Руководство пользователя calibre

Выпуск 8.5.0

Kovid Goyal

июня 20, 2025

Оглавление

1 Графический интерфейс пользователя 3

2 Добавление любимых новостных веб-сайтов 33

3 Просмотрщик электронных книг 53

4 Конвертирование книг 61

5 Редактирование книг 81

6 Контент-сервер calibre 115

7 Сравнение книг 123

8 Редактирование метаданных книги 127

9 Часто задаваемые вопросы 131

10 Руководства 157

11 Схема URL-адресов calibre:// 259

12 Настройка calibre 263

13 Командная строка 313

14 Настройка среды разработки calibre 355

15 Технические средства защиты авторских прав (DRM — digital rights management) 387

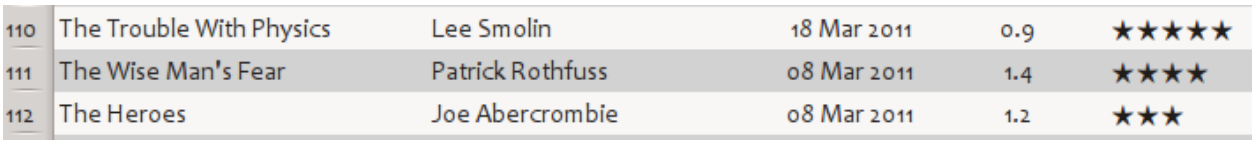
16 Глоссарий 389

Содержание модулей Python 391

Алфавитный указатель 393

calibre — программа управления электронной библиотекой. В ней можно читать, конвертировать и каталогизировать книги в большинстве известных форматов. Она умеет взаимодействовать со многими устройствами чтения электронных книг, выходить в Интернет и получать метаданные для ваших книг. Можно загружать газеты и для удобства превращать их в электронные книги. Программа кроссплатформенная: работает в Linux, Windows и macOS.

Итак, вы запустили calibre, что дальше? Прежде чем сделать что-либо с вашими электронными книгами, программа должна о них узнать. Перетащите несколько файлов электронных книг в calibre или нажмите кнопку «Добавить книги» и найдите те, с которыми хотите работать. Добавленные книги появятся в главном окне, примерно так:



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	★★★

Когда вы всласть налюбуетесь списком добавленных книг, вам, возможно, захочется прочесть одну из них. Для этого нужно преобразовать книгу в формат, понятный устройству. При первом запуске calibre откроется *Мастер начальной настройки* и настроит программу для вашего устройства чтения. Преобразование делается легче лёгкого. Просто выберите книгу, которую хотите преобразовать, и щёлкните по кнопке «Преобразовать книги». Проигнорируйте пока все опции и нажмите «OK». В правом нижнем углу начнёт вращаться маленький значок. Как только вращение прекратилось, преобразованная книга готова. Нажмите кнопку «Просмотр», чтобы приступить к чтению.

Если вы хотите читать книгу на своём устройстве чтения, подключите его к компьютеру, подождите, пока calibre определит его (10—20 секунд), и нажмите кнопку «Отправить на устройство». Когда значок снова перестанет вращаться, отключите устройство и погружайтесь в чтение! Если вы не преобразовали книгу на предыдущем шаге, calibre автоматически преобразует её в формат, понятный вашему устройству.

Чтобы использовать программу на более продвинутом уровне, прочтите раздел «*Графический интерфейс пользователя* (страница 3)». Станьте ещё круче и многогранней, изучив раздел «*Командная строка* (страница 313)». Также небезынтересными могут оказаться ответы на «*Часто задаваемые вопросы* (страница 131)».

Если у вас ещё остались вопросы, или вы хотите обсудить calibre с другими пользователями или попросить помощи с конкретными вещами, есть **:вебсайт:`форумы и другие справочные ресурсы <help>`**

Разделы

ГЛАВА 1

Графический интерфейс пользователя

Графический интерфейс пользователя *(GUI)* обеспечивает доступ ко всем функциям управления библиотекой и преобразования форматов электронных книг. Основная работа в calibre начинается с добавления в библиотеку книг с вашего жёсткого диска. Программа автоматически попытается прочитать метаданные из книг и добавить их во внутреннюю базу данных. Когда они окажутся в базе данных, с ними можно выполнять различные: ref:*actions*, включающие преобразование из одного формата в другой, отправку на устройство чтения, просмотр на компьютере и редактирование метаданных. Последнее, кроме прочего, позволяет изменять обложку, описание и теги. Обратите внимание, что calibre создаёт копии добавляемых файлов, а исходные файлы остаются нетронутыми.

Интерфейс разделён на различные секции:

- *Действия* (страница 4)
- *Параметры* (страница 11)
- *Каталоги* (страница 11)
- *Поиск и сортировка* (страница 12)
- *Интерфейс поиска* (страница 12)
- *Сохранение поисковых строк* (страница 18)
- *Поиск по полному тексту всех книг* (страница 19)
- *Виртуальные библиотеки* (страница 19)
- *Временно пометить книги* (страница 19)
- *Предположение метаданных на основе имён файлов* (страница 19)
- *Описание книги* (страница 21)
- *Браузер тегов* (страница 23)
- *Сетка обложек* (страница 26)

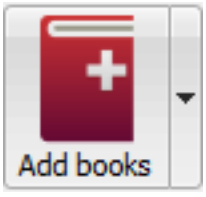
- *Браузер обложек* (страница 27)
- *Добавление примечаний для авторов, серий и т.д.* (страница 28)
- *Быстрый просмотр* (страница 29)
- *Задания* (страница 30)
- *Сочетания клавиш* (страница 30)

1.1 Действия

Панель действий предоставляет удобные ярлыки для часто используемых действий. Если вы щёлкните по ним правой кнопкой мыши, то сможете выполнить вариации стандартных действий. Обратите внимание, что панель действий будет выглядеть несколько иначе, если к компьютеру подключена читалка.

- *Добавить книги* (страница 4)
- *Редактировать метаданные* (страница 5)
- *Преобразовать книги* (страница 6)
- *Просмотр* (страница 6)
- *Отправить на устройство* (страница 7)
- *Собрать новости* (страница 7)
- *Библиотека* (страница 8)
- *Устройство* (страница 9)
- *Сохранить на диск* (страница 9)
- *Подключить/поделиться* (страница 10)
- *Удалить книги* (страница 10)

1.1.1 Добавить книги



Действие *Добавить книги* имеет семь вариантов, доступных по щелчку на нём правой кнопкой мыши.

1. **Добавить книги из одной папки**: открывает диалоговое окно выбора файлов и позволяет указать, какие книги в папке следует добавить. Это действие *зависит от контекста*, т. е. зависит от того, какой *catalog* (страница 11) вы выбрали. Если вы выбрали *Библиотека*, книги будут добавлены в библиотеку. Если вы выбрали устройство для чтения электронных книг, книги будут загружены на устройство и так далее.

2. **Добавить книги из папок и подпапок**: позволяет выбрать папку. Папка и все ее подпапки сканируются рекурсивно, и все найденные электронные книги добавляются в библиотеку. Вы можете выбрать, будет ли calibre добавлять все файлы, находящиеся в одной папке, к одной записи книги или к нескольким записям книги. calibre предполагает, что каждая папка содержит одну книгу. Предполагается, что все файлы электронных книг в папке являются одной и той же книгой в разных форматах. Это действие является обратным действию *Сохранить на диск* (страница 9), то есть вы можете *Сохранить на диск*, удалять книги и повторно добавлять их в режиме отдельной книги для каждой папки без потери информации, за исключением дату (предполагается, что вы не изменили никаких настроек для действия «Сохранить на диск»).
3. **Добавить несколько книг из архива (ZIP/RAR)**: позволяет добавить несколько электронных книг, хранящихся в выбранных файлах ZIP или RAR. Это удобный ярлык, который позволяет избежать необходимости распаковывать архив, а затем добавлять книги с помощью одного из вышеуказанных вариантов.
4. **Добавить пустую книгу (запись о книге без форматов)**: позволяет создать пустую книгу. Полезно, чтобы вручную заполнить информацию о книге, которой, возможно, ещё нет в вашей коллекции.
5. **Добавить по ISBN**: позволяет добавить одну или несколько книг, введя их ISBN.
6. **Добавить файлы к выбранным записям книг**: позволяет добавить или обновить файлы, связанные с существующей книгой в библиотеке.
7. **Добавить файлы данных к выбранным записям книг**: позволяет добавить любое количество дополнительных файлов, которые будут храниться в подкаталоге `data` в каталоге книг. Подробности смотрите в *Добавить дополнительные файлы данных в книгу* (страница 130).
8. **Добавить пустой файл к выбранным записям книг**: позволяет добавить пустой файл указанного формата к выбранным записям книг.

Действие *Добавить книги* может читать метаданные из самых разных форматов электронных книг. Кроме того, оно пытается угадать метаданные из имени файла. См. *Предположение метаданных на основе имён файлов* (страница 19), чтобы узнать, как это настроить.

Чтобы добавить дополнительный формат для существующей книги вы можете сделать любую из трёх вещей:

1. Перетащите файл на панель «Описание книги» в правой части главного окна
2. Щёлкните правой кнопкой мыши по значку *Добавить книги* и выберите *Добавить файлы к выбранным записям книг*.
3. Нажмите *Добавить книги* в верхней правой области диалогового окна *Редактирование метаданных*, доступного через действие *Редактировать метаданные* (страница 5).

1.1.2 Редактировать метаданные

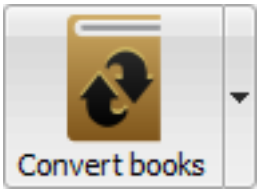
Действие *Редактировать метаданные* имеет четыре варианта, которые доступны по правому щелчку мышью на кнопке.

1. **Редактировать метаданные индивидуально**: позволяет редактировать метаданные книг поочерёдно с возможностью их получения, включая обложки, из Интернета. Также позволяет добавлять или удалять отдельные форматы электронных книг.
2. **Редактировать метаданные массово**: позволяет редактировать общие поля метаданных для большого количества книг одновременно. Работает со всеми книгами, выделенными в *библиотеке* (страница 12).
3. **Загрузить метаданные и обложки**: загружает метаданные и обложки (если доступны) для выбранных в списке книг.

4. **Объединить записи книг**: позволяет объединить метаданные и форматы двух и более записей книг. Вы можете как удалить, так и сохранить записи, нажатые не первыми.
5. **Управление файлами данных**: управление дополнительными файлами данных, связанными с выбранными книгами.

Подробнее смотрите *Редактирование метаданных книги* (страница 127).

1.1.3 Преобразовать книги



Электронные книги можно преобразовать из нескольких форматов в любой предпочтительный для вашего устройства чтения. Многие электронные книги, доступные для покупки, будут защищены технологией *Digital Rights Management* (страница 387) *(DRM)*. Такие книги calibre конвертировать не будет. Из многих форматов DRM легко удалить, однако это может быть незаконно, и вам придётся самостоятельно найти инструменты, чтобы освободить свои книги, а затем использовать calibre для их преобразования.

Для большинства людей преобразование — дело одного щелчка мыши. Если вы хотите узнать больше о процессе преобразования, смотрите *Конвертирование книг* (страница 61).

Действие *Преобразовать книги* имеет три варианта, доступные по правому щелчку мышью по кнопке.

1. **Индивидуальное преобразование**: позволяет указать параметры для настройки преобразования каждой выбранной электронной книги.
2. **Общее преобразование**: позволяет однократно указать параметры для преобразования нескольких электронных книг сразу.
3. **Создать каталог книг в вашей библиотеке calibre**: позволяет создать полный список книг в вашей библиотеке, включая все метаданные, в нескольких форматах, таких как XML, CSV, BiBTeX, EPUB и MOBI. Каталог будет содержать все книги, которые в данный момент отображены в списке библиотеки. Это позволяет использовать возможности поиска, чтобы ограничить количество книг, подлежащих каталогизации. Кроме того, если вы выделите несколько книг с помощью мыши, в каталог будут добавлены только они. Если вы создаёте каталог в формате электронной книги, таком как EPUB, MOBI или AZW3, при следующем подключении вашей читалки он будет автоматически отправлен на устройство. Подробнее о работе каталогов читайте в *Создание AZW3 • EPUB • MOBI каталогов* (страница 251).

1.1.4 Просмотр

Действие *View* отображает книгу в программе просмотра электронных книг. calibre имеет встроенную программу просмотра многих форматов электронных книг. Для других форматов используется приложение операционной системы по умолчанию. Вы можете настроить, какие форматы должны открываться во внутреннем средстве просмотра через *Preferences-Interface-Behavior*. Если книга имеет более одного формата, вы можете просмотреть определенный формат, щелкнув кнопку правой кнопкой мыши.

1.1.5 Отправить на устройство

Действие *Отправить на устройство* имеет восемь вариантов, которые доступны по правому щелчку мышью на кнопке.

1. **Отправить в основную память**: выбранные книги переносятся в основную память устройства чтения электронных книг.
2. **Отправить на карту памяти A**: выбранные книги переносятся на карту памяти A на устройстве чтения электронных книг.
3. **Отправить на карту памяти B**: выбранные книги переносятся на карту памяти B на устройстве чтения электронных книг.
4. **Отправить определённый формат…**: выделенные книги переносятся в выбранное место хранения на устройстве в указанном формате.
5. **Извлечь устройство**: отсоединяет устройство от calibre.
6. **Установить стандартное действие отправки на устройство**: позволяет задать, какая из настроек, с 1 по 5 выше или 7 ниже, будет стандартным действием при нажатии основной кнопки.
7. **Отправить и удалить из библиотеки**: выделенные книги передаются в выбранное место хранения на устройстве и потом **удаляются** из библиотеки.
8. **Извлечь заметки (экспериментально)**: переносит заметки, которые вы, возможно, сделали в электронной книге на своём устройстве, в метаданные комментариев к книге в библиотеке calibre.

Вы можете управлять именем файла и структурой папок файлов, отправленных на устройство, настроив шаблон в *Параметры → Импорт-экспорт → Передача книг на устройства*. См. также «*Язык шаблонов calibre* (страница 164)».

1.1.6 Собрать новости

Действие *Собрать новости* загружает новости с различных веб-сайтов и преобразовывает их в электронную книгу, которую можно прочитать на вашей читалке. Как правило, вновь созданная электронная книга добавляется в вашу библиотеку электронных книг, но устройство чтения подключено в момент завершения загрузки, то новости автоматически загружаются и на него.

Действие *Собрать новости* использует простые рецепты (10—15 строк кода) для каждого сайта новостей. Чтобы узнать, как создавать рецепты для ваших источников новостей, смотрите «*Добавление любимых новостных веб-сайтов* (страница 33)».

Действие *Получить новости* имеет три варианта, доступные по правому щелчку мышью на кнопке.

1. **Загрузка новостей по расписанию**: позволяет планировать загрузку выбранных источников новостей из списка сотен доступных. Расписание можно настроить индивидуально для каждого выбранного источника новостей, оно предоставляет гибкий выбор определённых дней недели или периодичности между загрузками.

2. **Добавить источник новостей**: позволяет создать простой рецепт загрузки новостей с произвольного новостного сайта, к которому вы бы хотели обращаться. Создание рецепта - просто указание URL-адреса RSS-ленты новостей или написание кода на Python. Подробнее смотрите *Добавление любимых новостных веб-сайтов* (страница 33).

3. **Загрузить все запланированные новостные источники**: заставляет calibre немедленно начать загрузку всех запланированных источников новостей.

1.1.7 Библиотека

Действие *Библиотека* позволяет создавать, переключаться, переименовывать или удалять библиотеки. calibre позволяет создать сколько угодно библиотек. Например, художественную, научно-популярную, на иностранных языках, библиотеку проекта или любую структуру, соответствующую вашим потребностям. Библиотеки — организационная структура высшего порядка в calibre. В каждой библиотеке свой набор книг, тегов, категорий и основное место хранения.

1. **Переключить/создать библиотеку**: позволяет: а) подключиться к уже существующей библиотеке calibre в другом месте, б) создать пустую библиотеку на новом месте или в) переместить текущую библиотеку на указанное новое место.

2. **Быстрое переключение**: позволяет переключаться между библиотеками, зарегистрированными или созданными в calibre.

3. **Переименовать библиотеку**: позволяет переименовать библиотеку.

4. **Выбрать случайную книгу**: выбирает для вас случайную книгу из библиотеки

5. **Удалить библиотеку**: позволяет отменить регистрацию библиотеки в calibre.

6. **Экспорт/импорт всех данных calibre**: позволяет либо экспортировать данные calibre для переноса на новый компьютер, либо импортировать экспортированные ранее данные.

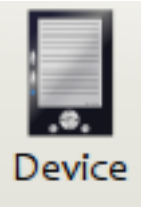
7. **<library name>**: действия 7, 8 и т. д. дают возможность мгновенно переключаться между несколькими созданными или присоединёнными библиотеками. В этом списке только 5 наиболее часто используемых библиотек. Полный список см. в меню «Быстрое переключение».

8. **Обслуживание библиотеки**: позволяет проверить наличие проблем соответствия данных в текущей библиотеке и восстановить её базу данных из резервных копий.

> **Примечание**
>
> Метаданные о ваших электронных книгах, например название, автор и теги, хранятся в одном файле в папке библиотеки calibre под названием metadata.db. Если этот файл повреждён (очень редкое событие), вы можете потерять метаданные. К счастью, calibre автоматически резервирует метаданные для каждой отдельной книги в их папках в виде файла OPF. С помощью вышеописанного действия «Восстановить базу данных» из «Обслуживания библиотеки», вы можете настроить восстановление файла metadata.db из отдельных файлов OPF.

Вы можете копировать или перемещать книги между разными библиотеками (если у вас настроено более одной), нажав правой кнопкой мыши на книгу и выбрав действие *Копировать в библиотеку*.

1.1.8 Устройство



Действие *Устройство* позволяет просматривать книги в основной памяти или на карте памяти вашего устройства или извлечь устройство (отключить от calibre). Значок автоматически появляется на главной панели инструментов calibre при подключении поддерживаемого устройства. Нажав на него, вы увидите книги на вашем устройстве. Также можно перетащить книги из библиотеки calibre на этот значок, чтобы перенести их на устройство. И наоборот, вы можете перетащить книги с вашего устройства на значок библиотеки на панели инструментов, чтобы перенести их с вашего устройства в библиотеку calibre.

1.1.9 Сохранить на диск

Действие *Сохранить на диск* имеет пять вариантов, которые доступны по правому щелчку мышью на кнопке.

1. **Сохранить на диск**: сохраняет выбранные книги на диск, организованные в папки. Структура папок выглядит так:

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

Вы можете управлять именем файла и структурой папок файлов, сохранённых на диске, настроив шаблон в меню *Параметры → Импорт-экспорт → Сохранение книг на диск*. Также см. *Язык шаблонов calibre* (страница 164).

2. **Сохранить на диск в одной папке**: сохраняет выбранные книги на диск в одной папке.

1 и 2 пункты сохраняют все доступные форматы и метаданные на диск для каждой выбранной книги. Метаданные хранятся в файле OPF. Сохранённые книги можно импортировать в библиотеку ещё раз без потери какой-либо информации с помощью действия *Добавить книги* (страница 4).

3. **Сохранить только *<your preferred>* формат на диск**: сохраняет выбранные книги на диск в структуре папок, как показано в (1.), но только в предпочитаемом вами формате электронной книги. Вы можете установить предпочтительный формат в *Preferences-Interface-Behavior-Preferred output format*

4. **Сохранить только *<your preferred>* формат на диск в одной папке**: сохраняет выбранные книги на диск в одной папке, но только в предпочитаемом вами формате электронной книги. Вы можете установить предпочтительный формат в *Preferences-Interface-Behavior-Preferred output format*

5. **Сохранить один формат на диск…**: сохраняет выбранные книги на диск в структуре папок, как показано в (1.), но только в формате, который вы выбираете из всплывающего списка.

1.1.10 Подключить/поделиться

Действие *Подключить/поделиться* позволяет вручную подключиться к устройству или папке на компьютере, а также настроить доступ к вашей библиотеке calibre через веб-браузер или электронную почту.

Действие *Подключить/поделиться* имеет четыре варианта, которые доступны по правому щелчку мышью по кнопке.

1. **Подключиться к папке**: позволяет подключиться к любой папке на компьютере как к устройству и использовать в ней все возможности calibre для устройств. Полезно, если ваше устройство не поддерживается calibre, но доступно как диск USB.
2. **Запустить контент-сервер**: запускает встроенный веб-сервер calibre. Когда он запущен, ваша библиотека calibre доступна через веб-браузер из интернета (по вашему выбору). Вы можете настроить способ доступа к веб-серверу, установив настройки в меню *Параметры → Общий доступ к файлам → Сетевой доступ*
3. **Настроить обмен книгами по электронной почте**: позволяет обмениваться книгами и новостными лентами по электронной почте. После настройки адресов электронной почты calibre будет отправлять на них обновления новостей и книг. Настроить, как calibre посылает электронную почту, можно в меню *Параметры → Общий доступ к файлам → Пересылка книг по электронной почте*. Когда вы настроите один или несколько адресов электронной почты, этот пункт меню изменится на отправку книг по заданным адресам.

1.1.11 Удалить книги

Действие *Удалить книги* **безвозвратно удаляет книги**, поэтому используйте его с осторожностью. Оно *зависит от контекста*, т. е. от выбранного *каталога* (страница 11). Если вы выбрали *Библиотека*, книги будут удалены из библиотеки. Если вы выбрали устройство для чтения электронных книг, книги будут удалены с устройства. Чтобы удалить только определённый формат для данной книги, используйте действие *Редактировать метаданные* (страница 5). Удаление книг также имеет пять вариантов, к которым можно получить доступ правым щелчком по кнопке.

1. **Удалить выбранные книги**: позволяет вам **безвозвратно** удалить все книги, выделенные в списке книг.
2. **Удалить файлы определённого формата из выбранных книг…**: позволяет **безвозвратно** удалить файлы электронных книг указанного формата из выделенных в списке книг.
3. **Удалить все форматы из выбранных книг, исключая…**: позволяет **безвозвратно** удалить файлы электронных книг любого формата, кроме указанного, из выделенных в списке книг.
4. **Удалить все форматы из выбранных книг**: позволяет **безвозвратно** удалить все файлы электронных книг, выбранных в списке. Останутся только метаданные.
5. **Удалить обложки из выбранных книг**: позволяет **безвозвратно** удалить файлы обложек из выделенных в списке книг.

6. **Удалить совпадающие книги с устройства**: позволяет удалить с подключённого устройства файлы электронных книг, совпадающие с выделенными в списке книгами.
7. **Восстановить недавно удаленные**: позволяет отменить удаление книг или форматов.

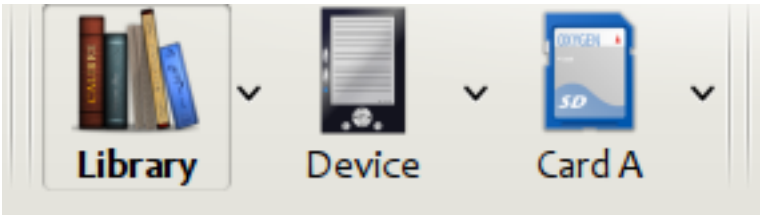
> **Примечание**
>
> Обратите внимание, что когда вы используете *Удалить книги* для удаления книг из библиотеки caliber, запись о книге удаляется, но книги временно хранятся в течение нескольких дней в папке корзины. Вы можете отменить удаление, щёлкнув правой кнопкой мыши по кнопке *Удалить книги* и выбрав *Восстановить недавно удалённые* книги.

1.2 Параметры

Действие *Параметры* позволяет вам изменить различные аспекты работы calibre. Имеет четыре варианта, доступные по правому щелчку мышью по кнопке.

1. **Параметры**: позволяет изменить различные аспекты работы calibre. Нажатие кнопки также выполняет это действие.
2. **Запустить мастер начальной настройки**: позволяет запустить *Мастер начальной настройки*, который появлялся при первом запуске calibre.
3. **Получить плагины для расширения calibre**: открывает новое окно, которое показывает плагины для calibre. Эти плагины создаются сторонними разработчиками для расширения функциональности calibre.
4. **Перезапустить в режиме отладки**: позволяет включить отладочный режим, который помогает разработчикам calibre в решении проблем, с которыми вы столкнулись. Для большинства пользователей он должен оставаться выключенным, если только разработчиком не дано соответствующее указание.

1.3 Каталоги



Каталог — это коллекция книг. calibre может управлять двумя типами различных каталогов:

1. **Библиотека**: это коллекция книг, хранящаяся в библиотеке calibre на вашем компьютере.
2. **Устройство**: это коллекция книг, хранящихся на вашем устройстве для чтения электронных книг. Она становится доступной при подключении устройства к компьютеру.

Многие операции, такие как добавление книг, удаление, просмотр и т. д., зависят от контекста. Например, если вы нажмёте кнопку *Просмотр*, когда выбран каталог **Устройство**, calibre откроет файлы с устройства. Если у вас выбран каталог **Библиотека**, то будут открываться файлы из вашей библиотеки calibre.

