
--really-do-it, 322
-r, 322
calibredb-search command line option
--limit, 324
-l, 324
calibredb-set_custom command line option
--append, 322
-a, 322
calibredb-set_metadata command line option
--field, 317
--list-fields, 317
-f, 317
-l, 317
calibredb-show_metadata command line option
--as-opf, 317
can_be_disabled (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 265
can_be_disabled (*Attribut von calibre.customize.conversion.OutputFormatPlugin*), 267
can_be_disabled (*Attribut von calibre.customize.InterfaceActionBase*), 285
can_be_disabled (*Attribut von calibre.customize.Plugin*), 257
can_be_disabled (*Attribut von calibre.customize.PreferencesPlugin*), 285
CAN_DO_DEVICE_DB_PLUGBOARD (*Attribut von calibre.devices.interface.DevicePlugin*), 269
can_get_multiple_covers (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
can_handle() (*Methode von calibre.devices.interface.DevicePlugin*), 270
can_handle_windows() (*Methode von calibre.devices.interface.DevicePlugin*), 270
can_handle_windows() (*Methode von calibre.devices.usbms.device.Device*), 278
CAN_SET_METADATA (*Attribut von calibre.devices.interface.DevicePlugin*), 269
CAN_SET_METADATA (*Attribut von calibre.devices.usbms.driver.USBMS*), 280
canonicalize_internal_url() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
capabilities (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
card_prefix() (*Methode von calibre.devices.interface.DevicePlugin*), 271
card_prefix() (*Methode von calibre.devices.usbms.device.Device*), 278
CatalogPlugin (*Klasse in calibre.customize*), 261
category (*Attribut von calibre.customize.PreferencesPlugin*), 285
category_order (*Attribut von calibre.customize.PreferencesPlugin*), 285
center_navbar (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 46
change_font() (*im Modul calibre.ebooks.oeb.polish.fonts*), 370
changed_signal (*Attribut von calibre.gui2.preferences.ConfigWidgetBase*), 287
changed_signal (*Attribut von calibre.gui2.preferences.ConfigWidgetInterface*), 286
clean_downloaded_metadata() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
cleanup() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
CLI (*Klasse in calibre.devices.usbms.cli*), 279
cli_main() (*Methode von calibre.customize.Plugin*), 258
cli_options (*Attribut von calibre.customize.CatalogPlugin*), 261
clone_browser() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
close_editor() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 372
command line option
--help, 313

--library-path, 313
--password, 314
--timeout, 314
--username, 314
--version, 314
--with-library, 313
-h, 313
commit() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
commit() (*Methode von calibre.gui2.preferences.ConfigWidgetBase*), 287
commit() (*Methode von calibre.gui2.preferences.ConfigWidgetInterface*), 286
commit_all_editors_to_container() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 372
commit_item() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
common_options (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 266
common_options (*Attribut von calibre.customize.conversion.OutputFormatPlugin*), 267
compress_covers() (*Methode von calibre.db.cache.Cache*), 355
compress_news_images (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 46
compress_news_images_auto_size (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 46
compress_news_images_max_size (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
config_help_message (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
config_widget (*Attribut von calibre.customize.PreferencesPlugin*), 285
config_widget() (*Klassenmethode von calibre.devices.interface.DevicePlugin*), 273
config_widget() (*Methode von calibre.customize.Plugin*), 257
config_widget() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
ConfigWidgetBase (*Klasse in calibre.gui2.preferences*), 287
ConfigWidgetInterface (*Klasse in calibre.gui2.preferences*), 286
Container (*Klasse in calibre.ebooks.oeb.polish.container*), 364
contains(), **160**
conversion_options (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
convert() (*Methode von calibre.customize.conversion.InputFormatPlugin*), 266
convert() (*Methode von calibre.customize.conversion.OutputFormatPlugin*), 267
copy_cover_to() (*Methode von calibre.db.cache.Cache*), 356
copy_format_to() (*Methode von calibre.db.cache.Cache*), 356
core_usage (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 266
cover() (*Methode von calibre.db.cache.Cache*), 356
cover_margins (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
create_action() (*Methode von calibre.gui2.tweak_book.plugin.Tool*), 372
create_inline_toc() (*im Modul calibre.ebooks.oeb.polish.toc*), 370
create_menu_action() (*Methode von calibre.gui2.actions.InterfaceAction*), 283
create_widget() (*Methode von calibre.customize.PreferencesPlugin*), 286
CSS, **377**
current_container (*calibre.gui2.tweak_book.plugin.Tool property*), 371
currently_editing (*calibre.gui2.tweak_book.boss.Boss property*), 372
custom_field_keys() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
customization_help() (*Methode von calibre.customize.Plugin*), 258
customization_help() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263