1.4 Поиск и сортировка

Раздел «Поиск и сортировка» позволяет выполнять несколько эффективных действий с вашими коллекциями книг.

- Вы можете сортировать их по названию, автору, дате, оценке и т. д., нажимая на заголовки столбцов. Также можно сортировать по нескольким столбцам. Например, если вы нажмёте на столбец «Название» а затем на столбец «Автор», книги будут отсортированы по автору, а потом все записи с одинаковым автором будут отсортированы по названию.
- Вы можете искать определённую книгу или набор книг с помощью строки поиска. Подробнее об этом ниже.
- Вы можете быстро и удобно редактировать метаданные, выбрав запись, которую вы хотите изменить, в списке и нажав клавишу `E`.
- Вы можете выполнять *Действия* (страница 4) с наборами книг. Чтобы выделить несколько книг, нужно:
  - удерживая клавишу `Ctrl`, нажимать на книги, которые хотите выбрать;
  - удерживая клавишу `Shift`, нажать на первую и последнюю книги из ряда книг, который хотите выбрать.
- Вы можете настроить, какие поля будут отображаться, в диалоговом окне *Параметры* (страница 11).
- Чтобы выполнить сложную подсортировку по нескольким столбцам, добавьте инструмент *Сортировать по* на панель инструментов через *Настройки-Панели инструментов и меню*.

1.5 Интерфейс поиска

Вы можете искать все метаданные, введя условия поиска в строку поиска. Например:

```
Asimov Foundation format:lrf
```

Это будет соответствовать всем книгам в вашей библиотеке, которые содержат в метаданных `Asimov` и `Foundation` и доступны в формате LRF. Ещё несколько примеров:

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

Виды поиска

Существует четыре вида поиска: *contains*, *equality*, *regular expression* (см. **`регулярные выражения https://en.wikipedia.org/wiki/Regular_expression`_**) и *character variant*. Выбор вида поиска - символом префикса.

„Contains" поиски

Поиски без символа префикса - *Содержит* и по умолчанию нечувствительны к регистру. Элемент соответствует, если строка поиска появляется где-либо в указанных метаданных. Можно сделать поиск *Содержит* чувствительным к регистру, установив флажок *Поиск с учетом регистра* в *Настройки/Поиск*. Если отмечена опция поиска «Символы без ударения соответствуют символам с диакритическими знаками, а знаки препинания игнорируются», то символ будет соответствовать всем его вариантам (например, *e* соответствует *é*, *è*, *ê* и *ë*) и все знаки препинания, включая пробелы, игнорируются. Например, если установлен флажок «Символы без диакритических знаков соответствуют …», то для двух названий книг:

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

то эти поиски находят:

- `title:"er"` совпадает с обоими („e" совпадает с обоими „é" и „e").
- `title:"g "` соответствует обоим, т.к. пробелы игнорируются.
- `title:"g,"` соответствует обоим, т.к. запятая игнорируется.
- `title:"gb"` соответствует обоим, т.к. „, „ игнорируются в книге 1, а пробелы игнорируются в книге 2.
- `title:"g b"` соответствует обоим, потому что запятая и пробел игнорируются.
- `title:"db"` соответствует #1, потому что пробел в „and Bad" игнорируется.
- `title:","` соответствует обоим (фактически соответствует всем книгам), потому что запятые игнорируются.

Если параметр *Символы без диакритических знаков соответствуют …* не отмечен, то варианты символов, знаки препинания и пробелы значимы.

Вы можете установить только один из параметров поиска: *Поиск с учетом регистра* и *Символы без диакритических знаков соответствуют символам с диакритическими знаками, а знаки препинания игнорируются*.

„Equality" поиски

Поиск равенства обозначается префиксом строки поиска со знаком равенства (=). Например, запрос `tag:"=science"` будет соответствовать *science*, но не *science fiction* или *hard science*. Варианты символов значимы: *é* не соответствует *e*.

Два варианта поиска равенства используются для иерархических элементов (например, A.B.C): иерархический поиск префикса и иерархический поиск компонентов. Первый, обозначенный точкой после равенства (`=.`), соответствует начальным частям иерархического элемента. Второй, обозначенный двумя точками после знака равенства (`=..`), соответствует внутреннему имени в иерархическом элементе. Примеры использования тега `History.Military.WWII` в качестве значения:

- `tags:"=.History"` : True. `History` префикс тега.
- `tags:"=.History.Military"` : True. `History.Military` префикс тега.

- `tags:"=.History.Military.WWII"` : True. `History.Military.WWII` префикс тега, хоть и неподходящий.
- `tags:"=.Military"` : False. `Military` не префикс тега.
- `tags:"=.WWII"` : False. `WWII` не префикс тега.
- `tags:"=..History"` : True. Иерархия содержит значение `History`.
- `tags:"=..Military"` : True. Иерархия содержит значение `Military`.
- `tags:"=..WWII"` : True. Иерархия содержит значение `WWII`.
- `tags:"=..Military.WWII"` : False. `..` поиск ищет одиночные значения.

„Regular expression" поиски

Поиск с использованием регулярных выражений обозначается префиксом строки поиска с помощью тильды (~). Можно использовать любое Python-совместимое регулярное выражение[1]. Обратная косая черта, используемая для экранирования специальных символов в регулярных выражениях, должна быть удвоена, так как одиночная обратная косая черта будет удалена при синтаксическом анализе запроса. Например, чтобы сопоставить литеральную круглую скобку, вы должны ввести``\(``. или, альтернативно, используйте `супер-кавычки` (см. ниже). Поиск по регулярным выражениям является поиском 'contains', если только выражение не привязано. Варианты символов значимы: ``~e`` не соответствует é.

„Character variant" поиски

Поиск вариантов символов обозначается префиксом строки поиска с символом вставки (^). Этот поиск аналогичен поиску *contains* (выше), за исключением того, что:

- регистр букв всегда игнорируется.
- варианты символов всегда соответствуют друг другу.
- знаки препинания и пробелы всегда значимы.

Параметры поиска *Символы без ударения соответствуют знакам с диакритическими знаками, а знаки препинания игнорируются* и *Поиск с учетом регистра* игнорируются. Они не влияют на поведение этого поиска.

Далее этот поиск сравнивается с поиском по содержанию, предполагая, что опция *Совпадение символов без диакритических знаков…* включена (см. выше) для двух одинаковых названий книг:

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

то эти поиски варианты символов находят:

- `title:"^er"` соответствует обоим („e" соответствует обоим „é" и „e")
- `title:"^g"` соответствует обоим
- `title:"^g "` соответствует #2, т.к. пробел значим
- `title:"^g,"` соответствует #1, т.к. запятая значима
- `title:"^gb"` ничего не соответствует, т.к. пробел и запятая значимы
- `title:"^g b"` соответствует #2, т.к. запятая значима
- `title:"^db"` ничего не соответствует
- `title:"^,"` соответствует #1 (а не всем книгам), т.к. запятая значима

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

Синтаксис поискового выражения

«Выражение поиска» представляет собой последовательность «терминов поиска», необязательно разделенных операторами `and` и `or`. Если два поисковых запроса встречаются без разделительного оператора, подразумевается `и`. Оператор `and` имеет приоритет над оператором `or`; например, выражение `a or b and c` такое же, как `a or (b and c)`. Вы можете использовать круглые скобки для изменения приоритета; например `(a or b) and c`, чтобы `or` оценивалось перед `and`. Вы можете использовать оператор `not` для отрицания (инвертирования) результата вычисления поискового выражения. Примеры:

- `not tag:foo` находит все книги, не содержащие тега `foo`
- `not (author:Asimov or author:Weber)` находит все книги, написанные не Азимовым или Вебером.

В приведенных выше примерах показаны примеры «условий поиска». Базовый «термин поиска» — это последовательность символов, не содержащая пробелов, кавычек (`"`), обратной косой черты (`\\`) или скобок (`( )`). Ему может предшествовать спецификатор имени столбца: *имя поиска* столбца, за которым следует двоеточие (`:`), например, `author:Asimov`. Если поисковый запрос должен содержать пробел, то весь термин должен быть заключен в кавычки, например, `title:"The Ring"`. Если поисковый запрос должен содержать кавычки, они должны быть экранированы обратной косой чертой. Например, чтобы найти серию под названием *The «Ball» and The «Chain»*, используйте:

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

Если вам нужна обратная косая черта, что часто случается при поиске с *регулярными выражениями*, используйте две из них (`\\`).

Иногда бывает трудно правильно подобрать все escape-последовательности, чтобы получить желаемый результат, особенно при поиске с *регулярными выражениями* и *по шаблону*. В этих случаях используйте *супер-кавычки*: `"""sequence of characters"""`. Символы в суперкавычках используются без изменений: обработка экранирования не выполняется.

Дополнительная информация

Для поиска строки, которая начинается с знака равенства, тильды или знака каретки; предварите строку обратной косой чертой (backslash).

Заключайте строки поиска в кавычки («), если они содержат скобки или пробелы. Например, для поиска по тегу `Science Fiction` нужно будет искать `tag:"=science fiction"`. Если вы ищете `tag:=science fiction`, вы найдёте все книги с тегом `science`, содержащие слово `fiction` в любых метаданных.

Вы можете легко создавать расширенные поисковые запросы с помощью окна *Расширенный поиск*, открывающегося нажатием кнопки .

Доступные поля для поиска: `tag`, `title`, `author`, `publisher`, `series`, `series_index`, `rating`, `cover`, `comments`, `format`, `identifiers`, `date`, `pubdate`, `search`, `size`, `vl` и пользовательские столбцы. Если устройство подключено, при поиске в библиотеке calibre становится доступно поле `ondevice`. Чтобы найти имя для поиска (которое на самом деле называется *lookup name*) для пользовательского столбца, наведите мышь на заголовок столбца в библиотеке.

Dates

Синтаксис поиска по датам:

```
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
```

Если дата неясна, для сравнения дат используется текущая локаль. Например, в локали мм/дд/гггг 2/1/2009 интерпретируется как 1 февраля 2009 года. В локали дд/мм/гггг та же дата интерпретируется как 2 января 2009

года. Доступны некоторые специальные строки дат. Строка `today` переводится в сегодняшнюю дату, какая бы она ни была. Строки `yesterday` и `thismonth` (или переведённый эквивалент на текущем языке) также работают. В добавление, строка `daysago` (также и переведённая) может использоваться для сравнения с датой, которая была несколько дней назад. Например:

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

Во избежание потенциальных проблем с переведёнными строками при использовании не английской версии calibre, строки `_today`, `_yesterday`, `_thismonth`, and `_daysago` всегда доступны. Они не переводятся.

Поиск дат и числовых значений с реляционным сравнением

Даты и числовые поля поддерживают операторы сравнения: = (равно), > (больше), >= (больше или равно), < (меньше), <= (меньше или равно) и != (не равно). Поля оценки считаются числовыми. Например, поиск `rating:>=3` найдёт все книги с оценкой 3 или выше.

Вы можете искать книги, которые имеют формат определённого размера, например:

- `size:>1.1M` найдёт книги размером более 1,1 МБ.
- `size:<=1K` найдет книги с размером меньше или равным 1 КБ

Вы можете искать по числу элементов в полях с несколькими значениями, такими как теги. Этот поиск начинается с символа `#`, потом используйте такой же синтаксис, как и для числовых полей. Например, чтобы найти книги с пятью и более тегами используйте `tags:#>4`. Для поиска всех книг с 10 тегами используйте `tags:#=10`.

Индексы серий

Индексы серии доступны для поиска. Для стандартной серии имя для поиска - `series_index`. Для пользовательски столбцов серии используйте имя поиска столбца с приставкой _index. Например, для поиска индексов для пользовательского столбца серии с названием `#my_series` вы могли бы использовать имя `#my_series_index`. Индексы серии - числа, так что вы можете использовать операторы отношений, описанные выше.

Сохранённые поиски

Специальное поле `search` используется для *saved searches* (страница 18). Поэтому, если вы сохраните поиск с именем «My spouse's books», вы можете ввести `search:"My spouse's books"` в строке поиска для повторного использования сохранённого поиска. Подробнее о сохранении запросов ниже.

Виртуальные библиотеки

Специальное поле `vl` используется для поиска книг в виртуальной библиотеке. Например, `vl: Read` найдет все книги в виртуальной библиотеке *Read*. Поиск `vl: Read and vl: «Science Fiction»` найдет все книги, которые находятся в виртуальных библиотеках *Read* и *Science Fiction*. Значение, следующее за `vl:`, должно быть именем виртуальной библиотеки. Если имя виртуальной библиотеки содержит пробелы, то окружайте его кавычками.

Имеет ли поле значение

Вы можете искать по наличию или отсутствию поля, используя специальные значения «true» и «false». Например:

- `cover:false` находит все книги без обложки
- `series:true` находит все книги из серии
- `series:false` находит все книги, не входящие в серию
- `comments:false` находит все книги с пустым комментарием
- `formats:false` находит все книги без файлов книг (пустые записи)

Yes/no пользовательские столбцы

Пользовательские столбцы да/нет доступны для поиска. Поиск по `false`, `empty`, или `blank` найдёт все книги с неопределёнными значениями в столбце. Поиск по `true` найдёт все книги, которые не имеют неопределённых значений в этом столбце. Поиск по `yes` или `checked` найдёт все книги с `Yes` в этом столбце. Поиск по `no` или `unchecked` найдёт все книги с `No` в этом столбце. Заметьте, что слова `yes`, `no`, `blank`, `empty`, `checked` and `unchecked` переводятся; вы можете использовать как эквивалент на текущем языке, так и английское слово. Слова `true` и `false` и специальные значения `_yes`, `_no`, и `_empty` не переводятся.

Идентификаторы

Идентификаторы (например, ISBN, DOI, LCCN и т. д.) также используют расширенный синтаксис. Идентификатор имеет форму `type:value`, как в `isbn:123456789`. Расширенный синтаксис позволяет вам независимо указывать тип и значение для поиска. И в части типа, и в значении запроса могут использоваться *search kinds* (страница 13). Примеры:

- `identifiers:true` будет искать книги с любым идентификатором.
- `identifiers:false` будет искать книги без идентификатора.
- `identifiers:123` будет искать книги с любым типом, имеющим значение содержащее *123*.
- `identifiers:=123456789` будет искать книги с любым типом, имеющим значение равное *123456789*.
- `identifiers:=isbn:` и `identifiers:isbn:true` найдут книги с типом, равным ISBN с любым значением
- `identifiers:=isbn:false` будет искать книги без эквивалентного ISBN типа.
- `identifiers:=isbn:123` будет искать книги с типом эквивалентным ISBN содержащим *123*.
- `identifiers:=isbn:=123456789` будет искать книги с типом, эквивалентным ISBN равным *123456789*.
- `identifiers:i:1` будет искать книги с типом содержащим *i* и имеющим значение содержащее *1*.

Категории видимые в Браузере тегов

Поиск `in_tag_browser:true` находит все книги, находящихся в категориях (теги, авторы и т. д.), отображаемых в данный момент в *Браузер тегов*. Это полезно, если вы установили два параметра: *Настройки → Оформление → Браузер тегов → Скрыть пустые категории (столбцы)* и *Найти и показать все соответствующие элементы*. С этими двумя настройками выполнение *Найти* в *Просмотр тегов* показывает только категории, содержащие элементы, соответствующие *Найти*. Затем поиск `in_tag_browser:true` дополнительно находит книги с этими категориями/элементами. Вы можете легко запустить этот поиск, нажав клавишу `Ctrl+Alt+Shift+F` или щелкнув кнопку настройки в *Браузер тегов* и выбрав пункт *Показывать только книги с видимыми категориями* .

Поиск по шаблонам

Вы можете искать, используя шаблон в *Язык шаблонов calibre* (страница 164) вместо поля метаданных. Для этого вы вводите шаблон, тип поиска и значение для поиска. Синтаксис:

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

`template` — это любой допустимый шаблон на языке шаблонов calibre. `search type` должен быть одним из `t` (текстовый поиск), `d` (поиск по дате), `n` (числовой поиск) или `b` (set/not set (логическое значение)). `value` может быть любым, которое вы хотите, и вы можете использовать *search kinds* (страница 13), описанные выше для различных типов поиска. Вы должны обрамить кавычками всю строку поиска, если в ней есть пробелы.

Примеры:

- `template:"program: connected_device_name('main')#@#:t:kindle"` – true когда подключено устройство `kindle`.
- `template:"program: select(formats_sizes(), 'EPUB')#@#:n:>1000000"` – найти книги с файлами EPUB размером более 1 МБ.

- `template:"program: select(formats_modtimes('iso'), 'EPUB')#@#:d:>10daysago"` – найти книги с EPUB файлами новее чем 10 дней тому назад
- `template:'''program: book_count('tags:^'' & $series & ''', 0) != 0#@#:n:1'''` – находит все книги, содержащие серию имя в тегах. В этом примере используются суперкавычки, так как шаблон использует как одинарные (`'`), так и двойные кавычки (`"`) при построении выражения поиска.

Вы можете легко создавать шаблоны поисковых запросов с помощью диалога *Расширенный поиск*, нажав кнопку . Можно протестировать шаблоны на определённых книгах, используя *Тестер шаблонов* calibre. Его можно добавить на панели инструментов или в меню через *Параметры → Панели инструментов и меню*. Ему также можно назначить сочетание клавиш через *Параметры → Комбинации клавиш*.

Диалог расширенного поиска

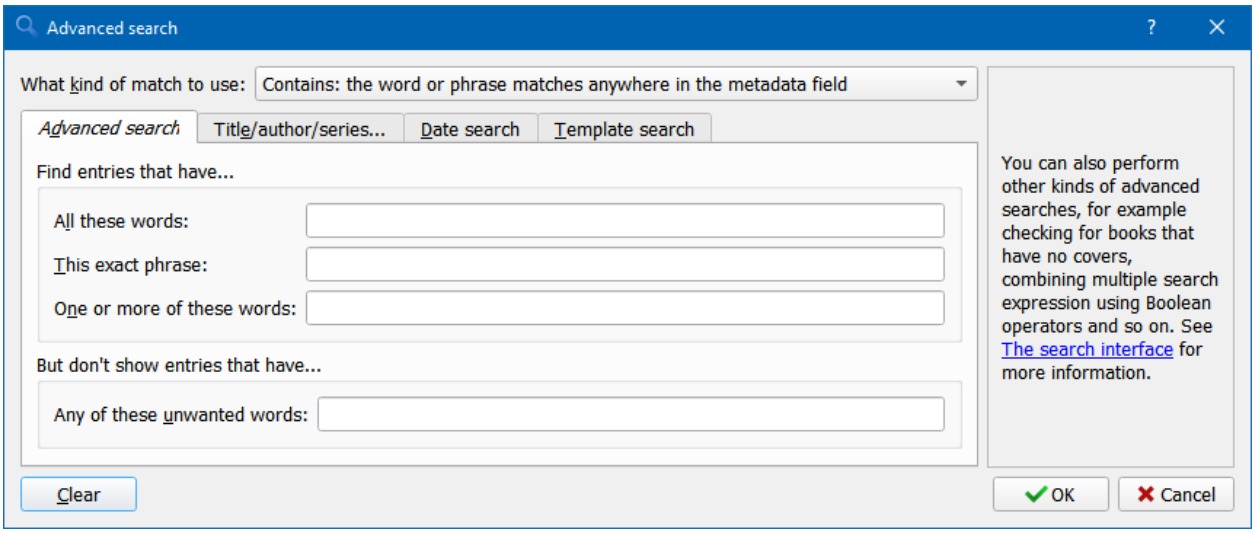


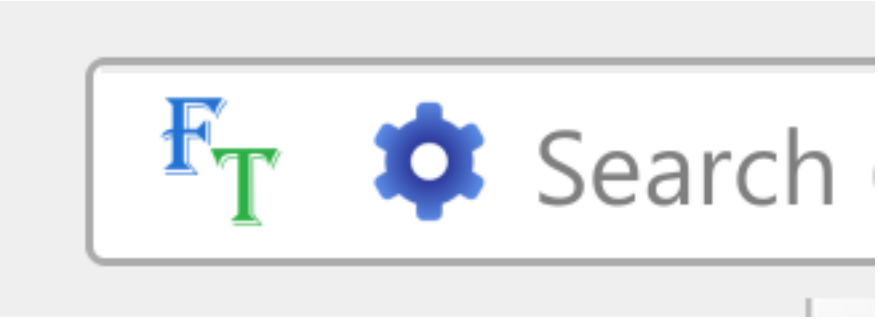
Рис. 1: Диалоговое окно *Расширенный поиск*

1.6 Сохранение поисковых строк

calibre позволяет вам сохранить часто используемый поиск под специальным именем, а затем повторно использовать этот поиск одним щелчком мыши. Для этого создайте свой поиск, набрав его на панели поиска или с помощью браузера тегов. Затем введите имя, которое вы хотели бы дать поиску в поле «Сохраненные поиски» рядом с панелью поиска. Нажмите значок «плюс» рядом с сохраненным полем поиска, чтобы сохранить поиск.

Теперь вам доступен сохранённый поиск в браузере тегов в разделе «*Сохранённые поиски*». Один щелчок позволит с лёгкостью использовать любой произвольно сложный запрос без необходимости его пересоздавать.

1.7 Поиск по полному тексту всех книг



calibre может *опционально* индексировать полный текст книг в библиотеке, чтобы обеспечить мгновенный поиск слов внутри любой книги. Чтобы использовать эту функцию, нажмите кнопку *FT* у левого края строки поиска.

Затем включите индексирование для текущей библиотеки. После завершения индексации вы можете искать весь текст в полной библиотеке. Когда вы добавляете новые книги в библиотеку, они автоматически индексируются в фоновом режиме. Этот поиск позволяет быстро найти все книги, содержащие слово или комбинацию слов. Вы даже можете искать слова, которые встречаются рядом с другими словами, как показано в примерах во всплывающем окне поиска. Обратите внимание, что этот инструмент поиска найдет только одно вхождение поискового запроса в конкретной книге, а не список всех вхождений, для этого лучше всего искать внутри книги с помощью calibre *E-book viewer*.

Вы можете переиндексировать отдельную книгу, щёлкнув правой кнопкой мыши на панели сведений о книге в calibre и выбрав *Переиндексировать эту книгу для полнотекстового поиска*

1.8 Виртуальные библиотеки

Виртуальная библиотека - это способ притвориться, что в вашей библиотеке calibre всего несколько книг вместо полной коллекции. Это отличный способ разбить вашу большую коллекцию книг на более мелкие, управляемые куски. Чтобы узнать, как создавать и использовать виртуальные библиотеки, см. Учебное пособие: *Виртуальные библиотеки* (страница 255).

1.9 Временно пометить книги

Вы можете временно отмечать произвольные наборы книг. Отмеченные книги будут отмечены булавкой, и их можно будет найти с помощью поиска `marked:true`. Чтобы отметить книгу, нажмите `Ctrl+M` или перейдите в *Настройки-Панели инструментов и меню* и добавьте кнопку *Отметить книги* на главную панель инструментов.

Вы можете пометить книги определенной текстовой меткой, щелкнув правой кнопкой мыши кнопку *Отметить книги* и выбрав *Пометить книги текстовой меткой*. Книги, отмеченные текстовыми метками, позже можно будет найти с помощью поиска `marked:"=the-text-you-entered"`.

1.10 Предположение метаданных на основе имён файлов

Обычно calibre считывает метаданные из файла книги. Однако вместо этого программу можно настроить для чтения метаданных из имени файла через *Параметры → Импорт-экспорт → Добавление книг → Считывать метаданные из содержимого файла*.

Вы также можете управлять чтением метаданных из имени файла с помощью регулярных выражений (см. *Всё об использования регулярных выражений в calibre* (страница 221)). В разделе *Добавление книг* диалогового окна конфигурации вы можете указать регулярное выражение, которое calibre будет использовать, чтобы попытаться

угадать метаданные из имен файлов электронных книг, которые вы добавляете в библиотеку. Регулярное выражение по умолчанию:

```
title - author
```

то есть предполагается, что все символы до первого «-» (дефиса) — название книги, а все последующие — автор. Например, имя файла:

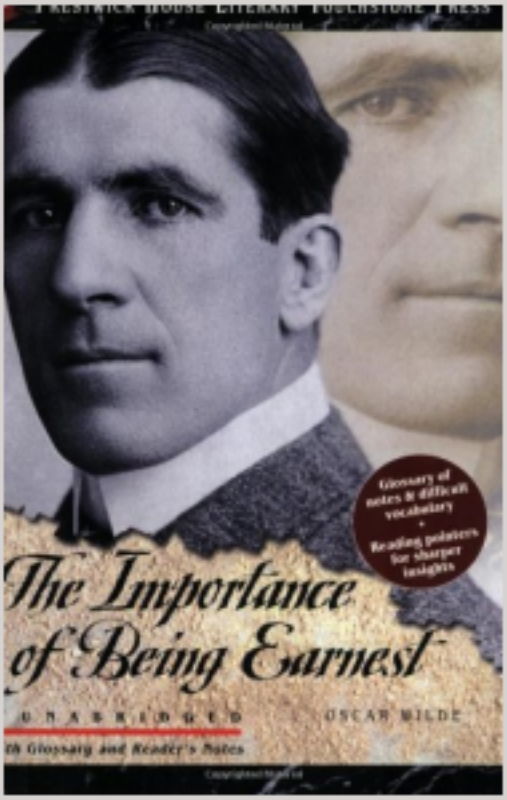
```
Foundation and Earth - Isaac Asimov.txt
```

будет интерпретировано как название «Foundation and Earth» и автор «Isaac Asimov».