D

data_for_find_identical_books() (*Methode von calibre.db.cache.Cache*), 356
data_for_has_book() (*Methode von calibre.db.cache.Cache*), 356
debug_managed_device_detection() (*Methode von calibre.devices.interface.DevicePlugin*), 270
deepcopy() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 210
default_cover() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
delay (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
delete_annotations() (*Methode von calibre.db.cache.Cache*), 356

delete_books() (*Methode von calibre.devices.interface.DevicePlugin*), 273
delete_books() (*Methode von calibre.devices.usbms.driver.USBMS*), 281
delete_custom_book_data() (*Methode von calibre.db.cache.Cache*), 356
delete_trash_entry() (*Methode von calibre.db.cache.Cache*), 356
description (*Attribut von calibre.customize.Plugin*), 256
description (*Attribut von calibre.customize.PreferencesPlugin*), 286
description (*Attribut von calibre.devices.usbms.driver.USBMS*), 279
description (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
description (*calibre.customize.conversion.OutputFormatPlugin property*), 267
detect_managed_devices() (*Methode von calibre.devices.interface.DevicePlugin*), 270
Device (*Klasse in calibre.devices.usbms.device*), 276
DevicePlugin (*Klasse in calibre.devices.interface*), 268
dirty() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 364
do_user_config() (*Methode von calibre.customize.Plugin*), 257
dont_add_to (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
dont_remove_from (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
download() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
download_cover() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 265
drop_event() (*Methode von calibre.gui2.actions.InterfaceAction*), 283

E

ebook-convert command line option
--add-alt-text-to-img, 332
--asciiize, 328
--author-sort, 334
--authors, 334
--base-font-size, 328
--book-producer, 334
--change-justification, 328
--chapter, 332
--chapter-mark, 332
--comments, 334
--cover, 334
--debug-pipeline, 335
--disable-dehyphenate, 331
--disable-delete-blank-paragraphs, 331
--disable-fix-indents, 331
--disable-font-rescaling, 328
--disable-format-scene-breaks, 331
--disable-italicize-common-cases, 331
--disable-markup-chapter-headings, 331
--disable-remove-fake-margins, 332
--disable-renumber-headings, 331
--disable-unwrap-lines, 331
--dont-split-on-page-breaks, 327
--duplicate-links-in-toc, 333
--embed-all-fonts, 328
--embed-font-family, 329
--enable-heuristics, 331
--epub-flatten, 327
--epub-inline-toc, 327
--epub-max-image-size, 327
--epub-toc-at-end, 327
--epub-version, 327
--expand-css, 329
--extra-css, 329
--extract-to, 327
--filter-css, 329
--flow-size, 327
--font-size-mapping, 329
--from-opf, 334
--help, 326
--html-unwrap-factor, 331
--input-encoding, 327
--input-profile, 326
--insert-blank-line, 329
--insert-blank-line-size, 329
--insert-metadata, 332
--isbn, 334
--keep-ligatures, 329
--language, 334
--level1-toc, 333
--level2-toc, 333
--level3-toc, 333
--line-height, 329
--linearize-tables, 329
--list-recipes, 326
--margin-bottom, 330
--margin-left, 330
--margin-right, 330
--margin-top, 330
--max-toc-links, 333
--minimum-line-height, 330
--no-chapters-in-toc, 333
--no-default-epub-cover, 328
--no-svg-cover, 328
--output-profile, 327
--page-breaks-before, 332
--prefer-metadata-cover, 333
--preserve-cover-aspect-ratio, 328
--pretty-print, 328

--pubdate, 334
--publisher, 334
--rating, 334
--read-metadata-from-opf, 334
--remove-first-image, 333
--remove-paragraph-spacing, 330
--remove-paragraph-spacing-indent-size, 330
--replace-scene-breaks, 331
--search-replace, 332
--series, 334
--series-index, 334
--smarten-punctuation, 330
--sr1-replace, 332
--sr1-search, 332
--sr2-replace, 332
--sr2-search, 332
--sr3-replace, 332
--sr3-search, 332
--start-reading-at, 333
--subset-embedded-fonts, 330
--tags, 334
--timestamp, 334
--title, 334
--title-sort, 334
--toc-filter, 333
--toc-threshold, 333
--toc-title, 328
--transform-css-rules, 330
--transform-html-rules, 330
--unsmarten-punctuation, 331
--use-auto-toc, 333
--verbose, 335
--version, 327
-d, 335
-h, 326
-m, 334
-v, 335

ebook-edit command line option
--detach, 335
--help, 335
--select-text, 335
--version, 335
-h, 335

ebook-meta command line option
--author-sort, 336
--authors, 336
--book-producer, 336
--category, 336
--comments, 336
--cover, 336
--date, 336
--disallow-rendered-cover, 336
--from-opf, 336
--get-cover, 336
--help, 336
--identifier, 336
--index, 336
--isbn, 336
--language, 336
--lrf-bookid, 336
--publisher, 336
--rating, 337
--series, 337
--tags, 337
--title, 337
--title-sort, 337
--to-opf, 337
--version, 337
-a, 336
-c, 336
-d, 336
-h, 336
-i, 336
-k, 336
-l, 336
-p, 336
-r, 337
-s, 337
-t, 337

ebook-polish command line option
-H, 337
-U, 338
--add-soft-hyphens, 337
--compress-images, 337
--cover, 337
--download-external-resources, 337
--embed-fonts, 338
--help, 338
--jacket, 338
--opf, 338
--remove-jacket, 338
--remove-soft-hyphens, 338
--remove-unused-css, 338
--smarten-punctuation, 338
--subset-fonts, 338
--upgrade-book, 338
--verbose, 338
--version, 338
-c, 337
-d, 337
-e, 338
-f, 338
-h, 338
-i, 337
-j, 338
-o, 338
-p, 338

-u, 338
ebook-viewer command line option
--continue, 339
--detach, 339
--force-reload, 339
--full-screen, 339
--fullscreen, 339
--help, 339
--new-instance, 339
--open-at, 339
--raise-window, 339
--version, 339
-f, 339
-h, 339
edit_file() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
eject() (*Methode von calibre.devices.interface.DevicePlugin*), 271
eject() (*Methode von calibre.devices.usbms.device.Device*), 279
embed_metadata() (*Methode von calibre.db.cache.Cache*), 356
encoding (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
exists() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
expire_old_trash() (*Methode von calibre.db.cache.Cache*), 356
export_note() (*Methode von calibre.db.cache.Cache*), 356
extra_css (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
extract_readable_article() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42

F

fast_field_for() (*Methode von calibre.db.cache.Cache*), 356
feeds (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 47
fetch-ebook-metadata command line option
-I, 340
--allowed-plugin, 340
--authors, 340
--cover, 340
--help, 340
--identifier, 340
--isbn, 340
--opf, 340
--timeout, 340
--title, 340
--verbose, 340
--version, 340
-a, 340
-c, 340
-d, 340
-h, 340
-i, 340
-o, 340
-p, 340
-t, 340
-v, 340
field_for() (*Methode von calibre.db.cache.Cache*), 357
field_ids_for() (*Methode von calibre.db.cache.Cache*), 357
field_supports_notes() (*Methode von calibre.db.cache.Cache*), 357
file_type (*Attribut von calibre.customize.conversion.OutputFormatPlugin*), 267
file_types (*Attribut von calibre.customize.CatalogPlugin*), 261
file_types (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 265
file_types (*Attribut von calibre.customize.FileTypePlugin*), 258
file_types (*Attribut von calibre.customize.MetadataReaderPlugin*), 260
file_types (*Attribut von calibre.customize.MetadataWriterPlugin*), 260
filename_callback() (*Methode von calibre.devices.usbms.device.Device*), 279
filesize() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
FileTypePlugin (*Klasse in calibre.customize*), 258
filter_css() (*im Modul calibre.ebooks.oeb.polish.css*), 370
filter_regexps (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48
find_identical_books() (*Methode von calibre.db.cache.Cache*), 357
fix_all_html() (*im Modul calibre.ebooks.oeb.polish.pretty*), 368
fix_html() (*im Modul calibre.ebooks.oeb.polish.pretty*), 368
for_viewer (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 266
format() (*Methode von calibre.db.cache.Cache*), 357
format_abspath() (*Methode von calibre.db.cache.Cache*), 357
format_added (*Attribut von calibre.db.cache.Cache.EventType*), 353
format_field() (*Methode von calibre.ebooks.metadata.book.base.Metadata*),

212
format_hash() (*Methode von calibre.db.cache.Cache*), 357
format_metadata() (*Methode von calibre.db.cache.Cache*), 357
FORMATS (*Attribut von calibre.devices.interface.DevicePlugin*), 268
FORMATS (*Attribut von calibre.devices.usbms.driver.USBMS*), 280
formats() (*Methode von calibre.db.cache.Cache*), 358
formats_removed (*Attribut von calibre.db.cache.Cache.EventType*), 353
free_space() (*Methode von calibre.devices.interface.DevicePlugin*), 272
free_space() (*Methode von calibre.devices.usbms.device.Device*), 278
from_files() (*im Modul calibre.ebooks.oeb.polish.toc*), 370
from_links() (*im Modul calibre.ebooks.oeb.polish.toc*), 370
from_xpaths() (*im Modul calibre.ebooks.oeb.polish.toc*), 370

G

generate_item() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
genesis() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
genesis() (*Methode von calibre.gui2.preferences.ConfigWidgetInterface*), 286
get_all_items_that_have_notes() (*Methode von calibre.db.cache.Cache*), 358
get_all_link_maps_for_book() (*Methode von calibre.db.cache.Cache*), 358
get_all_standard_metadata() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
get_all_user_metadata() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
get_annotations() (*Methode von calibre.devices.usbms.device.Device*), 279
get_article_url() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 42
get_author_tokens() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
get_book_url() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
get_book_url_name() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
get_book_urls() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 264
get_browser() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 43
get_cached_cover_url() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 264
get_categories() (*Methode von calibre.db.cache.Cache*), 358
get_collections() (*Methode von calibre.devices.interface.BookList*), 276
get_cover_url() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 43
get_custom_book_data() (*Methode von calibre.db.cache.Cache*), 358
get_device_information() (*Methode von calibre.devices.interface.DevicePlugin*), 271
get_device_information() (*Methode von calibre.devices.usbms.driver.USBMS*), 280
get_device_uid() (*Methode von calibre.devices.interface.DevicePlugin*), 274
get_driveinfo() (*Methode von calibre.devices.interface.DevicePlugin*), 271
get_extra_css() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 43
get_feeds() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 43
get_file() (*Methode von calibre.devices.interface.DevicePlugin*), 273
get_file_path_for_processing() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
get_id_map() (*Methode von calibre.db.cache.Cache*), 358
get_identifiers() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
get_ids_for_custom_book_data() (*Methode von calibre.db.cache.Cache*), 358
get_images() (*Methode von calibre.customize.conversion.InputFormatPlugin*), 266
get_item_id() (*Methode von calibre.db.cache.Cache*), 358
get_item_ids() (*Methode von calibre.db.cache.Cache*), 358
get_item_name() (*Methode von calibre.db.cache.Cache*), 358
get_item_name_map() (*Methode von calibre.db.cache.Cache*), 358
get_link_map() (*Methode von calibre.db.cache.Cache*), 358
get_masthead_title() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 43
get_masthead_url() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 43
get_metadata() (*Methode von calibre.customize.MetadataReaderPlugin*), 260
get_metadata() (*Methode von calibre.db.cache.Cache*),