Совет

Если в имени файла нет дефиса, вышеуказанное регулярное выражение не сработает.

1.11 Описание книги



Панель «Описание книги» показывает обложку и все метаданные выделенной в данный момент книги. Её можно

скрыть с помощью кнопки *Макет* в правом нижнем углу главного окна calibre. Имена авторов на панели описания книги кликабельны, и по умолчанию ведут на страницу автора в Википедии. Также их можно настроить, щёлкнув правой кнопкой мыши имя автора и выбрав «Управление этим автором».

Аналогично, если вы загружаете метаданные, панель описания книги автоматически покажет ссылки на веб-страницы Amazon, Worldcat и т. д., с которых они загружены.

Вы можете щёлкнуть правой кнопкой мыши по отдельным форматам на панели описания книги, чтобы удалить их, сравнить с оригинальными версиями, сохранить на диск, открыть с помощью внешней программы и т. д.

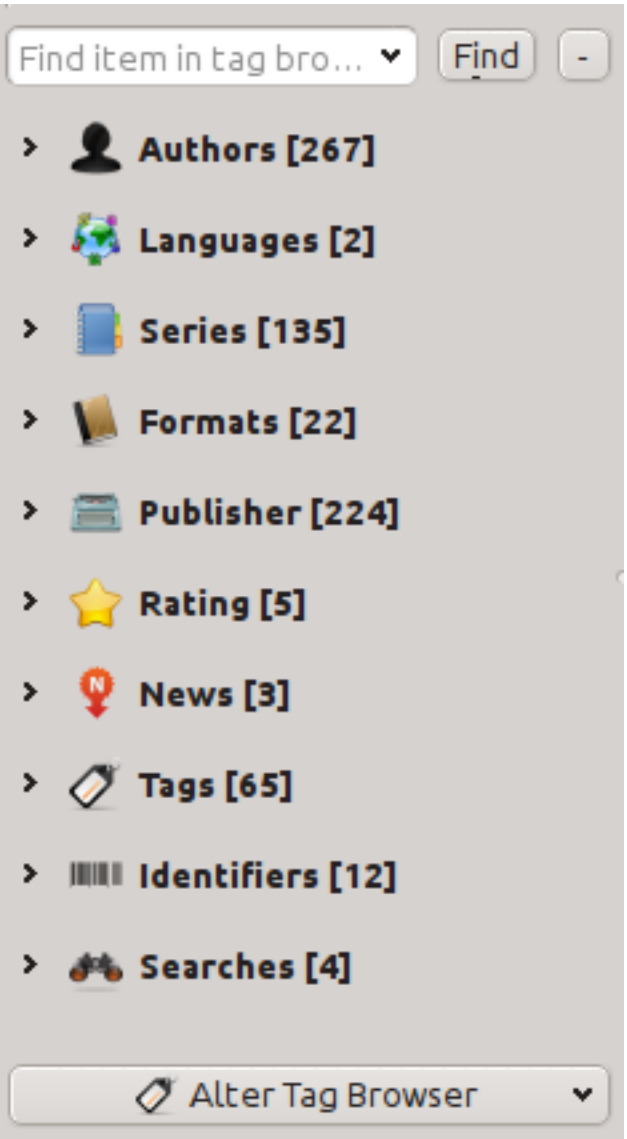
Вы можете изменить обложку книги, просто перетащив изображение на панель описания книги. Если вы хотите отредактировать обложку во внешней программе, просто щёлкните её правой кнопкой мыши и выберите *Открыть с помощью…*.

Вы также можете добавить файлы электронных книг в текущую книгу, перетащив их на панель описания книги.

Двойной щелчок на панели «Описание книги» откроет её в отдельном всплывающем окне.

Наконец, вы можете точно настроить, какая информация отображается на панели описания книги через *Параметры → Интерфейс → Оформление → Описание книги*.

1.12 Браузер тегов



Браузер тегов позволяет легко просматривать вашу коллекцию по автору/тегам/сериям/ и т. д. Если вы нажмёте любой элемент в браузере тегов, например, имя автора «Айзек Азимов», то список книг справа покажет только книги этого автора. Также можно нажимать на названия категорий. Например, нажатие «Серии» покажет вам все книги любых серий.

Первое нажатие на элемент ограничит список книг теми, которые содержат или соответствуют элементу. Продолжая вышеописанный пример, щёлкнув «Айзек Азимов», вы увидите книги этого автора. При повторном нажатии список изменится в зависимости от того, имеются ли дочерние элементы (см. подкатегории и иерархические элементы ниже). Вернёмся к примеру с Азимовым: повторное нажатие на «Айзек Азимов» ограничит список книгами, не принадлежащими Айзеку Азимову. Третье нажатие снимет ограничение, показав все книги. Если вы, удерживая клавиши `Ctrl` или `Shift`, нажимаете на несколько элементов, то создаются ограничения на основе нескольких элементов. Например, вы можете удерживать `Ctrl` и нажимать на теги «история» и «Европа» для поиска книг по истории Европы. Браузер тегов работает, создавая выражения поиска, которые автоматически вводятся в строку поиска. Наблюдать за тем, что генерирует браузер тегов — хороший способ научиться строить основные поисковые выражения.

Значки элементов в браузере тегов частично окрашены. Количество цвета зависит от средней оценки книг в категории. Так, например, если у книг Айзека Азимова в среднем четыре звезды, то его значок в браузере тегов будет окрашен на 4/5. Вы можете увидеть среднюю оценку, наведя указатель мыши на значок.

Элементы внешнего уровня в *Tag browser „ такие как Авторы и Серии, называются категориями. Вы можете создать свои собственные категории, называемые guilabel :`Категории пользователей*, это удобно для организации элементов. Например, можно использовать :guilabel: «Редактор категорий пользователей» (нажмите кнопку «guilabel:*Configure» в левом нижнем углу :guilabel:«Браузер тегов» и выберите :guilabel:«Управление авторами, сериями, и т.д. -> Категории пользователей*) для создания категории пользователей под названием «Любимые авторы», а затем поместите элементы для избранных в категорию. Категории пользователей могут иметь подкатегории. Например, категория пользователя `Favorites.Authors` является подкатегорией ``Favorites`. У вас также может быть `Favorites.Series`, и в этом случае под `Favorites` будут две подкатегории. Подкатегории можно создать, щелкнув правой кнопкой мыши по категории «Пользователь», выбрав :guilabel: «Добавить подкатегорию в …» и ввести имя подкатегории; Или с помощью :guilabel: «Редактор категорий пользователей», введя имена, подобные приведенному выше.

Вы можете искать категории пользователей так же, как и встроенные категории, нажав на них. Вы можете выполнить четыре разных запроса, нажав:

1. «всё соответствующее элементу в этой категории» отмечено одним зелёным знаком плюс.
2. «всё соответствующее элементу в этой категории или подкатегориях» отмечено двумя зелёными знаками минус.
3. «всё несоответствующее элементу в этой категории» показывается одним красным знаком минус.
4. «всё несоответсвующее элементу в категории или подкатегориях» показано двумя красными знаками минус.

Можно также создать иерархии внутри некоторых текстовых категорий, таких как теги, серии и пользовательские столбцы. Эти иерархии показываются с маленьким треугольником, позволяя скрыть подпункты. Чтобы использовать иерархии элементов в категории, вы должны сначала перейти к *Настройки → Интерфейс → Look & feel* и ввести названия категории в поле «Категории с иерархическими элементами». Как только это будет сделано, элементы в этой категории, которые содержат точки, будут показаны с помощью маленького треугольника. Например, предположим, что вы создали пользовательский столбец с именем «Жанр» и указали, что он содержит иерархические элементы. После этого такие элементы, как Mystery.Thriller и Mystery.English, будут отображаться как Mystery с небольшим треугольником рядом с ним. Нажатие на треугольник покажет триллер и английский в качестве подпунктов. Смотрите *Управление подгруппами книг, например «жанр»* (страница 157) для получения дополнительной информации.

Иерархические элементы (элементы с потомками) используют те же четыре «щелчка», что и категории пользователей. Элементы, у которых нет потомков, используют два запроса: «всё совпадающее» и «всё не совпадающее».

Вы можете перетаскивать элементы в браузере тегов в Пользовательские категории, чтобы добавить их в эту категорию. Если источник - категория пользователя, с удержанием клавиши shift при перетаскивании элемент будет перемещён в новую категорию. Вы также можете перетаскивать книги из списка книг на элементы в браузере тегов; Удаление книги по элементу - элемент автоматически применяется к перетащенным книгам. Например, перетаскивание книги на имя Исаака Азимова -> автор этой книги - Исаак Асимов. Перетаскивание её на тег History добавит тег History в теги книги.

Вы можете легко найти любой элемент в браузере тегов, нажав кнопку поиска в правом нижнем углу. Кроме того, вы можете щёлкнуть правой кнопкой мыши по любому элементу и выбрать одну из нескольких операций. В некоторых примерах можно скрыть его, переименовать или открыть диалоговое окно «Управление x», которое позволяет вам управлять элементами такого типа. Например, диалоговое окно *Управление авторами* позволяет вам переименовывать авторов и управлять сортировкой их имен.

Вы можете управлять сортировкой элементов в браузере тегов с помощью кнопки *Настроить* в его левом нижнем углу. Там можно выбрать сортировку по названию, средней оценке или популярности (популярность — это количество книг с тем или иным элементом в вашей библиотеке, например, популярность Айзека Азимова — это количество книг Айзека Азимова в вашей библиотеке).

Вы можете использовать свои собственные значки для категорий и значений в категориях. Чтобы изменить значок категории, щелкните правой кнопкой мыши по категории (элемент внешнего уровня) и выберите *Изменить значок категории (название категории)*. Откроется диалоговое окно, в котором вы можете выбрать изображение для использования в качестве значка. Чтобы восстановить значок по умолчанию, выберите „Восстановить значок по умолчанию (название категории)`.

Чтобы выбрать значки для значений в категориях, щелкните правой кнопкой мыши значение, затем выберите `Управлять значком для (название значения)“. Вы увидите список вариантов:

- `Выбрать значок для этого значения, но не для его дочерних элементов“. Откроется диалоговое окно, в котором вы выберете значок для значения. Дочерние элементы этого значения не унаследуют этот значок.
- *Выбрать значок для этого значения и его дочерних элементов*. Откроется диалоговое окно, в котором вы выберете значок для значения. Все дочерние элементы, у которых нет своего собственного указанного значка, унаследуют этот значок.
- *Использовать существующий значок для этого значения, но не для его дочерних элементов*. Эта опция предлагается, если у значения уже есть значок, который наследуется дочерними элементами значения. При его выборе значок будет применен к значению, но не к его дочерним элементам.
- *Использовать существующий значок для этого значения и его дочерних элементов*. Эта опция предлагается, если у значения уже есть значок, который не наследуется дочерними элементами значения. При его выборе значок будет применен к значению и его дочерним элементам.
- *Использовать значок по умолчанию для этого значения*. Эта опция предлагается, если у элемента есть значок. Она удаляет значок из значения и всех дочерних элементов, наследующих значок. Значок по умолчанию указан ниже.
- *Восстановить все значки значений на значок по умолчанию*. Этот параметр удаляет все значки значений элементов для категории. Он не удаляет шаблон, если таковой существует. Отмены нет.
- *Использовать/редактировать шаблон, чтобы выбрать значок значения по умолчанию*. Этот параметр позволяет вам предоставить шаблон calibre, который возвращает имя файла значка, который будет использоваться в качестве значка по умолчанию. Шаблон может использовать две переменные:
  - `category`: название категории для поиска, например `authors`, `series`, `#mycolumn`.
  - `value`: значение элемента внутри категории.
  - `count`: количество книг с этим значением. Если значение является частью иерархии, то количество включает дочерние элементы.
  - `avg_rating`: средний рейтинг для книг с этим значением. Если значение является частью иерархии, то среднее значение включает дочерние элементы.

  Метаданные книги, такие как название, недоступны. Функции базы данных шаблонов, такие как : ref: *ff_book_count* и :ref: *ff_book_values*, будут работать, но производительность может быть неприемлемой. Следующие шаблонные функции будут работать в графическом интерфейсе, но не будут работать на сервере контента: *connected_device_name* (страница 204), *connected_device_uuid* (страница 204), *current_virtual_library_name* (страница 204), *is_marked* (страница 206) и :ref: *ff_virtual_libraries*.

  В графическом интерфейсе шаблоны Python имеют полный доступ к базе данных calibre. На сервере контента шаблоны Python имеют доступ к новому API (см. Документация API для интерфейса базы данных[2]), но не к старому API (LibraryDatabase).

  Например, этот шаблон указывает, что любое значение в категории, на которую нажимают, начинающееся с *History*, будет иметь значок с именем `flower.png`:

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

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

Если шаблон возвращает пустую строку (''), то будет использоваться значок категории. Если шаблон возвращает имя файла, которое не существует, то значок не отображается.

- *Использовать значок категории по умолчанию*. Этот параметр указывает, что значок, используемый для категории, должен использоваться для любого значения, у которого в противном случае нет значка. При выборе этого параметра удаляется любая спецификация значка шаблона.

Значок выбирается с использованием следующей иерархии:

1. Значок, указанный для значения, если оно существует.
2. Значок, указанный для родительского узла, найденного путем перемещения вверх по дереву, если таковой существует.
3. Значок из шаблона, если шаблон существует и он возвращает непустую строку.
4. Значок категории по умолчанию, который существует всегда.

Значки для каждого пользователя, а не для каждой библиотеки, хранятся в папке конфигурации calibre. Значки для значений элементов хранятся во вложенной папке `tb_icons`. Значки, используемые шаблонами, находятся во вложенной папке `template_icons` из `tb_icons`.

1.13 Сетка обложек

Вы можете отображать сетку обложек книг, а не список книг, если предпочитаете просматривать свою коллекцию в виде обложек Опция *Сетка обложек* активируется нажатием кнопки *Макет* в нижнем правом углу основного окна. Вы можете настроить размеры обложки и фон *Cover grid* через *Настройки* → *Интерфейс* → *Внешний вид* → *Сетка обложек*. Вы можете даже отображать любое указанное поле под обложками, например, в заголовке или авторах или в рейтинге, или в специальном столбце, который вы можете создать.

1.14 Браузер обложек

В дополнение к *Сетке обложек*, описанной выше, вы можете также отображать обложки в одной строке. Это активируется с помощью кнопки *Макет* в нижнем правом углу главного окна. В меню *Параметры → Интерфейс → Оформление → Браузер обложек* вы можете изменить количество отображаемых обложек и даже отобразить *Браузер обложек* в отдельном всплывающем окне.

1.15 Добавление примечаний для авторов, серий и т.д.

Вы можете добавить примечания для автора/серии/тега/издателя и т.д. в вашу библиотеку calibre. Для этого щелкните правой кнопкой мыши на имени автора в *Браузере тегов* слева или на панели *Описание книги* справа и выберите *Создать примечание* или *Редактировать примечание*.

Простое всплывающее окно позволит вам вводить свои примечания с использованием базового форматирования и поддержкой ссылок и изображений. После создания примечания для автора его можно легко просмотреть в панели «описание книги», щелкнув по маленькому значку карандаша рядом с именем автора.

Вы можете искать все примечания в вашей библиотеке с помощью инструмента «Просмотр примечаний», нажав `Ctrl+Shift+N` или добавив его на панель инструментов через *Настройки → Панели инструментов и меню*.

1.16 Быстрый просмотр

Иногда вы хотите выбрать книгу и быстро получить список книг с одинаковым значением в какой-либо категории (авторы, теги, издатель, серия и т.д.) в качестве текущей выбранной книги, но не изменяя текущего представления библиотеки. Вы можете сделать это с помощью Quickview. Quickview открывает либо второе окно, либо панель в списке книг, отображающую список интересующих книг. Предположим, что вы хотите просмотреть список всех книг с одним или несколькими авторами выбранной в данный момент книги. Нажмите интересующую вас ячейку автора и нажмите клавишу «Q» или щелкните значок *Quickview* в разделе *Layout* окна calibre. Окно или панель откроются со всеми авторами для этой книги слева, а все книги выбранного автора справа.

Некоторые примеры использования Быстрого просмотра: быстро посмотреть, что другие книги:

- имеют некоторые теги, применяемые к текущей выбранной книге,
- в той же серии, что и выбранная книга
- имеют те же значения в пользовательском столбце, как и текущая книга
- написаны одним из тех же авторов, что и текущая книга
- разделяемые значения в настраиваемом столбце

Существует два варианта отображения информации Quickview:

1. Он может открыть «отстыкован»: поверх окна calibre и останется открытым, пока вы явно не закроете его.
2. Он может «пристыкован»: как панель в разделе списка книг главного окна калибра.

Вы можете переместить это окно из закреплённого в незакреплённое по желанию, используя кнопку «Закрепить/Открепить»

Панель Quickview может быть оставлена открытой навсегда, и в этом случае она следует за движениями в списке книг. Например, если вы щелкнете в представлении библиотеки calibre в столбце категории (теги, серии, издателя, авторов и т.д.) для книги, содержимое окна Quickview изменится, чтобы показать вам в левой боковой панели значения в этой категории для выбранной книги (например, теги для этой книги). Первый элемент в этом списке будет выбран, а Quickview покажет вам справа все книги в вашей библиотеке, использующие это значение. Нажмите на другое значение в левой панели, чтобы увидеть книги с этим другим значением.

Дважды щелкните по книге в окне Quickview, чтобы выбрать эту книгу в представлении библиотеки. Это также изменит отображение элементов в окне QuickView (левая панель), чтобы отобразить элементы в недавно выбранной книге.

`Shift-` или `Ctrl-` дважды щёлкните книгу в окне быстрого просмотра, чтобы открыть диалог редактирования её метаданных в окне calibre. Отредактированная книга будет «быстро просмотрена» после закрытия диалога редактирования метаданных.

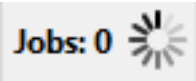
Вы можете увидеть, можно ли посмотреть столбец в Быстром просмотре, по подсказке, когда проводите мышью по заголовку столбца. Правым кликом мыши на заголовке столбца вы можете увидеть настройку «Быстрый просмотре» в показанном меню, таким образом будет ясно, в каких случаях выбор этой настройки эквивалентен нажатию „Q“ в текущей ячейке.

Опции (в *Параметры → Оформление → Быстрый просмотр*):

- Относиться (или нет) к текущей виртуальной библиотеке. Если этот флажок установлен, Quickview показывает только книги в текущей виртуальной библиотеке. По умолчанию: относиться к виртуальным библиотекам
- Изменяйте содержимое окна Quickview, когда столбец меняется в списке книг клавишами курсора. По умолчанию: не выполнять изменения, сделанные с помощью клавиш курсора
- Изменять столбец «quickview'ed», когда ячейка в окне Quickview дважды нажата. В противном случае книга будет изменена, но отображаемый столбец не будет. По умолчанию: изменить столбец

- Изменить столбец «quickview'ed» на текущий столбец, когда клавиша возврата нажата в панели Quickview. В противном случае книга будет изменена, но рассматриваемый столбец не будет. По умолчанию: изменить столбец
- Выберите, какие столбцы показывать в окне/на панели «Быстрый просмотр».

1.17 Задания



На панели Задания отображается количество выполняемых в настоящее время заданий. Задания - это задачи, выполняемые в отдельном процессе. Они включают в себя конвертацию электронных книг и обмен данными с читалкой. Вы можете щелкнуть на панели заданий, чтобы получить доступ к списку заданий. После завершения задания вы можете просмотреть подробный журнал этого задания, дважды щелкнув его в списке. Это полезно для отладки заданий, которые могут не завершиться успешно.

1.18 Сочетания клавиш

В calibre есть горячие клавиши для экономии времени и движений мышью. Эти сочетания активны при просмотре списка книг (когда вы не редактируете данные конкретной книги), и большинство из них влияет на выделенное вами наименование. Читалка электронных книг calibre *имеет собственные горячие клавиши* (страница 57), которые можно настроить, нажав в ней кнопку *Параметры*.

Примечание

Примечание: если специально не оговорено, то для горячих клавиш calibre не требуется ключ-модификатор (Command, Option, Control и т. д.). Необходимо нажать только клавишу с буквой, например «E» для редактирования.

Таблица 1: Горячие клавиши для главного окна calibre

Сочетание клавиш	Действие
`F2 (Enter для macOS)`	Редактировать метаданные текущей выбранной области в списке книг.
`A`	Добавить книги
`Shift+A`	Добавить форматы в выбранные книги
`C`	Конвертировать выбранные книги
`D`	Отправить на устройство
`Del`	Удалить выбранные книги
`E`	Редактировать метаданные выбранных книг
`G`	Приобрести книги
`I`	Показать описание книги
`K`	Редактировать оглавление
`M`	Объединить выбранные записи

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Сочетание клавиш	Действие
Alt+M	Объединить выбранные записи, сохранив оригиналы
O	Открыть папку
P	Полировка книг
S	Сохранить на диск
T	Редактировать книгу
V	Просмотр
Shift+V	Посмотреть последнюю прочитанную книгу
Alt+V/ Cmd+V для macOS	Просмотреть конкретный формат
Alt+Shi	Переключить список задач
Alt+Shi	Переключить браузер обложек
Alt+Shi	Переключить панель описания книги
Alt+Shi	Переключить браузер тегов
Alt+Shi	Переключить сетку обложек
Alt+A	Показать книги с тем же автором, что и у текущей
Alt+T	Показать книги с теми же тегами, что и текущая
Alt+P	Показать книги с таким же издательством, как у текущей
Alt+Shi	Показать книги из той же серии, что и текущая
/, Ctrl+F	Фокус на панель поиска
Shift+C	Открыть диалог расширенного поиска
Shift+A	Переключить строку поиска
Esc	Очистить текущий поиск
Shift+E	Фокус на список книг
Ctrl+Es	Очистить виртуальную библиотеку
Alt+Esc	Очистить дополнительное ограничение
Ctrl+*	Создать временную Виртуальную библиотеку на основе текущего поиска
Ctrl+Ri	Выбрать следующую вкладку Виртуальной библиотеки
Ctrl+Le	Выбрать предыдущую вкладку Виртуальной библиотеки
N или F3	Найти следующую книгу, которая соответствует текущим условиям поиска (работает только если в настройках поиска включена подсветка поиска)
Shift+N или Shift+F	Найти предыдущую книгу, которая соответствует текущим условиям поиска (работает только если в настройках поиска включена подсветка поиска)
Ctrl+D	Загрузить метаданные и обложки
Ctrl+R	Перезапустить calibre
Ctrl+Sh	Перезапустить calibre в отладочном режиме
Shift+C	Добавить пустые книги в calibre
Ctrl+M	Переключить статус „помечено“ для выбранных книг
Ctrl+/ или Ctrl+Al	Откройте всплывающее окно для поиска по полному тексту всех книг в библиотеке.
Q	Открыть окно Быстрого Просмотра для просмотра книг из связанных серий/тегов и т.д.
Shift+Q	Фокус на открытую панель быстрого просмотра
Shift+S	Выполнить поиск в панели быстрого просмотра

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Сочетание клавиш	Действие
F5	Re-apply the current sort
Ctrl+Q	Выйти из calibre
X	Включить автоматическую прокрутку списка книг
Ctrl+Al	Ограничить отображаемые книги только книгами, находящимися в категории, отображаемой в данный момент в *Браузер тегов*
B	Просмотреть аннотации (выделения и закладки), сделанные в просмотрщике calibre для всех книг в библиотеке
Ctrl+Sh	Просмотреть примечания, связанные с авторами/сериями/тегами/и т.д.
Alt+Shi	Переключить на широкий или узкий макет

ГЛАВА 2

Добавление любимых новостных веб-сайтов

calibre обладает мощной, гибкой и простой в использовании платформой для загрузки новостей из Интернета и преобразования их в электронную книгу. Далее на примерах будет показано, как получать новости с различных веб-сайтов.

Чтобы понять, как использовать платформу, следуйте примерам в порядке, указанном ниже:

- *Полностью автоматическая загрузка* (страница 33)
  - *Блог calibre* (страница 34)
  - *bbc.co.uk* (страница 35)
- *Настройка процесса получения* (страница 35)
  - *Использование печатной версии bbc.co.uk* (страница 35)
  - *Замена стилей статей* (страница 37)
  - *Шинковка ломтиками и кубиками* (страница 37)
  - *Пример из реальной жизни* (страница 38)
- *Советы по созданию новых рецептов* (страница 40)
- *Дальнейшее чтение* (страница 41)
- *Документация API* (страница 41)

2.1 Полностью автоматическая загрузка

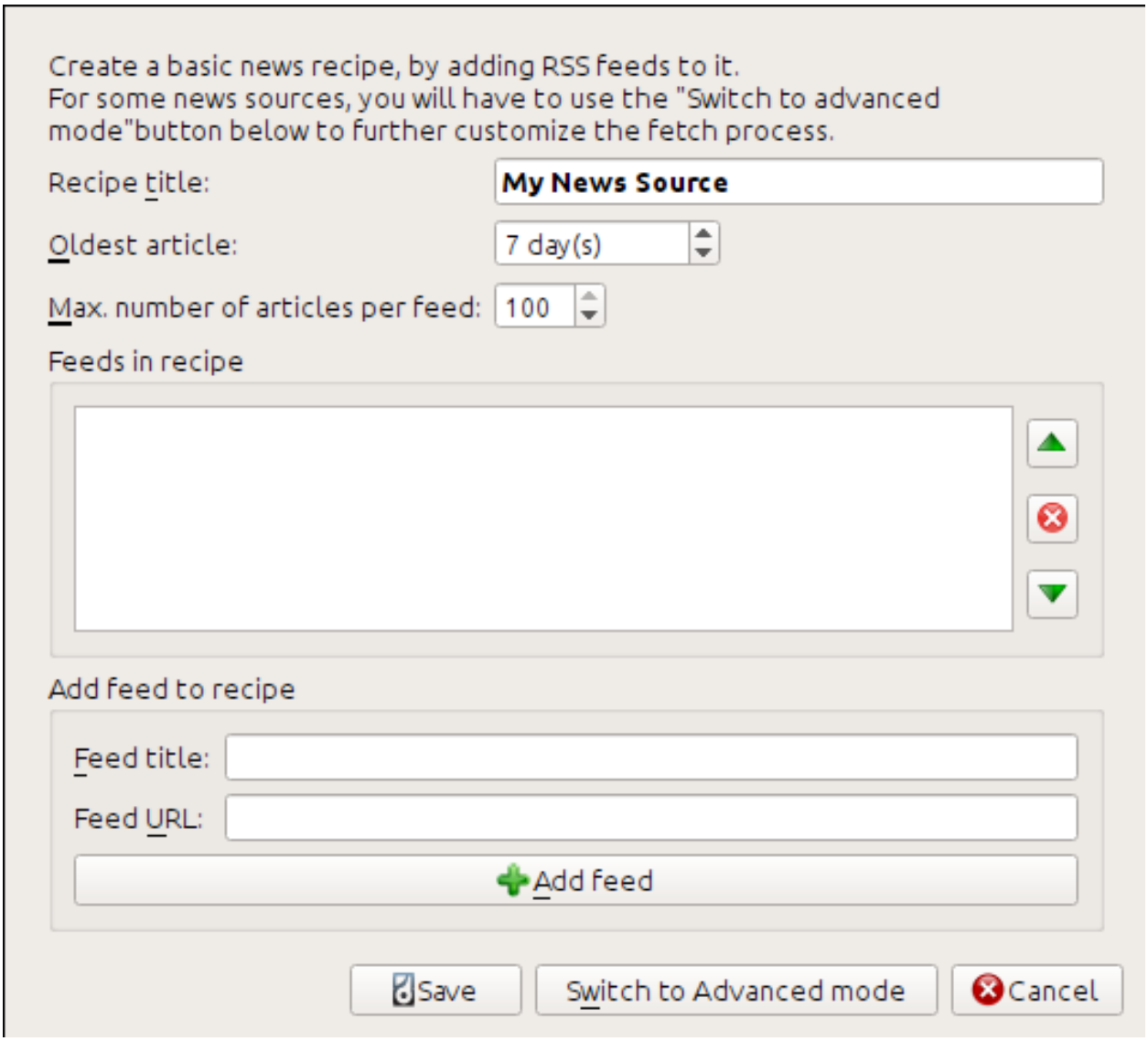
Если ваш источник новостей достаточно прост, calibre вполне может получить его полностью автоматически, все, что вам нужно сделать, это предоставить URL. calibre собирает всю информацию, необходимую для загрузки источника новостей, в *recipe*. Чтобы сообщить calibre об источнике новостей, вы должны создать для него *recipe*. Давайте посмотрим несколько примеров:

2.1.1 Блог calibre

Блог calibre - это блог постов, которые описывают многие полезные функции calibre простым и доступным способом для пользователей нового calibre. Чтобы загрузить этот блог в электронную книгу, мы полагаемся на *RSS* ленту блога:

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

Я получил URL-адрес RSS, посмотрев в разделе «Подписаться на» в нижней части страницы блога и выбрав *Posts → Atom*. Чтобы заставить calibre загружать каналы и конвертировать их в электронную книгу, вы должны щелкнуть правой кнопкой мыши по кнопке *Fetch news*, а затем *Добавить пользовательский источник новостей* и затем по кнопке :guilabel:`«Новый рецепт». Должно открыться диалоговое окно, подобное показанному ниже.



Сначала введите `Calibre Blog` в поле *Название рецепта*. Это будет название электронной книги, которая будет создана из статей в вышеупомянутых каналах.

Следующие два поля (*Самая старая статья* и :guilabel:` Максимальное количество статей`) позволяют вам контролировать, сколько статей следует загружать из каждого канала, и они довольно понятны.

Чтобы добавить каналы в рецепт, введите заголовок канала и URL-адрес канала и нажмите кнопку *Добавить*

канал. После добавления канала просто нажмите кнопку *Сохранить*, и все готово! Закройте диалог.

Чтобы проверить свой новый *recipe*, нажмите кнопку *Получить новости* и в подменю *Пользовательские источники новостей* нажмите *Блог calibre*. Через пару минут недавно загруженная электронная книга из сообщений блога появится в главном представлении библиотеки (если у вас подключена читалка, она будет помещена в читалку, а не в библиотеку). Выберите книгу и нажмите кнопку *View*, чтобы прочитать!

Причина, по которой это сработало так хорошо, с таким небольшим усилием, состоит в том, что блог предоставляет *полный контентные RSS* фиды, то есть контент статьи внедряется в сам фид. Для большинства новостных источников, которые предоставляют новости таким образом, с *полнотекстовыми* новостями, вам не нужно больше усилий, чтобы преобразовать их в электронные книги. Теперь мы рассмотрим источник новостей, который не предоставляет полную ленту контента. В таких каналах полная статья является веб-страницей, а лента содержит только ссылку на веб-страницу с кратким обзором статьи.

2.1.2 bbc.co.uk

Давайте попробуем следующие два фида из * BBC *:

1. Главная страница новостей: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_page/rss.xml
2. Наука/Природа: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/science/nature/rss.xml

Следуйте процедуре, описанной в *Блог calibre* (страница 34) выше, чтобы создать рецепт для *BBC* (используя фиды выше). Глядя на загруженную электронную книгу, мы видим, что calibre проделал достойную работу по извлечению только того контента, который вам интересен, с веб-страницы каждой статьи. Однако процесс извлечения не идеален. Иногда он оставляет нежелательный контент, такой как меню и навигационные средства, или удаляет контент, который должен был остаться, например заголовки статей. Чтобы обеспечить идеальное извлечение контента, нам нужно настроить процесс извлечения, как описано в следующем разделе.

2.2 Настройка процесса получения

Если вы хотите усовершенствовать процесс загрузки или загрузить контент с особенно сложного веб-сайта, вы можете воспользоваться всей мощью и гибкостью среды *recipe*. Для этого в диалоге *Добавить пользовательские источники новостей* просто нажмите кнопку *Переключиться в расширенный режим*.

Самая простая и часто наиболее производительная настройка заключается в использовании версии для печати онлайн-статей. Такая версия, как правило, имеет гораздо меньше ошибок и гораздо более гладко переводится в электронную книгу. Давайте попробуем использовать версию для печати статей из *BBC*.

2.2.1 Использование печатной версии bbc.co.uk

Первым шагом является просмотр электронной книги, которую мы скачали ранее *bbc.co.uk* (страница 35). В конце каждой статьи в электронной книге есть небольшая реклама, в которой рассказывается, откуда была загружена статья. Скопируйте и вставьте этот URL в браузер. Теперь на веб-странице статьи найдите ссылку, которая указывает на «Версия для печати». Нажмите на неё, чтобы увидеть печатную версию статьи. Это выглядит намного аккуратнее! Теперь сравните два URL. Для меня они были такими:

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

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

Таким образом, похоже, чтобы получить версию для печати, нам нужно добавить префикс к каждой статье:

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

Теперь в *Advanced Mode* диалога Custom news sources вы должны увидеть что-то вроде (не забудьте выбрать рецепт *BBC* перед переходом в расширенный режим):

Вы можете видеть, что поля из *Basic mode* были прямо переведены в код Python. Нам нужно добавить инструкции к этому рецепту, чтобы использовать печатную версию статей. Всё, что нужно, это добавить следующие две строки:

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

Это Python, поэтому отступы важны. После того, как вы добавили строки, это должно выглядеть так:

Выше `def print_version(self, url)` определяет *метод*, который вызывается calibre для каждой статьи. `url` - это URL оригинальной статьи. `print_version` - это взять этот URL и заменить его новым URL, который указывает на печатную версию статьи. Чтобы узнать о Python[3], см. tutorial[4].

Теперь нажмите кнопку *Добавить/обновить рецепт*, и ваши изменения будут сохранены. Повторно загрузите электронную книгу. У вас должна быть намного улучшенная электронная книга. Одна из проблем новой версии заключается в том, что шрифты на веб-странице версии для печати слишком малы. Это автоматически исправляется при преобразовании в электронную книгу, но даже после процесса исправления размер шрифта меню и панели навигации становится слишком большим по сравнению с текстом статьи. Чтобы исправить это, мы сделаем ещё несколько настроек в следующем разделе.

[3] https://www.python.org
[4] https://docs.python.org/tutorial/

2.2.2 Замена стилей статей

В предыдущем разделе мы видели, что размер шрифта для статей из печатной версии *BBC* был слишком мал. На большинстве веб-сайтов, включая *BBC*, этот размер шрифта устанавливается с помощью *CSS* таблиц стилей. Мы можем отключить выборку таких таблиц стилей, добавив строку:

```
no_stylesheets = True
```

Рецепт теперь выглядит так:

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/
```

Новая версия выглядит довольно хорошо. Если вы перфекционист, вам нужно прочитать следующий раздел, посвященный изменению загруженного контента.

2.2.3 Шинковка ломтиками и кубиками

Что касается управления загруженным содержимым, calibre обладает мощными и гибкими возможностями. Чтобы похвастаться парочкой, давайте снова посмотрим на нашего старого друга — рецепт *BBC* (страница 37). Взглянув на исходный код (*HTML*) пары статей (версия для печати), мы увидим в них нижний колонтитул без какой-либо полезной информации, содержащейся в

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

Это может быть удалено путем добавления:

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

к рецепту. Наконец, давайте заменим некоторые из *CSS*, которые мы отключили ранее, на наш собственный *CSS*, который подходит для преобразования в электронную книгу:

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

Благодаря этим дополнениям наш рецепт стал «производственным качеством».

Это *recipe* исследует только верхушку айсберга, когда дело доходит до силы calibre. Чтобы изучить больше возможностей calibre, мы рассмотрим более сложный пример из реальной жизни в следующем разделе.

2.2.4 Пример из реальной жизни

Достаточно сложный пример из реальной жизни, который раскрывает большую часть *API* `BasicNewsRecipe`, это *recipe* для *The New York Times*

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

class NYTimes(BasicNewsRecipe):

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

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

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

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

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

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

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                ans.append(key)

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

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

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

В этом мы видим несколько новых функций *recipe*. Во-первых, у нас есть:

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

Это устанавливает отображаемое время на первой странице созданной электронной книги в формате `Day, Day_Number Month, Year`. Смотрите *timefmt* (страница 52).

Затем мы видим группу директив для очистки загруженного файла *HTML*:

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

Они удаляют все до первого тега `<h1>` и все после первого тега с идентификатором `` footer``. Смотрите *remove_tags* (страница 51), *remove_tags_before* (страница 51), *remove_tags_after* (страница 51).

Следующая интересная особенность:

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

`needs_subscription = True` сообщает calibre, что для получения доступа к содержимому этого рецепта требуется имя пользователя и пароль. Это заставляет calibre запрашивать имя пользователя и пароль всякий раз, когда вы пытаетесь использовать этот рецепт. Код в *calibre.web.feeds.news.BasicNewsRecipe.get_browser()* (страница 43) фактически выполняет вход на веб-сайт NYT. После входа в систему calibre будет использовать тот же, вошедший в систему, экземпляр браузера, чтобы получить весь контент. Смотрите mechanize[5], чтобы понять код в `get_browser`.

Следующей новой функцией является метод *calibre.web.feeds.news.BasicNewsRecipe.parse_index()* (страница 45). Его работа заключается в том, чтобы перейти на https://www.nytimes.com/pages/todayspaper/index.html и получить список статей, которые появляются в *сегодняшнем* документе. Хотя этот рецепт сложнее, чем просто использовать *RSS*, этот рецепт создает электронную книгу, которая очень точно соответствует дневнику. `parse_index` интенсивно использует BeautifulSoup[6] для анализа ежедневной веб-страницы газеты. Вы также можете использовать другие, более современные парсеры, если вам не нравится BeautifulSoup. calibre поставляется с *lxml <https://lxml.de/>* _ и html5lib[7], которые являются рекомендуемыми парсерами. Чтобы использовать их, замените вызов `index_to_soup()` следующим:

```
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)
```

Последняя новая функция - это метод *calibre.web.feeds.news.BasicNewsRecipe.preprocess_html()* (страница 46). Его можно использовать для выполнения произвольных преобразований на каждой загруженной HTML-странице. Здесь он используется для обхода рекламы, которую nytimes показывает перед каждой статьей.

2.3 Советы по созданию новых рецептов

Лучший способ разработать новые рецепты - использовать интерфейс командной строки. Создайте рецепт, используя ваш любимый редактор Python, и сохраните его в файл, скажем `myrecipe.recipe`. Требуется расширение *.recipe*. Вы можете скачать контент, используя этот рецепт с командой:

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

Команда **ebook-convert** загрузит все веб-страницы и сохранит их в файл EPUB `myrecipe.epub`. Параметр `-vv` заставляет ebook-convert выдавать много информации о том, что он делает. Параметр `ebook-convert-recipe-input --test` позволяет загружать только пару статей максимум из двух каналов. Кроме того, ebook-convert поместит загруженный HTML-код в папку debug/input, где debug - это папка, указанная в параметре *ebook-convert --debug-pipeline* (страница 345).

После завершения загрузки вы можете просмотреть загруженный файл *HTML*, открыв файл `debug/input/index.html` в браузере. Убедившись, что загрузка и предварительная обработка выполняются правильно, вы можете создавать электронные книги в различных форматах, как показано ниже:

[5] https://mechanize.readthedocs.io/en/latest/
[6] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[7] https://github.com/html5lib/html5lib-python

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

Если вы удовлетворены своим рецептом и чувствуете, что существует достаточный спрос, чтобы оправдать его включение в набор встроенных рецептов, опубликуйте свой рецепт на форуме рецептов calibre[8], чтобы поделиться им с другими пользователями.

Примечание

В macOS инструменты командной строки находятся внутри пакета calibre, например, если вы установили calibre в `/Applications`, то инструменты командной строки будут в `/Applications/calibre.app/Contents/MacOS/`.

См. также

ebook-convert **(страница 335)**
Интерфейс командной строки для всех конвертаций электронных книг.

2.4 Дальнейшее чтение

Чтобы узнать больше о написании расширенных рецептов с использованием некоторых средств, доступных в `BasicNewsRecipe`, вам следует обратиться к следующим источникам:

Документация API **(страница 41)**
Документация класса `BasicNewsRecipe` и всех его важных методов и полей.

BasicNewsRecipe[9]
Исходный код `BasicNewsRecipe`

Встроенные рецепты[10]
Исходный код для встроенных рецептов, которые идут с calibre

Форум рецептов calibre[11]
Много знающих авторов рецепта calibre тусуются здесь.

2.5 Документация API

2.5.1 Документация API рецептов

API для написания рецептов определяется классом *BasicNewsRecipe* (страница 41)

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

Основной класс, содержащий логику, необходимую для всех рецептов. Переопределяя функциональные возможности этого класса, вы постепенно сможете создавать более настраиваемые/действенные рецепты. Вводный курс по созданию рецептов помещён в разделе «*Добавление любимых новостных веб-сайтов* (страница 33)».

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

classmethod adeify_images(*soup*)

Если ваш рецепт при преобразовании в EPUB имеет проблемы с изображениями при просмотре в Adobe Digital Editions, вызовите этот метод изнутри *postprocess_html()* (страница 45).

classmethod image_url_processor(*baseurl*, *url*)

Выполнить обработку URL-адресов изображений (возможно, удалить ограничения по размеру для динамически генерируемых изображений и т.д.) и вернуть обработанный URL-адрес. Вернуть None или пустую строку, чтобы пропустить получение изображения.

classmethod print_version(*url*)

Взять *url*, указывающий на веб-страницу с содержанием статьи, и вернуть *URL*, указывающий на печатную версию статьи. По умолчанию ничего не делает. Например:

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

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

Удобный метод для того, чтобы взять **`BeautifulSoup https://www.crummy.com/software/BeautifulSoup/bs4/doc/`_**: code:` Tag` и рекурсивно извлечь из него текст, включая любые разделы CDATA и атрибуты тегов alt. Вернуть возможно пустую строку Unicode.

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

tag: BeautifulSoup[12] `Tag`

abort_article(*msg=None*)

Вызовите этот метод в любом из методов предварительной обработки, чтобы прервать загрузку текущей статьи. Полезно для пропуска статей, содержащих неприемлемый контент, например видеостатей.

abort_recipe_processing(*msg*)

Заставляет систему загрузки рецептов прервать загрузку этого рецепта, отображая простое сообщение обратной связи для пользователя.

add_toc_thumbnail(*article*, *src*)

Вызовите из populate_article_metadata с атрибутом src тега <img> из статьи, который подходит для использования в качестве эскиза, представляющего статью в Оглавлении. Используется ли миниатюра на самом деле, зависит от устройства (сейчас используется только в Kindle). Обратите внимание, что указанное изображение должно быть успешно загружено, иначе оно будет проигнорировано.

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

Вернуть набор канонических представлений `url`. Реализация по умолчанию использует только имя хоста сервера и путь URL-адреса, игнорируя любые параметры запроса, фрагменты и т.д. Канонические представления должны быть уникальными для всех URL-адресов для этого источника новостей. В противном случае внутренние ссылки могут быть разрешены неверно.

Параметры

is_link – Имеет значение True, если URL-адрес поступает по внутренней ссылке в файле HTML. False, если URL-адрес является URL-адресом, используемым для загрузки статьи.

cleanup()

Вызывается после загрузки всех статей. Используйте его для любой очистки, например, для выхода с сайтов подписки и т. д.

clone_browser(*br*)

Клонировать браузер br. Клонированные браузеры используются для многопоточной загрузки, поскольку механизация не является потокобезопасной. Подпрограммы клонирования по умолчанию должны фиксировать большинство настроек браузера, но если вы делаете что-то экзотическое в своем рецепте, вам следует переопределить этот метод в своем рецепте и клонировать вручную.

Клонированные экземпляры браузера по умолчанию используют тот же потокобезопасный CookieJar, если вы не настроили обработку файлов cookie.

default_cover(*cover_file*)

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

download()

Загрузить и предварительно обработать все статьи из каналов в этом рецепте. Этот метод следует вызывать только один раз для конкретного экземпляра Recipe. Вызов его более одного раза приведет к неопределенному поведению. :return:Путь к index.html

extract_readable_article(*html*, *url*)

Извлечь основное содержание статьи из html, очистить и вернуть в виде кортежа (article_html, extract_title). На основе оригинального алгоритма читаемости от Arc90.

get_article_url(*article*)

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

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

Вернуть экземпляр браузера, используемый для получения документов из Интернета. По умолчанию он возвращает экземпляр браузера mechanize[15], который поддерживает файлы cookie, игнорирует robots.txt, обрабатывает обновления и имеет случайный пользовательский агент.

Чтобы настроить браузер, переопределите этот метод в вашем подклассе следующим образом:

```
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()

По умолчанию возвращает *self.extra_css*. Переопределите, если вы хотите программно сгенерировать файл extra_css.

get_feeds()

Вернуть список *RSS* каналов для получения в этом профиле. Каждый элемент списка д. б. двухэлементным кортежем формы (заголовок, URL). Если заголовок - None или пустая строка, используется заголовок из фида. Метод полезен, если рецепт требует вычисления списка каналов для загрузки. Если так, переопределите в своем подклассе.

get_masthead_title()

Переопределить в подклассе, чтобы использовать что-то другое, кроме названия рецепта

get_masthead_url()

Вернуть *URL* к изображению masthead для этой проблемы или *None*. По умолчанию возвращает значение члена self.masthead_url, которое обычно равно None. Если вы хотите, чтобы ваш рецепт загружал заголовок для электронной книги, переопределите этот метод в своем подклассе или установите переменную члена *self.masthead_url* перед вызовом этого метода. Изображения masthead используются в файлах Kindle MOBI.

get_obfuscated_article(*url*)

Если вы установите *articles_are_obfuscated*, этот метод вызывается для каждого URL-адреса статьи. Он должен вернуть путь к файлу в файловой системе, содержащему HTML-код статьи. Этот файл обрабатывается механизмом рекурсивной выборки HTML, поэтому он может содержать ссылки на страницы/изображения в Интернете. Альтернативно вы можете вернуть словарь вида: {„data“: <HTML data>, „url“: <the resolved URL of the article>}. Это позволяет избежать необходимости создавать временные файлы. Ключ *url* в словаре полезен, если эффективный URL-адрес статьи отличается от URL-адреса, переданного в этот метод, например, из-за перенаправлений. Его можно опустить, если URL-адрес не изменился.

Этот метод обычно полезен для сайтов, которые пытаются затруднить автоматический доступ к содержимому статьи.

get_url_specific_delay(*url*)

Возвращает задержку в секундах перед загрузкой этого URL. Если вы хотите программно определить задержку для указанного URL-адреса, переопределите этот метод в своём подклассе, возвращая self.delay по умолчанию для URL-адресов, на которые вы не хотите влиять.

Результат

Число с плавающей запятой, задержка в секундах.

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

Удобный метод, который принимает URL-адрес на страницу индекса и возвращает его **`BeautifulSoup https://www.crummy.com/software/BeautifulSoup/bs4/doc`__**.

url_or_raw: либо URL, либо загруженная страница индекса в виде строки

is_link_wanted(*url*, *tag*)

Вернуть True, если по ссылке нужно перейти, или False в противном случае. По умолчанию вызывает NotImplementedError, что заставляет загрузчик игнорировать его.

Параметры

- **url** – URL, по которому нужно следовать
- **tag** – Тег, из которого был получен URL

parse_feeds()

Создать список статей из списка каналов, возвращаемых *BasicNewsRecipe.get_feeds()* (страница 44). Вернуть список `Feed`.

parse_index()

Этот метод д. б. реализован в рецептах для анализа веб-сайта, а не в каналах для создания списка статей. Типичное использование - источники новостей, с веб-страницей «Для печати» со всеми статьями. Если эта функция реализована, она будет использоваться вместо *BasicNewsRecipe.parse_feeds()* (страница 45).

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

Каждый список статей должен содержать словари вида:

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

Для примера см. Рецепт загрузки *The Atlantic*. Кроме того, вы можете добавить *author* для автора статьи.

Если вы хотите по какой-то причине прервать обработку, то чтобы calibre показал пользователю простое сообщение вместо ошибки, вызовите *abort_recipe_processing()* (страница 42).

populate_article_metadata(*article*, *soup*, *first*)

Вызывается при загрузке каждой HTML-страницы, принадлежащей статье. Предназначен для получения метаданных статьи, таких как автор/резюме/и т. д. из разобранного HTML (soup)

Параметры

- **article** – Объект класса `calibre.web.feeds.Article`. Если вы измените сводку, не забудьте также изменить text_summary
- **soup** – Разобранный HTML, принадлежащий этой статье
- **first** – True, если проанализированный HTML - первая страница статьи.

postprocess_book(*oeb*, *opts*, *log*)

Запустить любую необходимую постобработку для проанализированной загруженной электронной книги.

Параметры

- **oeb** – OEBBook объект
- **opts** – Опции конвертации

postprocess_html(*soup*, *first_fetch*)

Этот метод вызывается для источника каждого загруженного файла *HTML* после его анализа на наличие ссылок и изображений. Можно использовать для постобработки *HTML*. После обработки он должен вернуть *soup*.