359
get_next_series_num_for() (*Methode von calibre.db.cache.Cache*), 359
get_notes_resource() (*Methode von calibre.db.cache.Cache*), 359
get_obfuscated_article() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 44
get_open_popup_message() (*Klassenmethode von calibre.devices.interface.DevicePlugin*), 269
get_option() (*Methode von calibre.devices.interface.DevicePlugin*), 275
get_proxy_metadata() (*Methode von calibre.db.cache.Cache*), 359
get_recommended_folders() (*im Modul calibre.ebooks.oeb.polish.replace*), 368
get_standard_metadata() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
get_title_tokens() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
get_url_specific_delay() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 44
get_usage_count_by_id() (*Methode von calibre.db.cache.Cache*), 359
get_user_blacklisted_devices() (*Methode von calibre.devices.interface.DevicePlugin*), 274
get_user_metadata() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
gui (*calibre.gui2.tweak_book.plugin.Tool property*), 371
gui_category (*Attribut von calibre.customize.PreferencesPlugin*), 285
gui_configuration_widget() (*Methode von calibre.customize.conversion.InputFormatPlugin*), 266
gui_configuration_widget() (*Methode von calibre.customize.conversion.OutputFormatPlugin*), 268
gui_layout_complete() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
gui_name (*Attribut von calibre.customize.PreferencesPlugin*), 285
guide_type_map (*calibre.ebooks.oeb.polish.container.Container property*), 365

H

handle_gzip (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48
has_book() (*Methode von calibre.db.cache.Cache*), 359
has_format() (*Methode von calibre.db.cache.Cache*), 359
has_html_comments (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
has_id() (*Methode von calibre.db.cache.Cache*), 359
has_name() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
href_to_name() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
HTML, **377**

I

icon (*Attribut von calibre.customize.PreferencesPlugin*), 286
icon (*Attribut von calibre.devices.interface.DevicePlugin*), 269
id_from_url() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 264
identify() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 264
identify_results_keygen() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 264
ignore_connected_device() (*Methode von calibre.devices.interface.DevicePlugin*), 274
ignore_duplicate_articles (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48
ignore_ssl_errors (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
image_url_processor() (*Klassenmethode von calibre.web.feeds.news.BasicNewsRecipe*), 41
import_note() (*Methode von calibre.db.cache.Cache*), 359
index_to_soup() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 44
indexing_progress_changed (*Attribut von calibre.db.cache.Cache.EventType*), 354
init() (*Methode von calibre.db.cache.Cache*), 359
initial_tab_changed() (*Methode von calibre.gui2.preferences.ConfigWidgetInterface*), 286
initialization_complete() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
initialize() (*Methode von calibre.customize.CatalogPlugin*), 261
initialize() (*Methode von calibre.customize.Plugin*), 257
initialize() (*Methode von calibre.gui2.preferences.ConfigWidgetBase*), 287
initialize() (*Methode von calibre.gui2.preferences.ConfigWidgetInterface*), 286
InputFormatPlugin (*Klasse in calibre.customize.conversion*), 265
insert_into_xml() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365

installation_type (*Attribut von calibre.customize.Plugin*), 257
InterfaceAction (*Klasse in calibre.gui2.actions*), 281
InterfaceActionBase (*Klasse in calibre.customize*), 284
InternalMetadataCompareKeyGen (*Klasse in calibre.ebooks.metadata.sources.base*), 265
is_configured() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 262
is_dir (*Attribut von calibre.ebooks.oeb.polish.container.Container*), 365
is_dynamically_controllable() (*Methode von calibre.devices.interface.DevicePlugin*), 274
is_image_collection (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 265
is_link_wanted() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 44
is_null() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 210
is_running() (*Methode von calibre.devices.interface.DevicePlugin*), 275
is_usb_connected() (*Methode von calibre.devices.interface.DevicePlugin*), 270
items_removed (*Attribut von calibre.db.cache.Cache.EventType*), 354
items_renamed (*Attribut von calibre.db.cache.Cache.EventType*), 354
items_with_notes_in_book() (*Methode von calibre.db.cache.Cache*), 359
iterlinks() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365

K

keep_only_tags (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48

L

language (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48
library_about_to_change() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
library_changed() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
link_for() (*Methode von calibre.db.cache.Cache*), 359
links_changed (*Attribut von calibre.db.cache.Cache.EventType*), 354
list_extra_files() (*Methode von calibre.db.cache.Cache*), 359
load_actual_plugin() (*Methode von calibre.customize.InterfaceActionBase*), 285
load_resources() (*Methode von calibre.customize.Plugin*), 257
load_resources() (*Methode von calibre.gui2.actions.InterfaceAction*), 283
location_selected() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
LRF, **377**
lrf2lrs command line option
    --dont-output-resources, 340
    --help, 340
    --output, 340
    --verbose, 341
    --version, 341
    -h, 340
    -o, 340
lrfviewer command line option
    --disable-hyphenation, 341
    --help, 341
    --profile, 341
    --verbose, 341
    --version, 341
    --visual-debug, 341
    --white-background, 341
    -h, 341
lrs2lrf command line option
    --help, 341
    --lrs, 341
    --output, 342
    --verbose, 342
    --version, 342
    -h, 341
    -o, 342

M

make_name_unique() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
MANAGES_DEVICE_PRESENCE (*Attribut von calibre.devices.interface.DevicePlugin*), 269
manifest_has_name() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
manifest_id_map (*calibre.ebooks.oeb.polish.container.Container property*), 365
manifest_items_of_type() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
manifest_items_with_property() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 365
manifest_type_map (*calibre.ebooks.oeb.polish.container.Container property*), 365

mark_as_cover() (*im Modul calibre.ebooks.oeb.polish.cover*), 369
mark_as_titlepage() (*im Modul calibre.ebooks.oeb.polish.cover*), 369
masthead_url (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48
match_regexps (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48
max_articles_per_feed (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 48
MAX_PATH_LEN (*Attribut von calibre.devices.usbms.device.Device*), 277
merge() (*im Modul calibre.ebooks.oeb.polish.split*), 369
merge_annotations_for_book() (*Methode von calibre.db.cache.Cache*), 360
merge_extra_files() (*Methode von calibre.db.cache.Cache*), 360
Metadata (*Klasse in calibre.ebooks.metadata.book.base*), 210
metadata_changed (*Attribut von calibre.db.cache.Cache.EventType*), 354
metadata_for_field() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
MetadataReaderPlugin (*Klasse in calibre.customize*), 260
MetadataWriterPlugin (*Klasse in calibre.customize*), 260
mi (*calibre.ebooks.oeb.polish.container.Container property*), 366
minimum_calibre_version (*Attribut von calibre.customize.Plugin*), 257
model_metadata() (*Klassenmethode von calibre.devices.interface.DevicePlugin*), 270
model_metadata() (*Klassenmethode von calibre.devices.usbms.device.Device*), 277
module
    calibre.customize, 256
    calibre.customize.conversion, 265
    calibre.db.cache, 353
    calibre.devices.interface, 268
    calibre.ebooks.metadata.book.base, 210
    calibre.ebooks.metadata.sources.base, 262
    calibre.ebooks.oeb.polish.container, 363
    calibre.ebooks.oeb.polish.cover, 369
    calibre.ebooks.oeb.polish.css, 370
    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, 184
    calibre.web.feeds.news, 41
move_book_from_trash() (*Methode von calibre.db.cache.Cache*), 360
move_format_from_trash() (*Methode von calibre.db.cache.Cache*), 360
multisort() (*Methode von calibre.db.cache.Cache*), 360
multisplit() (*im Modul calibre.ebooks.oeb.polish.split*), 369

N

name (*Attribut von calibre.customize.Plugin*), 256
name (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
name (*Attribut von calibre.gui2.tweak_book.plugin.Tool*), 371
name(), **160**
name_order (*Attribut von calibre.customize.PreferencesPlugin*), 285
name_to_abspath() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 366
name_to_href() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 366
names_that_must_not_be_changed (*calibre.ebooks.oeb.polish.container.Container property*), 366
names_that_must_not_be_removed (*calibre.ebooks.oeb.polish.container.Container property*), 366
names_that_need_not_be_manifested (*calibre.ebooks.oeb.polish.container.Container property*), 366
needs_subscription (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49
NEWS_IN_FOLDER (*Attribut von calibre.devices.usbms.device.Device*), 277
no_stylesheets (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49
normalize_path() (*Klassenmethode von calibre.devices.usbms.driver.USBMS*), 281
notes_changed (*Attribut von calibre.db.cache.Cache.EventType*), 354
notes_data_for() (*Methode von calibre.db.cache.Cache*), 360
notes_for() (*Methode von calibre.db.cache.Cache*), 360
notes_resources_used_by() (*Methode von calibre.db.cache.Cache*), 360
NUKE_COMMENTS (*Attribut von calibre.devices.interface.DevicePlugin*), 269

O

oldest_article (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49

on_import (*Attribut von calibre.customize.FileTypePlugin*), 258
on_postconvert (*Attribut von calibre.customize.FileTypePlugin*), 258
on_postdelete (*Attribut von calibre.customize.FileTypePlugin*), 258
on_postimport (*Attribut von calibre.customize.FileTypePlugin*), 258
on_postprocess (*Attribut von calibre.customize.FileTypePlugin*), 259
on_preprocess (*Attribut von calibre.customize.FileTypePlugin*), 259
open() (*Methode von calibre.devices.interface.DevicePlugin*), 271
open() (*Methode von calibre.devices.usbms.device.Device*), 278
open() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 366
open_book() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
OPEN_FEEDBACK_MESSAGE (*Attribut von calibre.devices.interface.DevicePlugin*), 269
opf (*calibre.ebooks.oeb.polish.container.Container property*), 366
opf_get_or_create() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 366
opf_version (*calibre.ebooks.oeb.polish.container.Container property*), 366
opf_version_parsed (*calibre.ebooks.oeb.polish.container.Container property*), 366
opf_xpath() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 366
options (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 266
options (*Attribut von calibre.customize.conversion.OutputFormatPlugin*), 267
options (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
OSX_MAIN_MEM_VOL_PAT (*Attribut von calibre.devices.usbms.device.Device*), 277
output_encoding (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 266
OutputFormatPlugin (*Klasse in calibre.customize.conversion*), 267