Параметры

- **soup** – A BeautifulSoup[16] instance containing the downloaded *HTML*.
- **first_fetch** – True если это первая страница статьи.

preprocess_html(*soup*)

Этот метод вызывается для источника каждого загруженного файла: term: *HTML* перед анализом на предмет ссылок и изображений. Он вызывается после очистки, remove_tags и т. д. Можно использовать для предварительной обработки *HTML*. После обработки он должен вернуть *soup*.

soup: экземпляр` BeautifulSoup <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>`__, содержащий загруженный *HTML*.

preprocess_image(*img_data*, *image_url*)

Выполнить обработку загруженных данных изображения. Вызывается для сырых данных перед изменением размера. Должен возвращать обработанные сырые данные. Вернуть None, чтобы пропустить изображение.

preprocess_raw_html(*raw_html*, *url*)

Этот метод вызывается для источника каждого загруженного файла *HTML* перед его синтаксическим анализом в дерево объектов. raw_html - это Юникод-строка, представляющая необработанный HTML-код, загруженный из Интернета. url - это URL-адрес, откуда был загружен HTML-код.

Note that this method acts *before* preprocess_regexps.

Этот метод должен возвращать обработанный raw_html как Unicode-объект .

publication_date()

Используйте этот метод, чтобы установить дату публикации этого выпуска. По умолчанию на момент загрузки. Должен вернуть `datetime.datetime` объект.

skip_ad_pages(*soup*)

Этот метод вызывается для источника каждого загруженного файла *HTML* перед применением любого из атрибутов очистки, таких как remove_tags, keep_only_tags. Обратите внимание, что preprocess_regexps уже был применен. Это сделано для того, чтобы рецепт пропускал рекламные страницы. Если soup - рекламная страница, вернуть HTML реальной страницы. Иначе вернуть None.

soup: экземпляр` BeautifulSoup <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>`__, содержащий загруженный *HTML*.

sort_index_by(*index*, *weights*)

Удобный метод сортировки заголовков в *index* по *weights*. index сортируется на месте. Возвращает *index*.

index: A list of titles.

weights: словарь, который отображает веса в названия. Если какие-либо названия в индексе не имеют веса, предполагается, что они имеют вес 0.

articles_are_obfuscated = False

Установите значение True и реализуйте *get_obfuscated_article()* (страница 44) для обработки веб-сайтов, пытающихся затруднить очистку содержимого.

auto_cleanup = False

Автоматически извлечь весь текст со страниц загруженных статей. Использует алгоритмы из проекта удобочитаемости. Установить значение в True чтобы не чистить загруженный HTML вручную (хотя ручная очистка всегда будет лучше).

auto_cleanup_keep = None

Укажите элементы, которые алгоритм автоматической очистки никогда не должен удалять. Синтаксис - это выражение XPath. Например:

```
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'

Имитируемый движок браузера для использования при загрузке с серверов. По умолчанию используется движок браузера Python mechanize, который поддерживает вход в систему. Однако, если вам не нужен вход в систему, рассмотрите возможность изменения этого параметра либо на „webengine“, который использует реальный браузер Chromium для выполнения сетевых запросов, либо на „qt“, который использует сетевой сервер Qt. И „webengine“, и „qt“ поддерживают HTTP/2, не поддерживаемый mechanize, и, следовательно, его сложнее идентифицировать для служб защиты от ботов.

center_navbar = True

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

compress_news_images = False

Установите значение False, чтобы игнорировать все параметры масштабирования и сжатия и передавать изображения без изменений. Если True и другие параметры сжатия оставлены в значениях по умолчанию, изображения JPEG будут масштабированы, чтобы соответствовать размерам экрана, установленным выходным профилем, и сжиматься до размера не более (w * h)/16, где w x h - размеры масштабированного изображения.

compress_news_images_auto_size = 16

Коэффициент, используемый при автоматическом сжатии изображений JPEG. Если установлено значение «Нет», автоматическое сжатие отключено. В противном случае изображения будут уменьшены в размере до (w * h) / compress_news_images_auto_size байтов, если это возможно, за счет снижения уровня качества, где w x h - размеры изображения в пикселях. Минимальное качество JPEG будет 5/100, поэтому м.б. это ограничение не будет выполнено. Этот параметр можно переопределить с помощью параметра compress_news_images_max_size, обеспечивающего фиксированный максимальный размер изображений. Обратите внимание: если вы включите scale_news_images_to_device, изображение сначала будет масштабировано, а затем его качество будет снижено до тех пор, пока его размер не станет меньше, чем (w * h) / factor, где w и h теперь являются * масштабированными * размерами изображения. Другими словами, это сжатие происходит после масштабирования.

compress_news_images_max_size = None

Установить качество JPEG, чтобы изображения не превышали указанный размер (в килобайтах). Если установлено, этот параметр отменяет автоматическое сжатие через compress_news_images_auto_size. Минимальное качество JPEG будет 5/100, поэтому м.б. это ограничение не будет выполнено.

conversion_options = {}

Определенные параметры рецепта для управления преобразованием загруженного содержимого в электронную книгу. Они переопределят любые значения, указанные пользователем или плагином, поэтому используйте их только в случае крайней необходимости. Например:

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

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

По умолчанию изображение обложки, возвращаемое функцией get_cover_url(), будет использоваться в качестве обложки для журнала. Отмена этого параметра в вашем рецепте дает команду calibre преобразовать загруженную обложку в рамку, ширина и высота которой выражаются в процентах от загруженной обложки. cover_margins = (10, 15, „#ffffff“) дополняет обложку белым полем 10 пикселей слева и справа, 15 пикселей сверху и снизу. Названия цветов определены здесь[17]. Учтите, что белый цвет почему-то не всегда работает в Windows. Вместо этого используйте #ffffff

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()* (страница 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

Указать переопределяемую кодировку для сайтов с неправильным указанием кодировки. Чаще всего указывается latin1 и используется cp1252. Если None, попробует определить кодировку. Объект вызывается с двумя аргументами: объектом рецепта и источником для декодирования. Он должен возвращать декодированный источник.

extra_css = None

Указать любую дополнительную :term: *CSS*, которая д. б. добавлена к загружаемым *HTML* файлам. Она будет вставлена в *<style>`теги `непосредственно перед закрывающим `</head>`тегом, переопределив все :term:`CSS*, за исключением той, которая объявлена с использованием стилевого атрибута в отдельных тегах *HTML*. Обратите внимание: если вы хотите программно сгенерировать extra_css, вместо этого переопределите метод `get_extra_css ()`. Например:

```
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 = []

Список регулярных выражений, определяющий, какие ссылки игнорировать. Если пустой - игнорируется. Используется, только если is_link_wanted не реализован. Например:

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

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

Должен быть определен только один из *BasicNewsRecipe.match_regexps* (страница 49) или *BasicNewsRecipe.filter_regexps* (страница 48).

handle_gzip = True

Установите значение False, если вы не хотите использовать gzip-сжатие передач в браузере mechanize. Внимание! Некоторые старые серверы могут захлебнуться при включенном gzip-сжатии.

ignore_duplicate_articles = None

Игнорировать дубликаты статей, которые присутствуют более чем в одном разделе. Дублирующаяся статья - это статья с таким же названием и/или URL. Чтобы игнорировать статьи с таким же названием, установите для этого параметра значение:

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

Для использования URL-адресов вместо этого, установите значение:

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

Для соответствия заголовку или URL, установите значение:

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

keep_only_tags = []

Оставить только указанные теги и их дочерние элементы. Формат указания тега см. *BasicNewsRecipe.remove_tags* (страница 51). Если этот список не пуст, то *<body>* тег будет очищен и повторно заполнен тегами, соответствующими записям в этом списке. Например:

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

сохранить только теги, у которых есть атрибут *id «content»* или *«heading»*.

language = 'und'

Язык новостей. Должен быть двух- или трёхзначным кодом ISO-639.

masthead_url = None

По умолчанию calibre будет использовать изображение по умолчанию для шапки (только Kindle). Переопределите это в своем рецепте, чтобы указать URL-адрес для использования в качестве шапки.

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]+']
```

будет соответствовать всем URL-адресам, в которых есть *page = some number*.

Должен быть определен только один из *BasicNewsRecipe.match_regexps* (страница 49) или *BasicNewsRecipe.filter_regexps* (страница 48).

max_articles_per_feed = 100

Максимальное количество статей для загрузки из каждого канала. Это в первую очередь полезно для каналов, в которых нет дат статьи. Для большинства каналов вы должны использовать *BasicNewsRecipe.oldest_article* (страница 49)

needs_subscription = False

Если True, графический интерфейс запросит у пользователя имя пользователя и пароль для использования при загрузке. Если установлено значение „optional“, использование имени пользователя и пароля становится необязательным.

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

Самая старая статья для загрузки из этого источника новостей. В днях.

preprocess_regexps = []

Список правил подстановки *regexp* для запуска в загруженном *HTML*. Каждый элемент списка должен быть двухэлементным кортежем. Первый элемент кортежа должен быть скомпилированным регулярным выражением, а второй - вызываемым, который принимает единственный соответствующий объект и возвращает строку для замены совпадения. Например:

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

удалит все от <!–Article ends here–>` до *</body>*.

publication_type = 'unknown'

Тип публикации Газета, журнал или блог. Если установлено значение Нет, метаданные типа публикации не будут записываться в файл opf.

recipe_disabled = None

Установите непустую строку, чтобы отключить этот рецепт. Строка будет использоваться как отключенное сообщение

recipe_specific_options = None

Укажите параметры, специфичные для этого рецепта. Они будут доступны пользователю для настройки на вкладке «Дополнительно» диалогового окна «Получение новостей» или в командной строке ebook-convert. Параметры указываются как имя параметра сопоставления словаря с метаданными об этом параметре. Например:

```
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',
    }
}
```

При запуске рецепта self.recipe_specify_options будет именем параметра сопоставления dict со значением параметра, указанным пользователем. Если параметр не указан пользователем, он будет иметь значение, указанное в значении «по умолчанию». Если значение по умолчанию не указано, этот параметр вообще не будет присутствовать в словаре, если он не указан пользователем.

recursions = 0

Количество уровней ссылок для перехода на веб-страницах статей

remove_attributes = []

Список атрибутов, которые нужно удалить из всех тегов. Например:

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

remove_empty_feeds = False

Если True, пустые каналы удаляются из вывода. Эта опция не действует, если parse_index переопределен в подклассе. Он предназначен только для рецептов, которые возвращают список кана-

лов с использованием *feeds* или *get_feeds()* (страница 44). Он также используется, с опцией ignore_duplicate_articles.

remove_javascript = True

Convenient flag to strip all JavaScript tags from the downloaded HTML

remove_tags = []

Список удаляемых тегов. Указанные теги удаляются из загруженного HTML. Тег задается как словарь формы:

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

Все ключи необязательны. Полное объяснение критериев поиска см. В **`Beautiful Soup https://www.crummy.com/software/BeautifulSoup/bs4/doc/#searching-the-tree`__** A common example:

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

Это приведет к удалению всех тегов * * и всех их дочерних элементов из загруженного *HTML*.

remove_tags_after = None

Удалить все теги, появляющиеся после указанного тега. Формат указания тега см. В *BasicNewsRecipe.remove_tags* (страница 51). Например:

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

удалит все теги после первого элемента с *id=»content»*.

remove_tags_before = None

Удалить все теги, появляющиеся перед указанным тегом. Формат указания тега см. В *BasicNewsRecipe.remove_tags* (страница 51). Например:

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

удалит все теги перед первым элементом с *id=»content»*.

requires_version = (0, 6, 0)

Минимальная версия calibre, необходимая для использования этого рецепта

resolve_internal_links = False

Если установлено значение True, то ссылки в загруженных статьях, которые указывают на другие загруженные статьи, изменяются, чтобы указывать на загруженную копию статьи, а не на ее исходный веб-URL. При установке значения True, вам также может потребоваться реализовать *canonicalize_internal_url()* (страница 42) для работы со схемой URL-адресов вашего конкретного веб-сайта.

reverse_article_order = False

Измените порядок статей в каждом фиде

scale_news_images = None

Максимальные размеры (ширина, высота) для масштабирования изображений. Если scale_news_images_to_device имеет значение True, для него устанавливаются размеры экрана

устройства, заданные выходным профилем, если профиль не задан, и в этом случае он остается в том значении, которое было назначено (по умолчанию 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

Количество одновременных загрузок. Установите 1, если сервер придирчив. Автоматически уменьшается до 1, если *BasicNewsRecipe.delay* (страница 48) > 0

summary_length = 500

Максимальное количество символов в кратком описании

```
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]'

Строка формата даты, показанной на первой странице. По умолчанию: Day_Name, Day_Number Month_Name Year

timeout = 120.0

Таймаут для загрузки файлов с сервера в секундах

title = 'Неизвестный новостной ресурс'

Название для электронной книги

use_embedded_content = None

Обычно попытка угадать, есть ли в фиде встроенные полные статьи, основывается на длине встроенного контента. Если «Нет», то используется угадывание по умолчанию. Если True - всегда предполагается, что у каналов есть встроенный контент, False - всегда предполагается, что у каналов нет встроенного контента.

[12] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[13] https://pythonhosted.org/feedparser/
[14] https://pythonhosted.org/feedparser/reference-entry-link.html
[15] https://mechanize.readthedocs.io/en/latest/
[16] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[17] https://www.imagemagick.org/script/color.php

ГЛАВА 3

Просмотрщик электронных книг

В calibre встроен просмотрщик, поддерживающий все основные форматы электронных книг. Он гибко настраивается и имеет множество дополнительных функций.

- *Запуск просмотрщика электронных книг* (страница 54)
- *Навигация по электронной книге* (страница 54)
- *Выделение текста* (страница 55)
- *Чтение вслух* (страница 55)
- *Поиск текста* (страница 56)
- *Переход по ссылкам только с клавиатуры* (страница 56)
- *Настройка внешнего вида по вашим предпочтениям* (страница 56)
- *Поиск в словаре* (страница 57)
- *Копирование текста и изображений* (страница 57)
- *Увеличение масштаба изображений* (страница 57)
- *Синхронизация с бумажным изданием текущей книги* (страница 57)
- *Горячие клавиши* (страница 57)
- *Неперетекающий контент* (страница 59)
- *Проектирование вашей книги для хорошего просмотра в просмотрщике электронных книг calibre* (страница 60)

3.1 Запуск просмотрщика электронных книг

Чтобы просмотреть или прочитать любую из книг в вашей библиотеке calibre, выделите её и нажмите кнопку *Просмотр*, и она откроется в просмотрщике. Также можно запустить сразу сам просмотрщик из меню «Пуск» в Windows. В macOS можно прикрепить её к доку и запускать оттуда, а в Linux — воспользоваться кнопкой запуска в меню рабочего стола или выполнить команду **ebook-viewer**.

3.2 Навигация по электронной книге

«Перелистывать страницы» книги можно:

- нажатием на левое или правое поле страницы мышкой;
- нажатием клавиш `пробел`, `page up`, `page down` или клавиш со стрелками;
- коснувшись текста или проведя пальцем влево и вправо — для сенсорных экранов.

Доступ к управлению просмотрщиком осуществляется:

- правым щелчком по тексту;
- нажатием клавиш `Esc` или `Menu`;
- нажатием на верхнюю треть экрана — для сенсорных экранов.

У просмотрщика есть два режима, «страничный» и «потоковый». В страничном режиме текст отображается в виде страниц, как в бумажных книгах, в потоковом — показывается непрерывно, как в веб-браузере. Переключаться между ними можно через «Параметры» просмотрщика в *Настройки* раздел *Макет страницы* или сочетанием клавиш `Ctrl+M`.

3.2.1 Закладки

Если вы, прочтя половину книги, закроете просмотрщик, он запомнит то место, где вы остановились, и вернётся к нему, когда вы откроете книгу в следующий раз. Также в книге можно делать закладки с помощью кнопки *Закладки* в меню управления просмотрщиком или сочетанием клавиш `Ctrl+B`. Для книг в формате EPUB закладки сохраняются прямо в файле EPUB. Вы можете, добавив закладки, переслать файл друзьям. Когда они его откроют, то увидят и ваши закладки. Отключается эта возможность в разделе *Разное* в параметрах просмотрщика.

3.2.2 Оглавление

Если в книге, которую вы читаете, есть оглавление, то получить к нему доступ можно, нажав кнопку *Оглавление*, после чего откроется список разделов книги. Нажав на любой из них, вы переместитесь в соответствующее место книги.

3.2.3 Навигация по местоположению

Электронные книги, в отличие от бумажных, не имеют понятия о страницах. Вы можете ссылаться на точные места в электронных книгах с помощью функции *Перейти → Место* в элементах управления просмотрщиком.

Вы можете использовать эту информацию о местоположении, чтобы однозначно ссылаться на части книг, когда обсуждаете её с друзьями или ссылаетесь на неё в других работах. Можно ввести эти местоположения в *Перейти → Место* в элементах управления просмотрщиком.

Это URL-адрес, который вы можете скопировать в буфер обмена и вставить в другие программы или документы. Щелчок по этому URL-адресу откроет книгу в просмотрщике электронных книг calibre в текущем местоположении.

Если вы нажмете на ссылки внутри электронной книги, чтобы перейти к различным частям книги, например к сноске, вы можете использовать кнопки *Назад* и *Вперед* в верхнем левом углу просмотрщика. Эти кнопки ведут себя так же, как кнопки в веб-браузере.

3.2.4 Режим отсылок

У calibre также есть очень удобный *Режим отсылок*. Вы можете включить его, нажав кнопку *Режим отсылок* в элементах управления просмотрщиком. Как только вы это сделаете, в начале каждого абзаца будет отображаться уникальный номер, состоящий из номеров разделов и абзацев.

Этот номер используется, чтобы иметь точные отсылки к частям книг, когда обсуждаете их с друзьями или ссылаетесь на них в других работах. Вы можете вводить такие номера в поле *Перейти к*, чтобы переместиться к конкретному месту, на которое ссылаетесь.

3.3 Выделение текста

Когда вы выделяете текст в средстве просмотра, рядом с выбранным фрагментом появляется небольшая всплывающая панель. Вы можете нажать кнопку выделения на этой панели, чтобы создать выделение. Вы можете добавлять заметки и изменять цвет выделения. На сенсорном экране нажмите и удерживайте слово, чтобы выбрать его и отобразить всплывающую панель. Находясь в режиме выделения, вы можете изменить выбранный текст, используя удобные ручки выбора на сенсорном экране. Перетащите маркеры к верхнему или нижнему полю для прокрутки при выборе. Вы также можете удерживать клавишу `Shift+click` или `right click`, чтобы расширить выделение, что особенно полезно для многостраничного выделения.

Вы можете использовать кнопку *Выделения* в элементах управления средства просмотра, чтобы отобразить отдельную панель со списком всех выделений в книге, отсортированных по главам.

Посмотреть *все выделения* в библиотеке calibre можно, щёлкнув правой кнопку *Просмотр* и выбрав *Просмотреть заметки*.

Наконец, если вы пользуетесь браузерной читалкой контент-сервера calibre, то можете синхронизировать с ней заметки основной читалки, перейдя там в *Параметры → Разное* и введя имя пользователя читалки контент-сервера для синхронизации. Воспользуйтесь специальным значением `*`, чтобы синхронизироваться с анонимными пользователями.

3.4 Чтение вслух

Просмотрщик может озвучивать текст книги. Чтобы использовать его, вы можете просто нажать кнопку *Читать вслух* в элементах управления средства просмотра. Слово, которое в данный момент читается, будет выделено. Речь синтезируется из текста с помощью служб вашей операционной системы для преобразования текста в речь или Piper[18] (нейронный движок преобразования текста в речь). Вы можете изменить используемый голос, щелкнув по значку шестеренки на панели, которая отображается, когда опция *Читать вслух* активна.

Вы также можете прочитать вслух выделенные отрывки, добавив кнопку :guilabel: *Читать вслух* на полосу выбора в настройках просмотрщика в разделе :guilabel:` Поведения выделения`.

Примечание

Поддержка преобразования текста в речь в браузерах очень неполная и изобилует ошибками, поэтому то, насколько хорошо *Read aloud* будет работать в программе просмотра в браузере, зависит от того, насколько хорошо базовый браузер поддерживает преобразование текста в речь.

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

3.5 Поиск текста

Просмотрщик имеет очень мощные возможности поиска. Нажмите клавишу `Ctrl+F` или откройте элементы управления просмотрщиком и щёлкните по поиску. Простейшая форма поиска — это просто поиск любого текста, который вы вводите в текстовое поле. Различные формы поиска выбираются в поле режима поиска под полем поиска. Доступные режимы:

1. *Содержит* - Самый простой режим по умолчанию. Текст, введенный в поле поиска, ищется где угодно. Все знаки препинания, ударения и пробелы игнорируются. Например, поиск: `Pena` будет соответствовать всем следующим: `penal, pen a, pen.a и Peña`. Если вы установите флажок *С учетом регистра*, акценты, пробелы и знаки препинания больше не будут игнорироваться.

2. *Целые слова* - Поиск целых слов. Так, например, при поиске `pena` будет найдено слово `Peña`, но не слово `Penal`. Как и в случае с поиском *Содержит*, указанным выше, акценты и знаки препинания игнорируются, если не установлен флажок *С учетом регистра*.

3. *Соседние слова* - Ищет целые слова, которые находятся рядом друг с другом. Так, например, поисковый запрос `calibre cool` будет соответствовать местам, где слова `calibre` и `cool` встречаются в пределах шестидесяти символов друг от друга. Чтобы изменить количество символов, добавьте новое число в конец списка слов. Например, `calibre cool awesome 120` будет соответствовать местам, где три слова встречаются в пределах 120 символов друг от друга. Обратите внимание, что знаки препинания и диакритические знаки *не* игнорируются при таком поиске.

4. *Regex* - Интерпретирует искомый текст как *регулярное выражение*. Чтобы узнать больше об использовании регулярных выражений, см. *учебник* (страница 221).

3.6 Переход по ссылкам только с клавиатуры

В просмотрщике есть :guilabel: *Режим подсказок*, позволяющий вам нажимать ссылки в тексте без использования мыши. Нажмите клавишу `Alt + F`, и все ссылки на текущем экране будут выделены цифрой или буквой над ними. Нажмите букву на клавиатуре, чтобы перейти по ссылке. Нажатие клавиши :kbd: *Esc* прервёт режим *Подсказки* без выбора какой-либо ссылки.

Если на экране более тридцати пяти ссылок, то некоторые из них будут иметь несколько букв. В этом случае введите первую и вторую или первую и нажмите `Enter` для активации. Вы также можете использовать клавишу `Backspace`, чтобы исправить ошибку при наборе текста.

3.7 Настройка внешнего вида по вашим предпочтениям

Размер шрифта можно менять на лету через *Размер шрифта* в меню управления просмотрщика, или сочетаниями клавиш `Ctrl++` и `Ctrl+-`, или, зажав клавишу `Ctrl`, крутить колёсико мыши.

Цвет можно изменить в разделе *Цвета* в параметрах просмотрщика.

Количество страниц на экране, а также размер полей можно изменить в разделе *Макет страницы* в параметрах просмотрщика.

Вы можете отображать пользовательские верхние и нижние колонтитулы, такие как оставшееся время для чтения, название текущей главы, положение книги и т. д., через раздел *Верхние и нижние колонтитулы* в настройках просмотрщика.

Более продвинутую настройку можно выполнить с помощью настроек *Стили*. Здесь вы можете указать фоновое изображение для отображения под текстом, а также таблицу стилей, которая будет применяться к каждой книге. Используя CSS, вы можете изменять стили абзаца, выравнивать текст и т. д. Примеры таблиц стилей от пользователей calibre, см. на форумах[19].

[19] https://www.mobileread.com/forums/showthread.php?t=51500.

3.8 Поиск в словаре

Вы можете узнать значение слов в текущей книге, дважды щелкнув или нажав и удерживая слово, которое вы хотите найти, а затем нажав кнопку поиска, которая выглядит как библиотека.

3.9 Копирование текста и изображений

Вы можете выбрать текст и изображения, перетаскивая содержимое мышью, а затем щёлкнув правой кнопкой мыши и выбрав *Копировать* для копирования в буфер обмена. Скопированный материал может быть вставлен в другое приложение в виде простого текста и изображений.

3.10 Увеличение масштаба изображений

Вы можете увеличить изображение, чтобы посмотреть его в полном размере в отдельном окне, двойным щелчком или долгим касанием. Также можно щёлкнуть по нему правой кнопкой мыши и выбрать *Рассмотреть изображение*.

3.11 Синхронизация с бумажным изданием текущей книги

Некоторые электронные книги, имеющие соответствующие печатные издания, включают метаданные, которые отмечают начало каждой бумажной страницы. Для таких электронных книг программа просмотра позволяет вам перейти на определенную страницу бумажного издания с помощью кнопки *Перейти к* в элементах управления программой просмотра. Вы также можете при желании отобразить бумажную страницу, соответствующую текущему местоположению в верхних или нижних колонтитулах книги, с помощью настроек просмотра, добавив :guilabel: *Страницы из бумажного издания* либо в верхний, либо в нижний колонтитул.

3.12 Горячие клавиши

Просмотрщик имеет обширные сочетания клавиш, как и остальная часть calibre. Их можно настроить в *Настройки*. Умолчальные сочетания клавиш ниже:

Таблица 1: Сочетания клавиш для просмотрщика calibre

Сочетание клавиш	Действие
`Home`, `Ctrl+Ar` `Ctrl+Ar`	Прокрутить до начала текущего файла в многофайловой книге
`Ctrl+Hc`	Прокрутить до начала книги
`Ctrl+En`	Прокрутить до конца книги
`End`, `Ctrl+Ar` `Ctrl+Ar`	Прокрутить до конца текущего файла в многофайловой книге
`ArrowUp`	Прокрутить назад, плавно в потоковом режиме и на полный экран в страничном
`ArrowDc`	Прокрутить вперёд, плавно в потоковом режиме и на полный экран в страничном

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Сочетание клавиш	Действие
ArrowLe	Прокрутить левее: слегка — в потоковом режиме и на страницу — в страничном
ArrowRi	Прокрутить правее: слегка — в потоковом режиме и на страницу — в страничном
PageUp, Shift+S	Прокрутить на полный экран назад
PageDow Spaceba	Прокрутить на полный экран вперёд
Ctrl+Pa	Перейти к предыдущему разделу
Ctrl+Pa	Перейти к следующему разделу
Alt+Arr	Назад
Alt+Arr	Вперёд
Ctrl+T	Переключить оглавление
Ctrl+S	Чтение вслух
Alt+P	Быстро меняйте настройки, создавая и переключаясь на *profiles*
Alt+f	Переходить по ссылкам с клавиатуры
Ctrl+C	Копировать в буфер
Alt+C	Копировать текущее местоположение в буфер обмена
Ctrl+Sh	Скопировать текущее местоположение как calibre:// URL в буфер обмена
/, Ctrl+f, Cmd+f	Начать поиск
F3, Enter	Найти следующее
Shift+F Shift+E	Найти предыдущее
Ctrl+Pl Meta+Pl	Увеличить размер шрифта
Ctrl+Mi Meta+Mi	Уменьшить размер шрифта
Ctrl+0	Восстановить размер шрифта по умолчанию
Ctrl+]	Увеличить количество страниц на экране
Ctrl+[	Уменьшить количество страниц на экране
Ctrl+Al	Автоматически определять количество страниц на экране
F11, Ctrl+Sh	Переключить полноэкранный режим
Ctrl+M	Переключиться между страничным и потоковым режимами
Ctrl+W	Переключить полосу прокрутки
Ctrl+X	Переключить режим отсылок
Ctrl+B	Показать/скрыть закладки
Ctrl+Al	Новая закладка
Ctrl+N, Ctrl+E	Показать метаданные книги

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Сочетание клавиш	Действие
Ctrl+Al	Перезагрузить книгу
Ctrl+Al	
Ctrl+Sh	Изменить текущее выделение на слово вперёд
Ctrl+Sh	Изменить текущее выделение на слово назад
Shift+A	Изменить текущий выбор вперед на символ
Shift+A	Изменить текущий выбор назад на символ
Shift+A	Изменить текущий выбор на строку вперед
Shift+H	Расширить текущий выбор до начала строки
Shift+E	Расширить текущий выбор до конца строки
Ctrl+A	Выделить всё
Shift+A	Изменить текущий выбор назад на строку
Ctrl+Sh	Изменить текущий выбор вперед на абзац
Ctrl+Sh	Изменить текущий выбор назад на абзац
Esc, MenuKey	Показать элементы управления просмотрщиком электронных книг
Ctrl+Cc	Показать параметры просмотрщика электронных книг
Ctrl+Es	
Meta+Es	
Meta+Cc	
Ctrl+G, ;, :	Перейти к определённому месту в книге
Ctrl+Sp	Переключить автопрокрутку
Alt+Arr	Ускорить автопрокрутку
Alt+Arr	Замедлить автопрокрутку
Ctrl+I	Показать/скрыть инспектор
Ctrl+L	Показать/скрыть панель поиска слов
Ctrl+Q (Cmd+Q on macOS)	Выйти
Ctrl+P	Печать книги в PDF
Ctrl+F1	Переключить панель инструментов
Ctrl+H	Переключить панель выделений
Ctrl+D	Редактировать эту книгу

3.13 Неперетекающий контент

Некоторые книги имеют очень широкий контент, который невозможно разбить по границам страницы. Например, таблицы или теги <pre>. В таких случаях вам следует переключить просмотрщик в *режим потока*, нажав Ctrl + M, чтобы прочитать это содержимое. В качестве альтернативы можно добавить следующий CSS-код в раздел *Styles* настройках просмотрщика, для разбивания строк текста в тегах <pre>:

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

3.14 Проектирование вашей книги для хорошего просмотра в просмотрщике электронных книг calibre

Просмотрщик calibre установит класс `is-calibre-viewer` для корневого элемента. Поэтому можно написать правила CSS, применимые только к нему. Кроме того, просмотрщик установит следующие классы для элемента `body`:

`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`
: Устанавливается в постраничном режиме

`body.calibre-viewer-scrolling`
: Устанавливается в потоковом режиме (без разбивки на страницы)

`body.calibre-footnote-container`
: Установить при отображении всплывающей сноски

Наконец, вы можете использовать цвета цветовой схемы calibre через CSS-переменные[20]. Просмотрщик calibre определяет следующие переменные: *–calibre-viewer-background-color*`, `--calibre-viewer-foreground-color` и необязательно `--calibre-viewer-link-color` в цветовых темах, определяющих цвет ссылки.

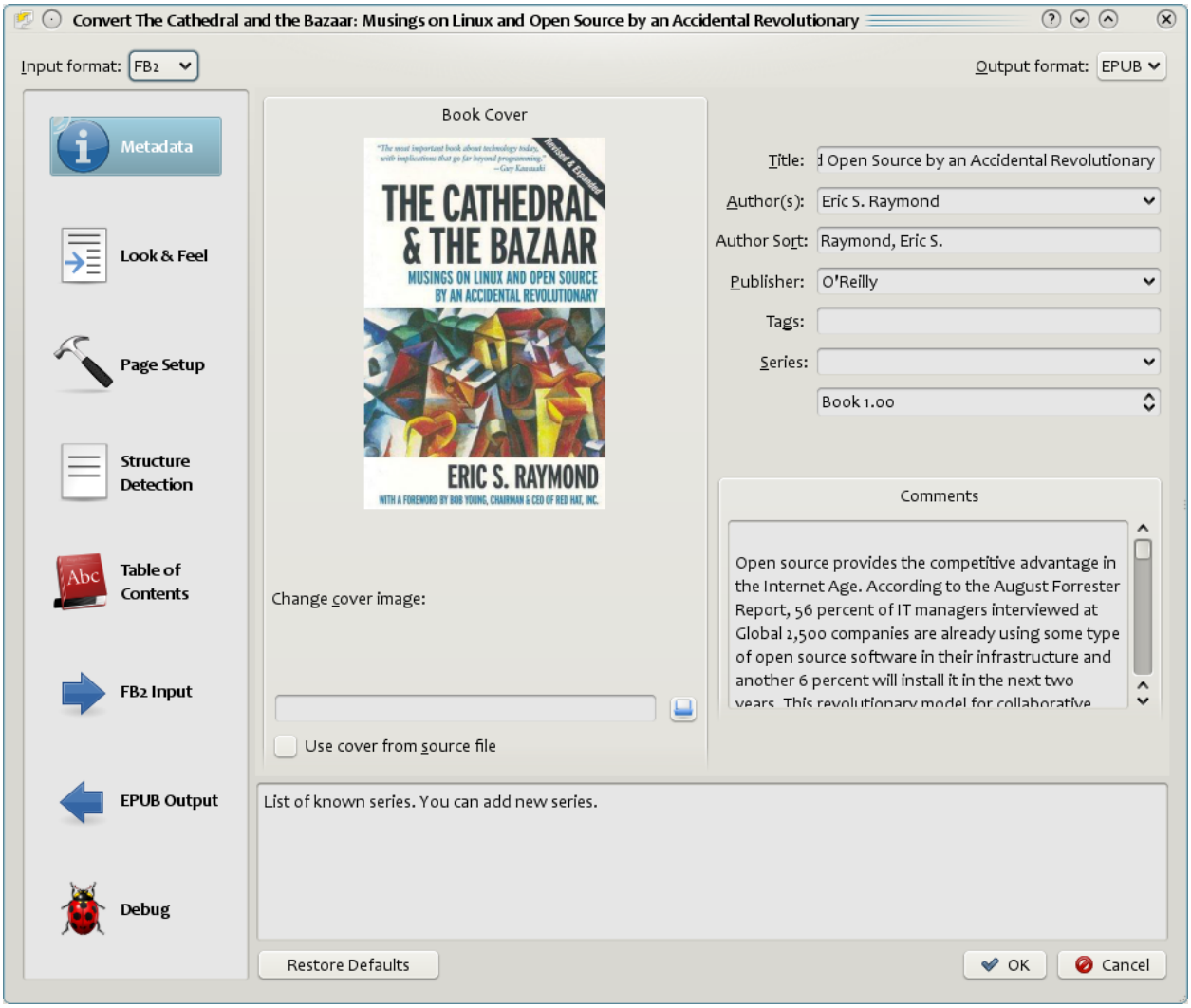
[20] https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties

ГЛАВА 4

Конвертирование книг

Calibre имеет очень простую в использовании систему конвертации. Обычно вы просто добавляете книгу в Calibre, нажимаете «конвертировать», и Calibre изо всех сил старается генерировать выходные данные, максимально приближенные к входным. Тем не менее, Calibre принимает очень большое количество входящих форматов, не все из которых подходят для конвертации в электронные книги. В случае таких входящих форматов или если вы просто хотите лучше контролировать систему конвертации, у Calibre есть много опций для точной настройки процесса конвертации. Однако обратите внимание, что система конвертации Calibre не заменяет полноценного редактора электронных книг. Чтобы редактировать электронные книги, я рекомендую сначала преобразовать их в EPUB или AZW3 с использованием Calibre, а затем с помощью функции *Edit book*, привести их в идеальную форму. Затем вы можете использовать отредактированную электронную книгу в качестве входящих данных для конвертации в другие форматы в Calibre.

Этот документ, в основном, будет ссылаться на настройки, находящиеся в диалоге конвертации, изображённом ниже. Все эти настройки также доступны через интерфейс командной строки для конвертации, задокументированный в *ebook-convert* (страница 335). В calibre можно получить справку по любому отдельному параметру, удерживая над ним курсор мыши до появления подсказки с описанием.



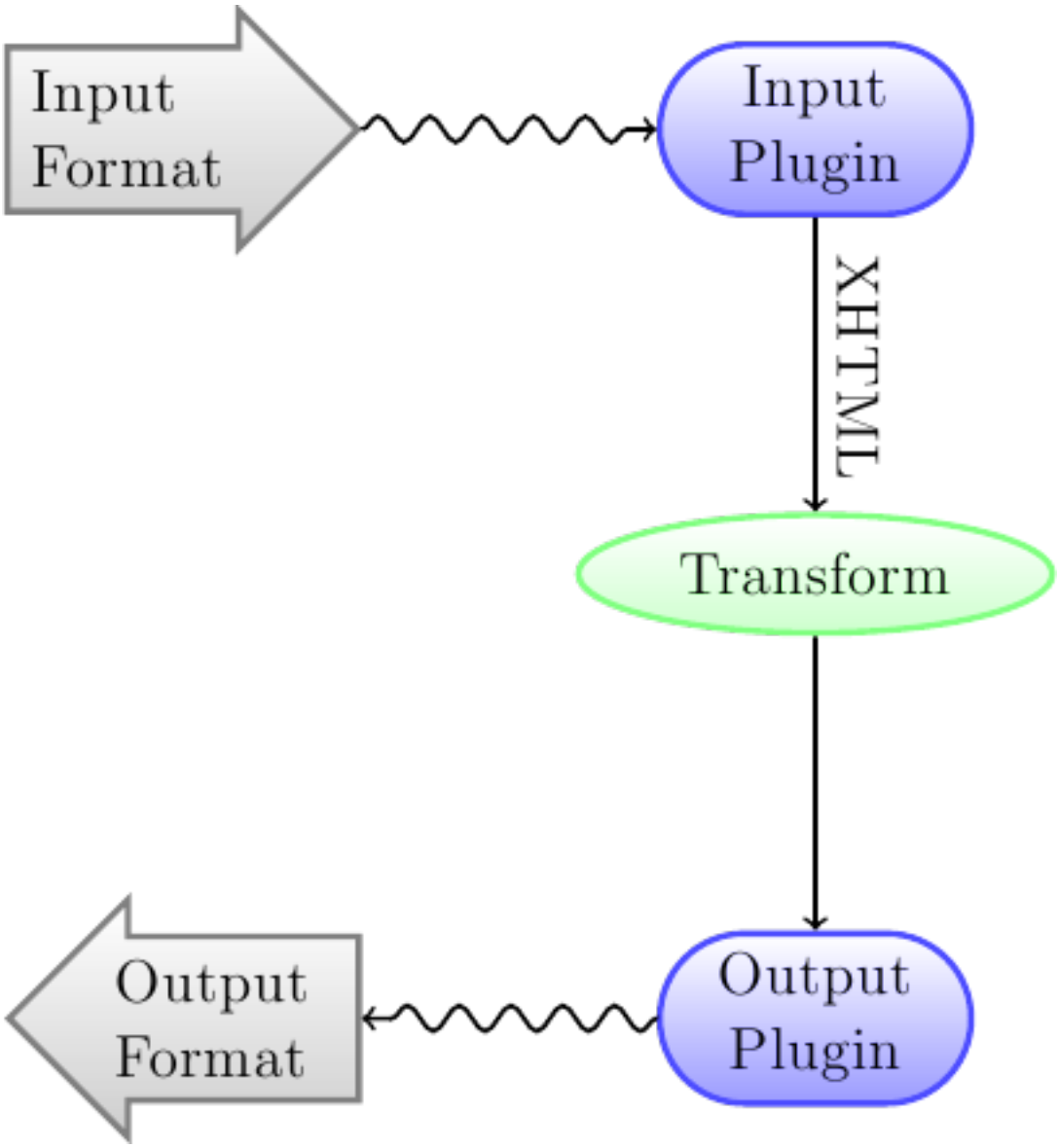
Содержание

- *Предисловие* (страница 63)
- *Оформление* (страница 64)
- *Настройка страницы* (страница 67)
- *Эвристическая обработка* (страница 67)
- *Поиск и замена* (страница 69)
- *Определение структуры* (страница 70)
- *Содержание* (страница 71)
- *Использование изображений в качестве заголовков глав при преобразовании входных документов HTML* (страница 72)
- *Использование атрибутов тега для предоставления текста для записей в оглавлении* (страница 73)
- *Как параметры устанавливаются/сохраняются для конвертации* (страница 73)

- *Формат-специфичные советы* (страница 74)

4.1 Предисловие

Первое, что нужно понять о системе конвертации, это то, что она спроектирована как конвейер. Схематически это выглядит так:



Формат ввода сначала конвертируется в XHTML с помощью соответствующего *плагина ввода*. Этот HTML затем *трансформируется*. На последнем шаге обработанный XHTML преобразуется в указанный формат вывода соответствующим *плагином вывода*. Результаты преобразования могут сильно различаться в зависимости от формата ввода. Список лучших исходных форматов для конвертации доступен: ref:*here<best-source-formats*>.

Конвертации, которые действуют на вывод XHTML, - это место, где происходит вся работа. Существуют различные конвертации: вставка метаданных книги в виде страницы в начало книги, обнаружение заголовков глав и автосоздание оглавления, пропорциональная настройка размеров шрифта и т. д. Все конвертации действуют на

вывод XHTML с помощью *плагина ввода*, а не на сам входной файл. При конвертации calibre-ом RTF-файла в EPUB, сначала он конвертируется во внутренний XHTML, к XHTML применяются различные конвертации, а затем плагин *Вывод* создаёт файл EPUB, автоматически генерируя для файла все метаданные, оглавление и т. д.

Вы можете увидеть этот процесс в действии, используя параметр отладки . Укажите путь к папке для вывода отладки. Во время конвертации calibre помещает XHTML, сгенерированный на различных этапах конвейера конвертации, в разные подпапки. Четыре подпапки:

Таблица 1: Этапы конвейера конвертации

Папка	Описание
ввод	Содержит HTML-вывод плагина ввода. Для отладки плагинов ввода.
разобранный	Результат предварительной обработки и конвертации в XHTML вывода из плагина ввода. Для отладки детектирования структуры.
структура	После обнаружения структуры, но перед сведением CSS и преобразованием размеров шрифта. Для отладки конвертации размера шрифта и преобразований CSS.
обработано	Перед тем как книга передается в плагин вывода. Для отладки плагинов вывода.

Если вы хотите немного отредактировать входной документ до того, как calibre преобразовал его, лучше всего отредактировать файлы в подпапке `input`, затем заархивировать его и использовать ZIP-файл в качестве входного формата для последующих преобразований. Для этого используйте диалоговое окно *Редактировать метаданные*, чтобы добавить ZIP-файл в качестве формата для книги, а затем в верхнем левом углу диалогового окна конвертации выберите ZIP в качестве входного формата.

Этот документ будет посвящен главным образом различным преобразованиям, которые работают с промежуточным XHTML, и тому, как их контролировать. В конце приведены некоторые советы, специфичные для каждого формата ввода/вывода.

4.2 Оформление

Содержание

- *Шрифты* (страница 65)
- *Текст* (страница 66)
- *Макет* (страница 66)
- *Стилизация* (страница 66)
- *Стили трансформации* (страница 67)
- *Трансформировать HTML* (страница 67)

Эта группа параметров контролирует различные аспекты внешнего вида конвертированной электронной книги.

4.2.1 Шрифты

Одной из самых приятных особенностей электронного чтения является возможность легко регулировать размеры шрифта под свои потребности и условия освещения. У calibre - сложные алгоритмы вывода, для гарантии одинаковых размеров шрифта книг, независимо от того, какие размеры шрифта указаны во входном документе.

Базовый размер шрифта документа - это наиболее распространенный размер шрифта в документе, т. е. размер основной части текста в этом документе. При указании: guilabel: *Базовый размер шрифта*, calibre пропорционально автоизменяет размеры всех шрифтов в документе, наиболее распространенный размер шрифта становится указанным базовым размером, а размеры других шрифтов отсчитываются от базового. Больший базовый размер шрифта - увеличение всех шрифтов в документе и наоборот. При установке базового размера шрифта также нужно установить ключ размера шрифта.

Обычно calibre автоматически выбирает основной размер шрифта, соответствующий выбранному вами выходному профилю (см .:ref:*page-setup*). Значение по умолчанию можно переопределить здесь.

Опция *Font size key* позволяет контролировать масштабирование размеров неосновных шрифтов. Алгоритм масштабирования шрифта работает с использованием ключа размера шрифта (просто список разделенных запятыми размеров шрифтов). Ключ размера шрифта сообщает calibre, на сколько «шагов» больше или меньше данного размера шрифта следует сравнивать с базовым размером шрифта. В документе должно быть ограниченное количество размеров шрифта. Один размер для основного текста, несколько размеров для разных уровней заголовков и несколько размеров для индексов super/sub и сносок. Ключ размера шрифта позволяет calibre разделить размеры шрифта во входных документах на отдельные «ячейки», соответствующие различным логическим размерам шрифта.

Давайте проиллюстрируем на примере. Предположим, исходный документ, который мы конвертируем, был создан кем-то с отличным зрением и имеет базовый размер шрифта 8pt. Это означает, что основная часть текста в документе имеет размер 8 пт, а заголовки несколько больше (скажем, 10 и 12 пт), а сноски несколько меньше - 6 пт. Теперь, если мы используем следующие настройки:

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

Выходной документ будет иметь основной размер шрифта 12pt, заголовки 14 и 16pt и сноски 8pt. Теперь предположим, что мы хотим выделить самый большой размер заголовка и сделать сноски немного больше. Для этого ключ шрифта должен быть изменен на:

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

Самые большие заголовки теперь станут 18pt, а сноски - 9pt. Вы можете поиграть с этими настройками, чтобы попытаться выяснить, что будет оптимальным для вас, используя мастер изменения размера шрифта, к которому можно перейти, нажав маленькую кнопку рядом с настройкой guilabel:*Font size key*.

Здесь также можно отключить изменение размера шрифта при преобразовании, если вы хотите сохранить размеры шрифта во входном документе.

Связанная настройка *Высота строки*. Высота строки управляет вертикальной высотой строк. По умолчанию (высота строки 0) никакие манипуляции с высотой строки не выполняются. Если вы укажете значение, отличное от значения по умолчанию, высота строк будет установлена во всех местоположениях, которые не указывают свои собственные высоты строк. Тем не менее, это что-то вроде тупого оружия и должно использоваться с осторожностью. Если вы хотите настроить высоту строки для какого-то раздела ввода, лучше использовать *Extra CSS* (страница 66).

В этом разделе вы также можете указать calibre встраивание любых ссылочных шрифтов в книгу. Это позволит работать со шрифтами на читалках, даже если их нет на устройстве.

4.2.2 Текст

Текст может быть выровненным или нет. Выровненный текст содерит избыточные пробелы между словами, чтобы получить ровный правый край. Некоторые люди предпочитают выровненный текст, другие нет. Обычно calibre сохраняет выравнивание в оригинальном документе. Если вы хотите переопределить его, вы можете использовать опцию *Выравнивание текста* в этом разделе.

Вы также можете сказать calibre :guilabel: *Smarten пунктуация*, который заменит простые кавычки, тире и многоточия их типографически правильными альтернативами. Обратите внимание, что этот алгоритм не идеален, поэтому стоит проверить результаты. Также доступно обратное, а именно: guilabel:*Unsmarted пунктуация*.

Наконец, есть *Кодировка входящих символов*. В старых документах иногда не указывается кодировка символов. При конвертировании это может привести к повреждению неанглийских символов или специальных символов, таких как смарт-кавычки. Calibre пытается автоматически определить кодировку символов исходного документа, но это не всегда удается. Вы можете заставить его принять определенную кодировку символов с помощью этого параметра. *cp1252* - это обычная кодировка документов, созданных с использованием программного обеспечения Windows. Вы также должны прочитать *Как конвертировать мой файл, если он включает не английские символы или «умные» кавычки?* (страница 132), чтобы узнать больше о проблемах кодирования.

4.2.3 Макет

Обычно абзацы в XHTML рендерятся с пустой строкой между ними без начального отступа текста. Calibre может конролировать это несколькими вариантами. *Убрать интервалы между абзацами* принудительно убедиться, что у всех абзацев нет интервала между абзацами. Также устанавливается отступ текста на 1,5em (можно изменить), чтобы отметить начало каждого абзаца. :guilabel: *Вставить пустую строку* делает противоположное, гарантируя, что между каждой парой абзацев ровно одна пустая строка. Обе эти опции всеобъемлющие, убирают пробел или вставляют его для *всех* абзацев (технически <p> и тегов). Это делается для того, чтобы вы могли просто установить параметр и быть уверенным, что он работает так, как объявлено, независимо от того, насколько грязен входной файл. Единственное исключение - когда входной файл использует жесткие разрывы строк для реализации межпараграфного интервала.

Если вы хотите удалить интервал между всеми абзацами, кроме нескольких избранных, не используйте эти параметры. Вместо этого добавьте следующий код CSS в *Extra CSS* (страница 66):

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

Затем в исходном документе отметьте абзацы, требующие интервала, с помощью *class=»spacious»*. Если ваш входной документ не в формате HTML, используйте опцию Debug, описанную во введении, чтобы получить HTML (используйте подпапку `input`).

Ещё одна полезная опция *Линеаризовать таблицы*. Некоторые плохо оформленные документы используют таблицы для управления макетом текста на странице. При преобразовании в этих документах часто появляется текст, который выходит за пределы страницы и другие артефакты. Эта опция будет извлекать содержимое из таблиц и представлять его в линеаризованной форме. Обратите внимание, что этот параметр линеаризует *все* таблицы, поэтому используйте его, только если вы уверены, что входной документ не использует таблицы для законных целей, таких как представление табличной информации.

4.2.4 Стилизация

Опция *Extra CSS* позволяет вам указать произвольный CSS, который будет применяться ко всем HTML-файлам во входных данных. Этот CSS-код применяется с очень высоким приоритетом и поэтому должен перекрывать большую часть CSS-кода, присутствующего во **входном документе**. Вы можете использовать этот параметр для точной настройки презентации/макета вашего документа. Например, если вы хотите, чтобы все абзацы класса *endnote* были выровнены по правому краю, просто добавьте

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

или если вы хотите изменить отступ всех абзацев:

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

Extra CSS - очень мощный вариант, но вам нужно понять, как работает CSS, чтобы использовать его в полной мере. Вы можете использовать опцию конвейера отладки, описанную выше, чтобы увидеть, какой CSS присутствует во входном документе.