P

parse_feeds() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 44
parse_index() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 44
parsed() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 366
path_sep (*Attribut von calibre.devices.interface.DevicePlugin*), 269
Plugin (*Klasse in calibre.customize*), 256
populate_article_metadata() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 45
popup_type (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
post_yank_cleanup() (*Methode von calibre.devices.interface.DevicePlugin*), 271
post_yank_cleanup() (*Methode von calibre.devices.usbms.device.Device*), 279
postadd() (*Methode von calibre.customize.FileTypePlugin*), 260
postconvert() (*Methode von calibre.customize.FileTypePlugin*), 259
postdelete() (*Methode von calibre.customize.FileTypePlugin*), 259
postimport() (*Methode von calibre.customize.FileTypePlugin*), 259
postprocess_book() (*Methode von calibre.customize.conversion.InputFormatPlugin*), 266
postprocess_book() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 45
postprocess_html() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 45
pref() (*Methode von calibre.db.cache.Cache*), 360
prefer_results_with_isbn (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
PreferencesPlugin (*Klasse in calibre.customize*), 285
prepare_addable_books() (*Methode von calibre.devices.interface.DevicePlugin*), 274
preprocess_html() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 45
preprocess_image() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 45
preprocess_raw_html() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 45
preprocess_regexps (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49
pretty_all() (*im Modul calibre.ebooks.oeb.polish.pretty*), 368
pretty_css() (*im Modul calibre.ebooks.oeb.polish.pretty*), 368
pretty_html() (*im Modul calibre.ebooks.oeb.polish.pretty*), 368
pretty_xml() (*im Modul calibre.ebooks.oeb.polish.pretty*), 368

print_version() (*Klassenmethode von calibre.web.feeds.news.BasicNewsRecipe*), 41
priority (*Attribut von calibre.customize.Plugin*), 257
priority (*Attribut von calibre.gui2.actions.InterfaceAction*), 282
PRODUCT_ID (*Attribut von calibre.devices.interface.DevicePlugin*), 268
PRODUCT_ID (*Attribut von calibre.devices.usbms.device.Device*), 277
publication_date() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 45
publication_type (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49

R

raw_data() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 366
re:test(), **160**
read_backup() (*Methode von calibre.db.cache.Cache*), 360
recipe_disabled (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49
recipe_specific_options (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49
recommendations (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 266
recommendations (*Attribut von calibre.customize.conversion.OutputFormatPlugin*), 267
recursions (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49
refresh_gui() (*Methode von calibre.gui2.preferences.ConfigWidgetInterface*), 286
Regexp, **377**
register() (*Methode von calibre.gui2.preferences.ConfigWidgetBase*), 287
register_shortcut() (*Methode von calibre.gui2.tweak_book.plugin.Tool*), 371
relpath() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
remove_attributes (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 49
remove_book() (*Methode von calibre.devices.interface.BookList*), 276
remove_books() (*Methode von calibre.db.cache.Cache*), 360
remove_books_from_metadata() (*Klassenmethode von calibre.devices.interface.DevicePlugin*), 273
remove_books_from_metadata() (*Methode von calibre.devices.usbms.driver.USBMS*), 281
remove_empty_feeds (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_extra_files() (*Methode von calibre.db.cache.Cache*), 360
remove_formats() (*Methode von calibre.db.cache.Cache*), 360
remove_from_spine() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
remove_from_xml() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
remove_item() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
remove_items() (*Methode von calibre.db.cache.Cache*), 361
remove_jacket() (*im Modul calibre.ebooks.oeb.polish.jacket*), 368
remove_javascript (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_stale_user_metadata() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
remove_tags (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_tags_after (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_tags_before (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_unused_css() (*im Modul calibre.ebooks.oeb.polish.css*), 370
rename() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
rename_extra_files() (*Methode von calibre.db.cache.Cache*), 361
rename_files() (*im Modul calibre.ebooks.oeb.polish.replace*), 368
rename_items() (*Methode von calibre.db.cache.Cache*), 361
replace() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
replace_links() (*im Modul calibre.ebooks.oeb.polish.replace*), 367
replace_links() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
requires_version (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
reset() (*Methode von calibre.devices.interface.DevicePlugin*), 270
reset() (*Methode von calibre.devices.usbms.device.Device*), 277

resolve_internal_links (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
restart_critical (*Attribut von calibre.gui2.preferences.ConfigWidgetBase*), 287
restart_critical (*Attribut von calibre.gui2.preferences.ConfigWidgetInterface*), 286
restore_book() (*Methode von calibre.db.cache.Cache*), 361
restore_defaults() (*Methode von calibre.gui2.preferences.ConfigWidgetBase*), 287
restore_defaults() (*Methode von calibre.gui2.preferences.ConfigWidgetInterface*), 286
restore_defaults_desc (*Attribut von calibre.gui2.preferences.ConfigWidgetInterface*), 286
restore_original_format() (*Methode von calibre.db.cache.Cache*), 361
reverse_article_order (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
rewind_savepoint() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
RSS, **377**
run() (*Methode von calibre.customize.CatalogPlugin*), 261
run() (*Methode von calibre.customize.FileTypePlugin*), 259

S

safe_read_lock (*calibre.db.cache.Cache property*), 361
sanitize_callback() (*Methode von calibre.devices.usbms.device.Device*), 279
sanitize_path_components() (*Methode von calibre.devices.usbms.device.Device*), 279
save_book() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
save_original_format() (*Methode von calibre.db.cache.Cache*), 361
save_settings() (*Klassenmethode von calibre.devices.interface.DevicePlugin*), 273
save_settings() (*Methode von calibre.customize.Plugin*), 257
save_settings() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
scale_news_images (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 50
scale_news_images_to_device (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
Schema, **377**
search() (*Methode von calibre.db.cache.Cache*), 361
search_annotations() (*Methode von calibre.db.cache.Cache*), 362
search_notes() (*Methode von calibre.db.cache.Cache*), 362
serialize_item() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
set_all_user_metadata() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
set_annotations_for_book() (*Methode von calibre.db.cache.Cache*), 362
set_conversion_options() (*Methode von calibre.db.cache.Cache*), 362
set_cover() (*im Modul calibre.ebooks.oeb.polish.cover*), 369
set_cover() (*Methode von calibre.db.cache.Cache*), 362
set_driveinfo_name() (*Methode von calibre.devices.interface.DevicePlugin*), 273
set_driveinfo_name() (*Methode von calibre.devices.usbms.driver.USBMS*), 280
set_field() (*Methode von calibre.db.cache.Cache*), 362
set_identifier() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
set_identifiers() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
set_library_info() (*Methode von calibre.devices.interface.DevicePlugin*), 274
set_link_map() (*Methode von calibre.db.cache.Cache*), 362
set_metadata() (*Methode von calibre.customize.MetadataWriterPlugin*), 261
set_metadata() (*Methode von calibre.db.cache.Cache*), 362
set_modified() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
set_notes_for() (*Methode von calibre.db.cache.Cache*), 362
set_option() (*Methode von calibre.devices.interface.DevicePlugin*), 275
set_plugboards() (*Methode von calibre.devices.interface.DevicePlugin*), 273
set_pref() (*Methode von calibre.db.cache.Cache*), 363
set_progress_reporter() (*Methode von calibre.devices.interface.DevicePlugin*), 271
set_progress_reporter() (*Methode von calibre.devices.usbms.device.Device*), 277
set_spine() (*Methode von calibre.ebooks.oeb.polish.container.Container*), 367
set_user_blacklisted_devices() (*Methode von calibre.devices.interface.DevicePlugin*), 274
set_user_metadata() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
settings() (*Klassenmethode von calibre.devices.interface.DevicePlugin*), 273

show_current_diff() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
show_editor() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
shutdown() (*Methode von calibre.devices.interface.DevicePlugin*), 274
shutting_down() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
simultaneous_downloads (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
skip_ad_pages() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 46
SLOW_DRIVEINFO (*Attribut von calibre.devices.interface.DevicePlugin*), 269
smart_update() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
sort_index_by() (*Methode von calibre.web.feeds.news.BasicNewsRecipe*), 46
Source (*Klasse in calibre.ebooks.metadata.sources.base*), 262
specialize() (*Methode von calibre.customize.conversion.InputFormatPlugin*), 266
specialize_css_for_output() (*Methode von calibre.customize.conversion.OutputFormatPlugin*), 268
specialize_global_preferences() (*Methode von calibre.devices.interface.DevicePlugin*), 274
specialize_options() (*Methode von calibre.customize.conversion.OutputFormatPlugin*), 267
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() (*im Modul calibre.ebooks.oeb.polish.split*), 368
split_if_is_multiple_composite() (*Methode von calibre.db.cache.Cache*), 363
split_jobs() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
standard_field_keys() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
STANDARD_METADATA_FIELDS (*in Modul calibre.ebooks.metadata.book.base*), 212
start_plugin() (*Methode von calibre.devices.interface.DevicePlugin*), 274
startup() (*Methode von calibre.devices.interface.DevicePlugin*), 274
stop_plugin() (*Methode von calibre.devices.interface.DevicePlugin*), 274
summary_length (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
supported_platforms (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 265
supported_platforms (*Attribut von calibre.customize.conversion.OutputFormatPlugin*), 267
supported_platforms (*Attribut von calibre.customize.InterfaceActionBase*), 284
supported_platforms (*Attribut von calibre.customize.MetadataReaderPlugin*), 260
supported_platforms (*Attribut von calibre.customize.MetadataWriterPlugin*), 261
supported_platforms (*Attribut von calibre.customize.Plugin*), 256
supported_platforms (*Attribut von calibre.customize.PreferencesPlugin*), 285
supported_platforms (*Attribut von calibre.devices.usbms.driver.USBMS*), 279
supported_platforms (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
supports_collections() (*Methode von calibre.devices.interface.BookList*), 276
supports_gzip_transfer_encoding (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
supports_restoring_to_defaults (*Attribut von calibre.gui2.preferences.ConfigWidgetBase*), 287
supports_restoring_to_defaults (*Attribut von calibre.gui2.preferences.ConfigWidgetInterface*), 286
sync_booklists() (*Methode von calibre.devices.interface.DevicePlugin*), 273
sync_booklists() (*Methode von calibre.devices.usbms.driver.USBMS*), 281
sync_preview_to_editor() (*Methode von calibre.gui2.tweak_book.boss.Boss*), 373
synchronize_with_db() (*Methode von calibre.devices.interface.DevicePlugin*), 275