Более простой вариант - использовать *Фильтровать информацию о стиле*. Это позволяет удалить все свойства CSS указанных типов из документа. Например, вы можете использовать его, чтобы удалить все цвета или шрифты.

4.2.5 Стили трансформации

Это самое мощное средство для стилизации. Вы можете использовать его для определения правил, которые меняют стили в зависимости от условий. Например, вы можете использовать его, чтобы изменить все входения зелёного цвета на синий, или удалить все полужирные стили из текста или раскрасить все заголовки определённым цветом и т.д.

4.2.6 Трансформировать HTML

Подобно стилям трансформации, но позволяет вносить изменения в HTML-содержимое книги. Вы можете заменить один тег другим, добавить классы или другие атрибуты к тегам в зависимости от их содержимого и т. д.

4.3 Настройка страницы

Параметры *Page setup* предназначены для управления макетом экрана, например, полями и размерами экрана. Существуют варианты настройки полей страницы, которые будут использоваться плагином вывода, если выбранный формат вывода поддерживает поля страницы. Кроме того, вы должны выбрать Входной профиль и Выходной профиль. Оба набора профилей в основном имеют дело с тем, как интерпретировать измерения в документах ввода/вывода, размеры экрана и стандартные ключи изменения масштаба шрифта.

Если вы знаете, что файл, который вы конвертируете, предназначен для использования на определённом устройстве/программной платформе, выберите соответствующий профиль ввода, в противном случае просто выберите профиль ввода по умолчанию. Если вы знаете, что создаваемые вами файлы предназначены для определённого типа устройства, выберите соответствующий выходной профиль. В противном случае выберите один из общих выходных профилей. Если вы конвертируете в MOBI или AZW3, выбирайте один из выходных профилей Kindle. В противном случае для современных устройств для чтения электронных книг лучше всего выбрать выходной профиль *Generic e-ink HD*.

Выходной профиль также контролирует размер экрана. Это приведет, например, к автоматическому изменению размера изображений, чтобы они соответствовали экрану в некоторых выходных форматах. Поэтому выберите профиль устройства, размер экрана которого соответствует вашему устройству.

4.4 Эвристическая обработка

Эвристическая обработка предоставляет множество функций, которые можно использовать для обнаружения и исправления типичных проблем в плохо отформатированных входных документах. Используйте эти функции, если ваш входной документ страдает от плохого форматирования. Поскольку эти функции основаны на общих шаблонах, имейте в виду, что в некоторых случаях один из вариантов может привести к худшим результатам, поэтому используйте его с осторожностью. В качестве примера, некоторые из этих опций будут удалять все объекты без пробелов или могут включать ложно-положительные совпадения, относящиеся к функции.

Включить эвристическую обработку
Эта опция активирует *Эвристическая обработка* конвейера конвертирования. Должно быть включено для применения различных субфункций

Развернуть строки
Включение этой опции приведет к тому, что calibre попытается обнаружить и исправить жесткие разрывы строк, существующие в документе, используя знаки препинания и длину строки. Calibre сначала попытается определить, существуют ли жесткие разрывы строк, если они, по-видимому, не существуют, calibre не будет пытаться развернуть строки. Коэффициент развертывания строки может быть уменьшен, если вы хотите «заставить» calibre развернуть строки.

Коэффициент разворачивания строки
Эта опция контролирует алгоритм, используемый calibre для удаления жёстких разрывов строк. Например, если значение этого параметра равно 0,4, это означает, что calibre удалит жёсткие разрывы строк в конце строк, длина которых меньше, чем длина 40% всех строк в документе. Если в вашем документе есть только несколько разрывов строк, которые нуждаются в исправлении, то это значение следует уменьшить до значения между 0,1 и 0,2.

Обнаружение и разметка неформатированных заголовков и подзаголовков глав
Если ваш документ не имеет заголовков и заголовков глав, отформатированных иначе, чем остальная часть текста, calibre может использовать эту опцию, чтобы попытаться обнаружить их и окружить их тегами заголовков. <h2> теги используются для заголовков глав; <h3> теги используются для любых обнаруженных заголовков.

Эта функция не будет создавать TOC, но во многих случаях она приведет к тому, что настройки обнаружения глав по умолчанию для calibre будут правильно определять главы и создавать TOC. Настройте XPath в разделе «Определение структуры», если оглавление не создается автоматически. Если в документе нет других заголовков, тогда настройка «//h:h2» в разделе «Определение структуры» будет самым простым способом создания оглавления для документа.

Вставленные заголовки не отформатированы, чтобы применить форматирование, используйте параметр *Extra CSS* в настройках преобразования Look and Feel. Например, чтобы центрировать теги заголовков, используйте следующее:

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

Перенумеровать последовательности <h1> или <h2> теги
Некоторые издатели форматируют заголовки глав, используя несколько <h1> или <h2> тегов последовательно. Настройки конвертации по умолчанию приведут к тому, что такие заголовки будут разделены на две части. Эта опция будет перенумеровывать теги заголовков, чтобы предотвратить расщепление.

Удалить пустые строки между абзацами
Эта опция заставит calibre анализировать пустые строки, включенные в документ. Если каждый абзац перемежается пустой строкой, тогда calibre удалит все эти пустые абзацы. Последовательности из нескольких пустых строк будут считаться разрывами сцены и сохраняться как один абзац. Этот параметр отличается от параметра *Удалить интервал между абзацами* в разделе *Look and Feel* тем, что он фактически изменяет содержимое HTML, в то время как другой параметр изменяет стили документа. Эта опция также может удалить абзацы, которые были вставлены с помощью опции calibre *Insert blank line*.

Убедиться, что разрывы сцен последовательно отформатированы
С помощью этой опции calibre попытается обнаружить общие маркеры разрыва сцены и убедиться, что они выровнены по центру. «Мягкие» маркеры разрыва сцены, т. е. Разрывы сцены, определяемые только дополнительным пробелом, имеют стиль, гарантирующий, что они не будут отображаться вместе с разрывами страниц.

Replace scene breaks
Если эта опция настроена, то calibre заменит найденные маркеры разрыва сцены на текст замены, указанный

пользователем. Обратите внимание, что некоторые декоративные символы могут поддерживаться не всеми устройствами чтения.

В общем, вы должны избегать использования тегов HTML, calibre отбрасывает любые теги и использует предопределенную разметку. <hr /> теги, то есть горизонтальные линейки, и <img> теги являются исключениями. Горизонтальные линейки можно опционально указывать с помощью стилей, если вы решите добавить свой собственный стиль, обязательно включите параметр „width“, иначе информация о стиле будет отброшена. Теги изображений могут использоваться, но calibre не предоставляет возможность добавлять изображение во время преобразования, это необходимо сделать после, используя функцию «Редактировать книгу».

Пример тега изображения (поместите изображение в папку „Images“ внутри EPUB после преобразования):

Пример горизонтальной линейки со стилями:

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

Remove unnecessary hyphens

calibre будет анализировать весь контент со знаками переноса в документе, когда эта опция включена. Сам документ используется в качестве словаря для анализа. Это позволяет точно удалять дефисы для любых слов в документе на любом языке, наряду с вымышленными и неясными научными словами. Основной недостаток - слова, появляющиеся только один раз в документе, не будут изменены. Анализ происходит в два прохода, первый проход анализирует окончания строки. Строки развертываются только в том случае, если в документе есть слово с дефисом или без него. Во втором проходе анализируются все дефисные слова по всему документу, дефисы удаляются, если слово существует в другом месте документа без совпадения.

Выделять общие слова и шаблоны курсивом

Когда включено, calibre будет искать общие слова и образцы, которые обозначают курсив и выделять их курсивом. Примерами являются обычные текстовые соглашения, такие как ~слово~ или фразы, которые обычно должны быть выделены курсивом, например латинские фразы типа „etc.“ или „et cetera“.

Заменить существующие отступы CSS-отступами

В некоторых документах используется соглашение об определении отступов текста с использованием неразрывных пробелов. Когда эта опция включена, calibre будет пытаться обнаружить этот вид форматирования и преобразовать их в 3% текстовый отступ, используя CSS.

4.5 Поиск и замена

Эти параметры полезны главным образом для конвертирования документов PDF или преобразования OCR, хотя их также можно использовать для устранения многих специфических проблем документов. Например, некоторые преобразования могут оставлять верхние и нижние колонтитулы в тексте. Эти параметры используют регулярные выражения, чтобы попытаться обнаружить заголовки, нижние колонтитулы или другой произвольный текст и удалить или заменить их. Помните, что они работают на промежуточном XHTML, созданном конвейером конвертации. Существует мастер, который поможет вам настроить регулярные выражения для вашего документа. Нажмите волшебную палочку рядом с полем выражения и нажмите кнопку «Тест» после составления поискового выражения. Успешные совпадения будут выделены желтым цветом.

Поиск работает с использованием регулярных выражений Python. Весь подобранный текст просто удаляется из документа или заменяется с помощью шаблона замены. Шаблон замены является необязательным, если оставить его пустым, текст, соответствующий шаблону поиска, будет удален из документа. Вы можете узнать больше о регулярных выражениях и их синтаксисе по адресу *Всё об использовании регулярных выражений в calibre* (страница 221).

4.6 Определение структуры

Обнаружение структуры заставляет calibre, пытаться обнаружить структурные элементы во входном документе, когда они не определены должным образом. Например, главы, разрывы страниц, верхние и нижние колонтитулы и т. д. Как вы можете себе представить, этот процесс сильно варьируется от книги к книге. К счастью, у calibre есть очень мощные возможности для управления этим. С силой приходит сложность, но если вы потратите время на изучение сложности, вы найдете, что она того стоит.

4.6.1 Главы и разрывы страниц

calibre имеет два набора параметров для *определение главы* и *вставка разрывов страниц*. Иногда это может немного сбивать с толку, так как по умолчанию calibre вставляет разрывы страниц перед обнаруженными главами, а также местоположения, определяемые параметром разрывов страниц. Причиной этого является то, что часто есть места, где следует вставлять разрывы страниц, которые не являются границами глав. Кроме того, обнаруженные главы могут быть дополнительно вставлены в автоматически сгенерированное оглавление.

calibre использует *XPath*, мощный язык, позволяющий пользователю определять границы глав/разрывы страниц. Поначалу XPath может показаться немного сложным, к счастью, в руководстве пользователя есть Учебник по XPath. Помните, что определение структуры работает на промежуточном XHTML, созданном конвейером конвертации. Используйте опцию отладки, описанную в *Предисловие* (страница 63), чтобы выяснить значение соответствующих настроек для вашей книги. Существует также кнопка для мастера XPath, которая поможет с генерацией простых выражений XPath.

По умолчанию calibre использует следующее выражение для определения глав:

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

Это выражение довольно сложное, потому что оно пытается обрабатывать несколько общих случаев одновременно. Это означает, что calibre будет предполагать, что главы начинаются с тегов *<h1>* или *<h2>*, в которых есть любое из слов *(глава, книга, раздел или часть) chapter, book, section or part* или которые имеют атрибут *class=»chapter»*.

Связанная опция *Метка главы*, которая позволяет вам контролировать, что делает calibre при обнаружении главы. По умолчанию он вставляет разрыв страницы перед главой. Вы можете добавить вставленную линию вместо этого или в дополнение к разрыву страницы. Вы также можете этого не делать.

Настройка по умолчанию для обнаружения разрывов страниц:

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

это означает, что calibre будет вставлять разрывы страниц перед каждым тегом *<h1>* и *<h2>* по умолчанию.

> **Примечание**
>
> Выражения по умолчанию могут меняться в зависимости от формата ввода, который вы конвертируете.

4.6.2 Разное

В этом разделе есть еще несколько опций.

Вставить метаданные как страницу в начале книги
: Одна из замечательных особенностей calibre заключается в том, что он позволяет вам хранить полные метаданные обо всех ваших книгах, например, оценку, теги, комментарии и т. д. Эта опция создаст одну страницу со всеми этими метаданными и вставит ее в сконвертированную электронную книгу, как правило,

сразу после обложки. Думайте об этом как о способе создать свою собственную настроенную обложку книги.

Удалить первое изображение
Иногда исходный документ, который вы конвертируете, включает обложку как часть книги, а не как отдельную обложку. Если вы также укажете обложку calibre, то в конвертированной книге будет две обложки. Эта опция просто удалит первое изображение из исходного документа, тем самым гарантируя, что конвертированная книга имеет только одну обложку, ту, которая указана в calibre.

4.7 Содержание

Когда входной документ содержит оглавление в своих метаданных, calibre просто использует его. Однако некоторые старые форматы либо не поддерживают оглавление на основе метаданных, либо отдельные документы не имеют такового. В этих случаях параметры в этом разделе могут помочь вам автоматически сгенерировать оглавление в сконвертированной электронной книге на основе фактического содержания во входном документе.

Примечание

Использование этих опций может быть немного сложным, чтобы получилось абсолютно правильно. Если вы предпочитаете создавать/редактировать оглавление вручную, преобразуйте его в форматы EPUB или AZW3 и установите флажок в нижней части раздела «Оглавление» диалогового окна преобразования, которое гласит :guilabel: «Вручную настроить таблицу содержания после конвертации`. Запустится редактор таблицы содержания после конвертации. Он позволит вам создавать записи в оглавлении, просто щёлкнув место в книге, на которое будет указывать запись. Вы также можете использовать Редактор оглавления самостоятельно, без конвертации. Перейдите по адресу *Настройки → Интефейс → Панели инструментов* и добавьте *ToC редактор* на главную панель инструментов. Затем просто выберите книгу, которую вы хотите отредактировать, и нажмите кнопку *ToC редактор*.

Первый вариант *Принудительно использовать автосгенерированное оглавление*. Отметив эту опцию, вы можете перезаписать любое оглавление, найденное в метаданных входного документа автоматически сгенерированным.

Способ создания автоматически сгенерированного оглавления по умолчанию заключается в том, что calibre сначала попытается добавить любые обнаруженные главы в сгенерированное оглавление. Вы можете узнать, как настроить обнаружение глав в разделе *Определение структуры* (страница 70) выше. Если вы не хотите включать обнаруженные главы в сгенерированное оглавление, отметьте опцию *Не добавлять обнаруженные главы*.

Если было обнаружено меньше глав, чем число *Порог главы*, то calibre добавит любые гиперссылки, найденные во входном документе, в оглавление. Это часто хорошо работает, многие входные документы включают в себя содержание с гиперссылкой в самом начале. Опция *Количество ссылок* может использоваться для управления этим поведением. Если установлено в ноль, ссылки не добавляются. Если установлено значение больше нуля, добавляется самое большое количество ссылок.

calibre будет автоматически фильтровать дубликаты из сгенерированного содержания. Однако, если есть некоторые дополнительные нежелательные записи, вы можете отфильтровать их, используя опцию *TOC Filter*. Это регулярное выражение, которое будет соответствовать заголовку записей в сгенерированном оглавлении. Всякий раз, когда совпадение найдено, оно будет удалено. Например, чтобы удалить все записи заголовков «Next» или «Previous», используйте:

```
Next|Previous
```

Опции *Уровень 1,2,3 TOC* позволяют вам создавать сложные многоуровневые оглавления. Это выражения XPath, которые соответствуют тегам в промежуточном XHTML, созданном конвейером преобразования. Чтобы узнать, как получить доступ к этому XHTML, смотрите *Предисловие* (страница 63). Также прочитайте *Учебник по XPath* (страница 162), чтобы узнать, как создавать выражения XPath. Рядом с каждым параметром находится кнопка,

которая запускает мастер, помогающий создавать основные выражения XPath. В следующем простом примере показано, как использовать эти параметры.

Предположим, у вас есть входной документ с XHTML, который выглядит следующим образом:

```
<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>
```

Затем мы устанавливаем параметры как:

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

Это приведет к автоматически сгенерированному двухуровневому оглавлению, которое выглядит следующим образом:

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

Предупреждение

Не все выходные форматы поддерживают многоуровневое оглавление. Сначала вы должны попробовать с выходом EPUB. Если это работает, попробуйте ваш формат выбора.

4.8 Использование изображений в качестве заголовков глав при преобразовании входных документов HTML

Предположим, вы хотите использовать изображение в качестве названия главы, но все же хотите, чтобы calibre мог автоматически генерировать для вас оглавление из названий глав. Для этого используйте следующую разметку HTML:

```
<html>
    <body>
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        <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>
```

Установите для *Уровень 1 TOC* значение `//h:h2`. Затем, во второй главе, calibre получит заголовок из значения атрибута `title` тега `<h2>`, поскольку у тега нет текста.

4.9 Использование атрибутов тега для предоставления текста для записей в оглавлении

Если у вас особенно длинные заголовки глав и вам нужны сокращенные версии в оглавлении, вы можете использовать атрибут title для достижения этого, например:

```
<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>
```

Установите для *Level 1 TOC* значение `//h:h2/@title`. Тогда calibre получит заголовок из значения атрибута `title` в тегах `<h2>` вместо использования текста внутри тега. Обратите внимание на завершающий `/@title` в выражении XPath, вы можете использовать эту форму, чтобы указать calibre, что нужно получить текст из любого понравившегося вам атрибута.

4.10 Как параметры устанавливаются/сохраняются для конвертации

Есть два места, где параметры конвертации могут быть установлены в calibre. Первый находится в Предпочтения-> Конверсия. Эти настройки являются настройками по умолчанию для параметров конвертации. Всякий раз, когда вы пытаетесь сконвертировать новую книгу, настройки, установленные здесь, будут использоваться по умолчанию.

Вы также можете изменить настройки в диалоге конвертации для каждой конвертации книги. Когда вы конвертируете книгу, calibre запоминает настройки, которые вы использовали для этой книги, так что если вы конвертируете её снова, сохраненные настройки для отдельной книги будут иметь приоритет над значениями по умолчанию, установленными в *Настройки*. Вы можете восстановить отдельные настройки по умолчанию, используя кнопку *Восстановить настройки по умолчанию* в диалоге конвертации отдельной книги. Вы можете удалить сохраненные настройки для группы книг, выбрав все книги и нажав кнопку *Редактировать метаданные*, чтобы открыть диалоговое окно массового редактирования метаданных, в нижней части диалогового окна есть опция для удаления настроек сохраненной конвертации.

Когда вы массово конвертируете набор книг, настройки берутся в следующем порядке (выигрывает последняя):

- Из значений по умолчанию, установленных в Настройки->Конвертация

- Из сохраненных настроек конвертации для каждой конвертируемой книги (если есть). Это можно отключить с помощью параметра в верхнем левом углу диалогового окна «Массовая конвертация».
- Из настроек, установленных в диалоговом окне Массовая конвертация

Обратите внимание, что окончательные настройки для каждой книги в Массовой конвертации будут сохранены и использованы повторно, если книга будет преобразована снова. Так как наивысший приоритет в массовой конвертации преобразовании отдается настройкам в диалоговом окне «Массовая конвертация», они переопределяют любые параметры, специфичные для книги. Таким образом, вы должны конвертировать только те книги, которые нуждаются в похожих настройках. Исключение составляют метаданные и специфичные для входного формата настройки. Поскольку в диалоговом окне «Массовая конвертация» нет настроек для этих двух категорий, они будут взяты из настроек книги (если они есть) или значений по умолчанию.

Примечание

Вы можете увидеть фактические настройки, использованные во время любой конвертации, щелкнув вращающийся значок в правом нижнем углу, а затем дважды щёлкнув на отдельном задании конвертации. Это вызовет журнал конвертаций, который будет содержать фактические используемые настройки в верхней части.

4.11 Формат-специфичные советы

Здесь вы найдете советы, касающиеся конвертации определенных форматов. Параметры, специфичные для конкретного формата, независимо от того, доступны ли вход или выход в диалоге преобразования в их собственном разделе, например, *TXT input* или` EPUB output`.

4.11.1 Конвертация документов Microsoft Word

calibre может автоматически конвертировать файлы `.docx`, созданные в Microsoft Word 2007 и новее. Просто добавьте файл в calibre и нажмите «Конвертировать».

Примечание

Существует :download_file:`demo .docx file <demos/demo.docx>, демонстрирующий возможности механизма конвертации calibre. Просто скачайте его и конвертируйте в EPUB или AZW3, чтобы узнать на что способен calibre.

calibre автоматически сгенерирует оглавление на основе заголовков, если вы пометите свои заголовки стилями «Заголовок 1», «Заголовок 2» и т. д. в Microsoft Word. Откройте выходную электронную книгу в просмотрщике электронных книг calibre и нажмите кнопку *Оглавление*, чтобы просмотреть сгенерированное оглавление.

Старые файлы .doc

Для старых файлов .doc вы можете сохранить документ в формате HTML с помощью Microsoft Word, а затем преобразовать полученный HTML-файл в calibre. При сохранении в формате HTML обязательно используйте параметр «Сохранить как веб-страницу, отфильтрованный», поскольку при этом будет получен чистый HTML, который будет хорошо конвертироваться. Обратите внимание, что Word создает грязный HTML, его преобразование может занять много времени, так что наберитесь терпения. Если у вас есть более новая версия Word, вы также можете напрямую сохранить файл в формате docx.

Другой альтернативой является использование бесплатного OpenOffice/LibreOffice. Откройте файл .doc в OpenOffice/LibreOffice и сохраните его в формате .docx, которыей может напрямую конвертироваться в calibre

4.11.2 Конвертировать TXT документы

Документы TXT не имеют четко определенного способа задания форматирования, такого как полужирный шрифт, курсив и т. д., или структуры документа, такой как абзацы, заголовки, разделы и т. д., но существует множество общепринятых соглашений. По умолчанию calibre пытается автоматически определить правильное форматирование и разметку на основе этих соглашений.

Ввод TXT поддерживает ряд опций, позволяющих различать способы обнаружения абзацев.

Стиль абзаца: Авто
Анализировать текстовый файл и пытаться автоматически определить, как определяются абзацы. Этот параметр обычно работает нормально, если вы достигнете нежелательных результатов, попробуйте один из вариантов вручную.