T

tag_browser_context_action() (*Methode von calibre.gui2.actions.InterfaceAction*), 284
tag_to_string() (*Klassenmethode von calibre.web.feeds.news.BasicNewsRecipe*), 42
tags_older_than() (*Methode von calibre.db.cache.Cache*), 363
template_css (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
template_to_attribute() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 211
temporary_file() (*Methode von calibre.customize.Plugin*), 258

test_fields() (*Methode von calibre.ebooks.metadata.sources.base.Source*), 263
THUMBNAIL_COMPRESSION_QUALITY (*Attribut von calibre.devices.interface.DevicePlugin*), 268
THUMBNAIL_HEIGHT (*Attribut von calibre.devices.interface.DevicePlugin*), 268
timefmt (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
timeout (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
title (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
to_html() (*Methode von calibre.ebooks.metadata.book.base.Metadata*), 212
Tool (*Klasse in calibre.gui2.tweak_book.plugin*), 371
toolbar_button_popup_mode (*Attribut von calibre.gui2.tweak_book.plugin.Tool*), 371
total_space() (*Methode von calibre.devices.interface.DevicePlugin*), 271
total_space() (*Methode von calibre.devices.usbms.device.Device*), 278
touched_fields (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262
type (*Attribut von calibre.customize.CatalogPlugin*), 261
type (*Attribut von calibre.customize.conversion.InputFormatPlugin*), 265
type (*Attribut von calibre.customize.conversion.OutputFormatPlugin*), 267
type (*Attribut von calibre.customize.FileTypePlugin*), 259
type (*Attribut von calibre.customize.InterfaceActionBase*), 285
type (*Attribut von calibre.customize.MetadataReaderPlugin*), 260
type (*Attribut von calibre.customize.MetadataWriterPlugin*), 261
type (*Attribut von calibre.customize.Plugin*), 257
type (*Attribut von calibre.customize.PreferencesPlugin*), 285
type (*Attribut von calibre.devices.interface.DevicePlugin*), 268
type (*Attribut von calibre.ebooks.metadata.sources.base.Source*), 262

U

unretire_note_for() (*Methode von calibre.db.cache.Cache*), 363
update_annotations() (*Methode von calibre.db.cache.Cache*), 363
upload_books() (*Methode von calibre.devices.interface.DevicePlugin*), 272
upload_books() (*Methode von calibre.devices.usbms.driver.USBMS*), 280
upload_cover() (*Methode von calibre.devices.usbms.driver.USBMS*), 281
URL, **377**
USBMS (*Klasse in calibre.devices.usbms.driver*), 279
use_embedded_content (*Attribut von calibre.web.feeds.news.BasicNewsRecipe*), 51
user_categories_for_books() (*Methode von calibre.db.cache.Cache*), 363
user_feedback_after_callback (*Attribut von calibre.devices.interface.DevicePlugin*), 269
UserAnnotation (*Attribut von calibre.devices.interface.DevicePlugin*), 269

V

VENDOR_ID (*Attribut von calibre.devices.interface.DevicePlugin*), 268
VENDOR_ID (*Attribut von calibre.devices.usbms.device.Device*), 277
version (*Attribut von calibre.customize.MetadataReaderPlugin*), 260
version (*Attribut von calibre.customize.MetadataWriterPlugin*), 261
version (*Attribut von calibre.customize.Plugin*), 256
VIRTUAL_BOOK_EXTENSION_MESSAGE (*Attribut von calibre.devices.interface.DevicePlugin*), 269
VIRTUAL_BOOK_EXTENSIONS (*Attribut von calibre.devices.interface.DevicePlugin*), 269

W

WANTS_UPDATED_THUMBNAILS (*Attribut von calibre.devices.interface.DevicePlugin*), 268
web2disk command line option
    --base-dir, 342
    --delay, 342
    --dont-download-stylesheets, 342
    --encoding, 342
    --filter-regexp, 342
    --help, 342
    --match-regexp, 342
    --max-files, 342
    --max-recursions, 342
    --timeout, 342
    --verbose, 343
    --version, 343
    -d, 342
    -h, 342
    -n, 342
    -r, 342
    -t, 342
WINDOWS_CARD_A_MEM (*Attribut von calibre.devices.usbms.device.Device*), 277

WINDOWS_CARD_B_MEM (*Attribut von calibre.devices.usbms.device.Device*), 277
WINDOWS_MAIN_MEM (*Attribut von calibre.devices.usbms.device.Device*), 277
windows_sort_drives() (*Methode von calibre.devices.usbms.device.Device*), 278