Стиль абзаца: Блок
Предположить, что одна или несколько пустых строк являются границей абзаца:

```
This is the first.

This is the
second paragraph.
```

Стиль абзаца: Одиночный
Предположить, что каждая строка - абзац:

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

Стиль абзаца: Печать
Предположить, что каждый абзац начинается с отступа (либо табуляция, либо 2+ пробелов). Абзацы заканчиваются, когда достигается следующая строка, начинающаяся с отступа:

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

  This is the
third.
```

Стиль абзаца: Неформатированный
Предположить, что документ не имеет форматирования, но использует жесткие переносы строк. Пунктуация и средняя длина строки используются для воссоздания абзацев.

Стиль форматирования: Авто
Попытаться определить тип используемой разметки форматирования. Если разметка не используется, будет применено эвристическое форматирование.

Стиль форматирования: Эвристика
Анализировать документ на наличие общих заголовков глав, разрывов сцен и выделенных курсивом слов и применить соответствующую разметку HTML во время конвертирования.

Стиль форматирования: Markdown
calibre также поддерживает запуск ввода TXT через препроцессор преобразования, известный как Markdown. Markdown позволяет добавлять базовое форматирование в документы TXT, такие как полужирный шрифт, курсив, заголовки разделов, таблицы, списки, оглавление и т. д. Пометить заголовки глав начальным # и установить для выражения обнаружения XPath главы

значение «//h:h1 «- это самый простой способ получить правильное оглавление, сгенерированное из документа TXT. Вы можете узнать больше о синтаксисе Markdown в daringfireball[21].

Стиль форматирования: Нет
К тексту не применяется никакого специального форматирования, документ конвертируется в HTML без каких-либо других изменений.

4.11.3 Преобразование PDF документов

PDF документы - один из худших форматов для конвертации. Они имеют фиксированный размер страницы и формат размещения текста. То есть очень трудно определить, где заканчивается один абзац и начинается другой. calibre попытается развернуть абзацы с помощью настраиваемого *Line Un-Wrapping Factor*. Это шкала, используемая для определения длины, на которой должна быть развернута строка. Допустимые значения: десятичное число от 0 до 1. Значение по умолчанию - 0,45, чуть меньше длины средней строки. Уменьшите это значение, чтобы включать больше текста в развертывание. Увеличьте, чтобы включать меньше. Вы можете отрегулировать это значение в настройках преобразования в разделе *PDF Input*.

Кроме того, их верхние и нижние колонтитулы часто включаются в основной текст документа. Используйте панель *Поиск и замена* для удаления верхних и нижних колонтитулов, чтобы смягчить эту проблему. Если верхние и нижние колонтитулы не удалены из текста, это может привести к разворачиванию абзаца. Чтобы узнать, как использовать параметры удаления верхнего и нижнего колонтитула, прочитайте *Всё об использования регулярных выражений в calibre* (страница 221).

Некоторые ограничения ввода с PDF:

- Сложные документы с несколькими столбцами и изображениями не поддерживаются.
- Извлечение векторных изображений и таблиц из документа также не поддерживается
- В некоторых PDF-файлах используются специальные глифы для обозначения ll или ff или fi и т. д. Конвертация может их обработать или не обработать в зависимости от того, как они внутренне представлены в PDF.
- Ссылки и оглавления не поддерживаются
- PDF-файлы, в которых для представления неанглоязычных символов используются встроенные шрифты, отличные от Юникода, приведут к искажённому выводу этих символов.
- Некоторые PDF-файлы состоят из фотографий страницы с распознанным текстом OCR позади них. В таких случаях calibre использует распознанный текст, который может сильно отличаться от того, что вы видите при просмотре файла PDF.
- PDF-файлы, которые используются для отображения сложного текста, такого как языки с письмом справа налево и математические наборы, не будут правильно сконвертированы

Повторяю **PDF - это действительно очень плохой** формат для использования в качестве входных данных. Если вам абсолютно необходимо использовать PDF, будьте готовы к выводу в диапазоне от приличного до непригодного для использования, в зависимости от входного PDF.

4.11.4 Коллекции комиксов

Коллекция комиксов - это файл .cbc. Файл .cbc представляет собой ZIP-файл, содержащий другие файлы CBZ/CBR. Кроме того, файл .cbc должен содержать простой текстовый файл с именем comics.txt, закодированный в UTF-8. Файл comics.txt должен содержать список файлов комиксов в файле .cbc в форме имя файла:заголовок, как показано ниже:

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

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

Файл .cbc будет содержать:

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

calibre автоматически преобразует этот файл .cbc в электронную книгу с оглавлением, указывающим на каждую запись в comics.txt.

4.11.5 Демоверсия расширенного форматирования EPUB

Различное расширенное форматирование для файлов EPUB демонстрируется в этом демофайле demo file[22]. Файл был создан из HTML-кода, написанного вручную в calibre, это - шаблон для создания ваших собственных EPUB.

Исходный HTML-код, из которого он был создан, здесь demo.zip[23]. Настройки, используемые для создания EPUB из файла ZIP:

```
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
```

Обратите внимание, что, так как этот файл раскрывает потенциал EPUB, большая часть расширенного форматирования не будет работать на читалках, с меньшими возможностями, чем встроенный в calibre EPUB просмотрщик.

4.11.6 Конвертировать ODT документы

calibre может напрямую конвертировать файлы ODT (OpenDocument Text). Вы должны использовать стили для форматирования вашего документа и минимизировать использование прямого форматирования. При вставке изображений в документ необходимо привязать их к абзацу, изображения, привязанные к странице, в конечном итоге окажутся в начале преобразования.

Чтобы включить автоматическое определение глав, вам необходимо пометить их встроенными стилями *Heading 1*, *Heading 2*, …, *Heading 6* (*Heading 1* соответствует HTML tag `<h1>`, *Heading 2* - `<h2>`, и т.д.). При конвертировании в calibre вы можете указать стиль, который вы использовали, в поле *Определить главы в*. Пример:

- Если вы помечаете главы стилем *Heading 2*, установите для поля „Определить главы по“ значение `//h:h2`
- Для вложенного оглавления с разделами, помеченными как *Heading 2* и разделами, помеченными как *Heading 3*, необходимо ввести `//h:h2|//h:h3`. На странице «Преобразование - TOC» установите для поля *Level 1 TOC* значение `//h:h2`, а для поля *Level 2 TOC* - `//h:h3`.

Хорошо известные свойства документа (Название, Ключевые слова, Описание, Создатель) распознаются, и calibre будет использовать первое изображение (не слишком маленькое и с хорошим соотношением сторон) в качестве изображения обложки.

Существует также расширенный режим преобразования свойств, который активируется установкой настраиваемого свойства `opf.metadata` (тип „Yes или No“) на Yes в вашем документе ODT (Файл->Свойства-

[22] https://calibre-ebook.com/downloads/demos/demo.epub
[23] https://calibre-ebook.com/downloads/demos/demo.zip

>Пользовательские свойства). Если это свойство обнаружено calibre, распознаются следующие пользовательские свойства (`opf.authors` переопределяет создателя документа):

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

В дополнение к этому вы можете указать изображение для использования в качестве обложки, назвав его `opf.cover` (щелкните правой кнопкой мыши, Изображение->Опции->Имя) в ODT. Если изображение с таким именем не найдено, используется «умный» метод. Поскольку обнаружение обложки может привести к двойным обложкам в определённых выходных форматах, процесс удалит абзац (только если содержит только обложку!) из документа. Но это работает только с именованным изображением!

Чтобы отключить обнаружение обложки, вы можете установить для настраиваемого свойства `opf.nocover` (тип «Yes или No») значение «Yes» в расширенном режиме.

4.11.7 Конвертация в PDF

Первое и наиболее важное решение для преобразования в PDF — настройка размера страницы. По умолчанию в calibre используется размер «U.S. Letter». Вы можете изменить его на другой стандартный или полностью произвольный размер страницы в разделе *PDF Output* диалога преобразования. Если вы создаёте PDF для определённого устройства, вместо этого можно включить опцию использования размера страницы из *output profile*. Так, если выходной профиль — Kindle, calibre создаст PDF-файл с размером страницы, подходящим для просмотра на маленьком экране.

Верхние и нижние колонтитулы

Вы можете вставить произвольные верхние и нижние колонтитулы на каждую страницу PDF, указав шаблоны верхнего и нижнего колонтитула. Шаблоны - это просто фрагменты HTML-кода, которые отображаются в верхнем и нижнем колонтитулах. Например, чтобы отображать номера страниц по центру внизу каждой страницы зеленым цветом, используйте следующий шаблон нижнего колонтитула:

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

calibre автоматически заменит `_PAGENUM_` на номер текущей страницы. Вы можете даже размещать различное содержимое на четных и нечетных страницах, например, следующий шаблон заголовка покажет заголовок на нечетных страницах и автора на четных страницах:

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

calibre автоматически заменит `_TITLE_` и `_AUTHOR_` на название и автора конвертируемого документа. Установка `justify-content` в :code:` flex-end` приведет к выравниванию текста по правому краю.

Вы также можете отображать текст по левому и правому краям и изменять размер шрифта, как показано в этом шаблоне заголовка:

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

Это отобразит заголовок слева и автора справа, размер шрифта будет меньше основного текста.

Вы также можете использовать текущий раздел в шаблонах, как показано ниже:

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

SECTION заменяется именем текущего раздела. Эти имена взяты из метаданных Оглавление в документе (PDF Outline). Если в документе нет оглавления, он будет заменен пустым текстом. Если на одной странице PDF есть несколько разделов, будет использоваться первый раздел на странице. Аналогично, существует переменная с именем _TOP_LEVEL_SECTION_, которая может быть использована для получения имени текущего раздела верхнего уровня.

Вы даже можете использовать JavaScript внутри шаблонов верхнего и нижнего колонтитула, например, следующий шаблон приведет к тому, что номера страниц будут начинаться с 4 вместо 1:

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

Кроме того, есть ещё несколько переменных, описанных ниже для использования в верхних и нижних колонтитулах:

- _TOTAL_PAGES_ - общее количество страниц в файле PDF, полезно, например, для реализации счётчика прогресса.
- _TOP_LEVEL_SECTION_PAGES_ - общее количество страниц в текущем разделе верхнего уровня
- _TOP_LEVEL_SECTION_PAGENUM_ - номер текущей страницы в текущем разделе верхнего уровня
- _WIDTH_PIXELS_ - ширина области верхнего / колонтитула в пикселях
- _HEIGHT_PIXELS_ - высота области верхнего / колонтитула в пикселях

Примечание

При добавлении верхних и нижних колонтитулов убедитесь, что для верхних и нижних полей страницы установлены достаточно большие значения в разделе *PDF Output* диалога конвертации.

Оглавление для печати

Вы также можете вставить печатное оглавление в конец PDF-файла, в котором перечислены номера страниц для каждого раздела. Это очень полезно, если вы собираетесь распечатать PDF на бумаге. Если вы хотите использовать PDF на электронном устройстве, PDF Outline предоставляет эту функцию и создается по умолчанию.

Вы можете настроить внешний вид сгенерированного оглавления с помощью параметра «Дополнительное преобразование CSS» в разделе «Внешний вид» диалогового окна конвертации. Используемый css по умолчанию приведен ниже, просто скопируйте его и внесите все необходимые изменения.

```
.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 }
```

Пользовательские поля страницы для отдельных файлов HTML

Если вы конвертируете файл EPUB или AZW3 с несколькими отдельными файлами HTML внутри него и хотите изменить поля страницы для определённого файла HTML, вы можете добавить следующий блок стилей в файл HTML с помощью редактора calibre:

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

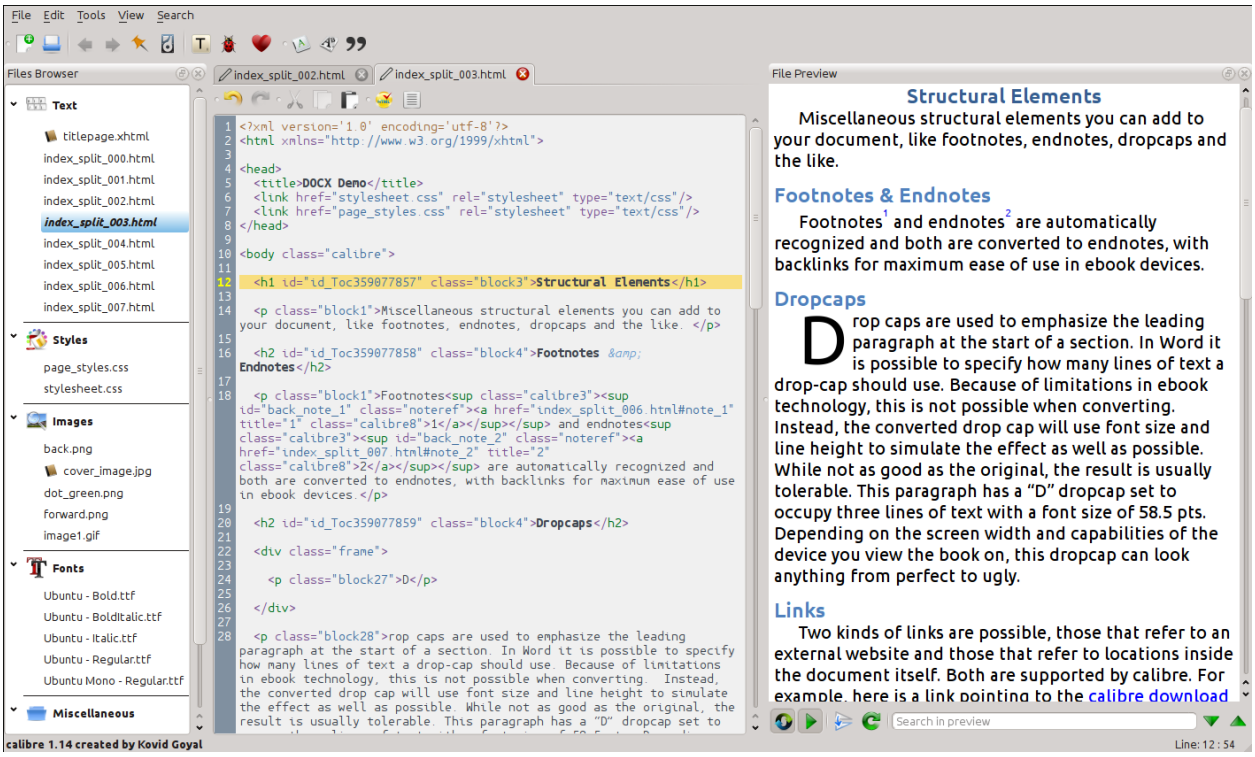
Затем в разделе вывода PDF диалогового окна конвертации включите параметр *Использовать поля страницы из преобразовываемого документа*. Теперь все страницы, сгенерированные из этого HTML-файла, будут иметь поля `10pt`.

ГЛАВА 5

Редактирование книг

В calibre есть встроенный редактор электронных книг, который может быть использован для редактирования книг в форматах EPUB, KEPUB (Kobo) и AZW3 (Kindle) . Редактор показывает HTML и CSS, которые использованы непосредственно в файле книги, с с живым просмотром всех сделанных изменений. Он также содержит различные автоматизированные средства для выполнения общей очистки и внесения исправлений.

Вы можете использовать этот редактор, щелкнув правой кнопкой на любой книге в calibre и выбрав: : *Редактировать книгу*.



Содержание

- *Основной рабочий процесс* (страница 83)
- *Браузер файлов* (страница 86)
  - *Изменение имени файлов* (страница 87)
  - *Объединить файлы* (страница 87)
  - *Изменение порядка текстовых файлов* (страница 87)
  - *Создание обложек* (страница 88)
  - *Удаление файлов* (страница 88)
  - *Экспортировать файлы* (страница 88)
  - *Добавление новых изображений/шрифтов/и т. д. или создание новых пустых файлов* (страница 88)
  - *Замена файлов* (страница 88)
  - *Эффективное связывание таблиц стилей с файлами HTML* (страница 89)
- *Поиск и замена* (страница 89)
  - *Сохранённые поиски* (страница 89)
  - *Функциональный режим* (страница 89)
  - *Поиск, игнорирующий теги HTML* (страница 90)
- *Автоматизированные инструменты* (страница 90)
  - *Редактирование оглавления* (страница 90)
  - *Проверка книги* (страница 91)
  - *Добавление обложки* (страница 92)
  - *Внедрение упоминаемых шрифтов* (страница 92)
  - *Сокращение внедрённых шрифтов* (страница 92)
  - *Улучшение пунктуации* (страница 92)
  - *Преобразование свойств CSS* (страница 92)
  - *Удаление неиспользуемых CSS правил* (страница 93)
  - *Исправление HTML* (страница 93)
  - *Улучшение файлов* (страница 93)
  - *Вставка встроенного Оглавления* (страница 93)
  - *Установка семантики* (страница 94)
  - *Фильтрация информации о стилях* (страница 94)
  - *Обновление внутренней структуры книги* (страница 94)
- *Контрольные точки* (страница 94)
- *Панель живого предпросмотра* (страница 96)
  - *Разделение HTML файлов* (страница 97)
- *Панель живого CSS* (страница 98)

- *Разные инструменты* (страница 99)
  - *Вид Оглавления* (страница 99)
  - *Проверка правописания слов в книге* (страница 99)
  - *Вставка специальных символов* (страница 101)
  - *Вид инспектор кода* (страница 102)
  - *Проверка внешних ссылок* (страница 102)
  - *Скачать внешние ресурсы* (страница 102)
  - *Упорядочить файлы в папках по типу* (страница 102)
  - *Импорт файлов других форматов электронных книг в EPUB* (страница 102)
  - *Инструмент Отчётов* (страница 113)
- *Специальные функции в редакторе кода* (страница 113)
  - *Подсветка синтаксиса* (страница 113)
  - *Контекстная помощь* (страница 114)
  - *Автозавершение* (страница 114)
  - *Фрагменты* (страница 114)

5.1 Основной рабочий процесс

Примечание

Видеотур по редактору электронных книг calibre доступен здесь[24].

Когда вы впервые откроете книгу с помощью Редактора, вам будет представлен список файлов слева. Это отдельные файлы HTML, таблицы стилей, изображения и т. д., из которых и состоит книга. Дважды щёлкните на файле, чтобы начать его редактирование. Обратите внимание, что если вы хотите сделать что-то более сложное, чем несколько небольших изменений, вам нужно знать HTML Tutorial[25] и CSS Tutorial[26].

Изменения HTML или CSS внесённые в редакторе будут показаны на панели предварительного просмотра справа. Когда Вы будете удовлетворены ими нажмите кнопку «Сохранить» или воспользуйтесь *Файл → Сохранить* для сохранения изменений в электронной книге.

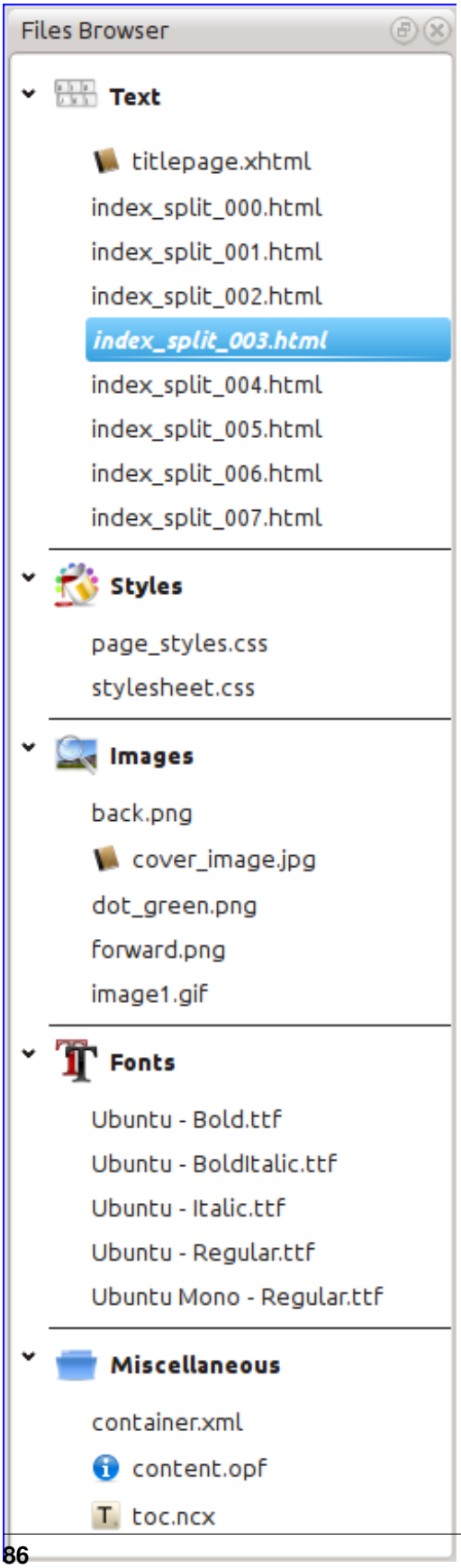
Полезная функция :guilabel: *Контрольные точки*. Прежде чем приступить к каким-либо масштабным изменениям, вы можете создать контрольную точку. Контрольная точка сохранит текущее состояние книги, а затем сделанные изменения в книге можно вернуть к состоянию при создании контрольной точки. Чтобы создать контрольную точку, используйте *Правка → Создать контрольную точку*. Контрольные точки будут автосоздаваться каждый раз, при запуске автоматизированного инструмента, вроде глобального поиска и замены. Функциональность контрольных точек - дополнение к обычному механизму отмены/возврата при редактировании отдельных файлов. Контрольные точки полезны, когда изменения распространяются на несколько файлов в книге.

Это основной рабочий процесс редактирования книг - откройте файл, внесите изменения, просмотрите и сохраните. В оставшейся части этого руководства будут обсуждаться различные инструменты и функции, которые

[24] https://calibre-ebook.com/ru/demo#tutorials
[25] https://www.w3schools.com/Html/default.asp
[26] https://www.w3schools.com/css/default.asp

позволяют вам эффективно выполнять определенные задачи.

5.2 Браузер файлов



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

Наведите указатель мыши на элемент, вы увидите его размер, а также внизу экрана полный путь к файлу внутри книги. Обратите внимание, что файлы внутри электронных книг сжаты, поэтому размер окончательной книги не является суммой размеров отдельных файлов.

Многие файлы имеют особое значение в книге. Обычно у них будет значок рядом с именем, указывающий на особое значение. Например, на рисунке слева вы видите, что файлы *cover_image.jpg* и *titlepage.xhtml* имеют значок обложки рядом с ними. Это - изображение обложки книги и титульный лист. Аналогично, файл *content.opf* имеет значок метаданных рядом с ним, указывающий на наличие метаданных книги, а файл *toc.ncx* имеет значок T, указывающий, что это Таблица контента (содержание).

Вы можете совершить множество действий с каждым файлом, щелкнув по нему правой кнопкой мыши.

5.2.1 Изменение имени файлов

Вы можете переименовать отдельный файл щелкнув по нему правой кнопкой мыши и выбрав *Переименовать*. Изменение имени файла автоматически обновляет все ссылки и сноски на них в книге. Таким образом, все что Вам надо сделать - это обозначить новое имя, calibre сделает все остальное.

Вы также можете переименовать много файлов одновременно. Это полезно, для того чтобы файлы имели простой шаблон имени. Например, можно переименовать все HTML-файлы с именами Chapter-1.html, Chapter-2.html и так далее. Выберите файлы для переименования, удерживая нажатой клавишу `Shift` или `Ctrl` и щёлкая на файлах. Затем щёлкните правой кнопкой мыши и выберите *Массовое переименование*. Введите префикс и номер, с которого вы хотите начать автоматическую нумерацию, нажмите OK, и всё готово. Диалоговое окно массового переименования также позволяет переименовывать файлы в порядке их появления в книге, а не в том порядке, в котором вы их выбрали, что полезно, например, для переименования всех изображений в порядке их появления.

Наконец, вы можете массово изменить расширение файла для всех выбранных файлов. Выберите несколько файлов, как указано выше, щелкните правой кнопкой мыши и выберите *Изменить расширение файла для выбранных файлов*.

5.2.2 Объединить файлы

Иногда вам может потребоваться объединить два HTML-файла или два CSS-файла вместе. Иногда бывает полезно иметь всё в одном файле. Однако будьте осторожны, если поместить много содержимого в один файл, это приведет к проблемам с производительностью при просмотре книги в типичной читалке электронных книг.

Чтобы объединить несколько файлов, выберите их, удерживая клавишу `Ctrl` и щёлкая по ним (убедитесь, что вы выбираете только файлы одного типа, либо все файлы HTML, либо все файлы CSS и т. д.). Затем щёлкните правой кнопкой мыши и выберите Объединить. Вот и всё, calibre объединит файлы, автоматически позаботившись о переносе всех ссылок и ссылок на объединённые файлы. Обратите внимание, что объединение файлов может иногда приводить к изменению стиля текста, поскольку отдельные файлы могли использовать разные таблицы стилей.

Вы также можете выбрать текстовые файлы, а затем перетащить текстовые файлы в другой текстовый файл, чтобы объединить перетаскиваемые текстовые файлы в целевой текстовый файл.

5.2.3 Изменение порядка текстовых файлов

Вы можете изменить порядок, в котором текстовые (HTML) файлы открываются при чтении книги, просто перетащив их в *Браузер файлов* или щёлкнув по файлу для перемещения, а затем нажав модификаторы `Ctrl`+`Shift` с помощью клавиш `Вверх`, `Вниз`, `Home` или `End`. Для технических специалистов это называется переупорядочиванием корешка книги.

Обратите внимание, что вы должны бросать пункты *между* другими пунктами, а не поверх них, это может быть неудобно, пока вы не привыкнете к этому. Перетаскивание поверх другого файла приведет к слиянию файлов.

5.2.4 Создание обложек

Обычно у электронных книг есть обложка. Это изображение отмечено в Браузере файлов значком с изображением коричневой книги рядом с названием. Если Вы хотите установить другое изображение в качестве обложки, это можно сделать, щёлкнув правой кнопкой мыши на файле и выбрав *Отметить как обложку*.

Кроме того, файлы EPUB имеют концепцию *titlepage*. Титульный лист - это HTML-файл, который выполняет роль титульного листа/обложки книги. Вы можете пометить HTML-файл как титульную страницу при редактировании EPUB, щёлкнув правой кнопкой мыши. Будьте внимательны, чтобы помеченный вами файл содержал только сопроводительную информацию. Если он содержит другое содержимое, такое как первая глава, то это содержимое будет потеряно, если пользователь когда-либо преобразует файл EPUB в calibre в другой формат. Это связано с тем, что при конвертации calibre предполагает, что отмеченная титульная страница содержит обложку и только её.

5.2.5 Удаление файлов

Вы можете удалить файлы, щёлкнув по ним правой кнопкой мыши или выбрав их и нажав клавишу Delete. Удаление файла удаляет все ссылки на файл из файла OPF, уберегая вас от этой рутинной работы. Однако ссылки в других местах не удаляются, вы можете использовать инструмент Check Book, чтобы легко найти и удалить/заменить их.

5.2.6 Экспортировать файлы

Вы можете экспортировать файл из книги в другое место на вашем компьютере. Это полезно, если вы хотите работать с файлом изолированно, с помощью специализированных инструментов. Для этого просто щелкните правой кнопкой мыши файл и выберите :guilabel: *Экспорт*.

Как только вы закончите работать с экспортированным файлом, вы можете повторно импортировать его в книгу, снова щелкнув правой кнопкой мыши на файле и выбрав *Заменить на файл…*, который позволит вам заменить файл в книге с ранее экспортированным файлом.

Вы также можете копировать файлы между несколькими экземплярами редактора. Выберите файлы, которые вы хотите скопировать, в :guilabel: *Браузер файлов*, затем щелкните правой кнопкой мыши и выберите *Копировать выбранные файлы в другой экземпляр редактора*. Затем в другом экземпляре редактора щелкните правой кнопкой мыши в *Браузер файлов* и выберите *Вставить файл из другого экземпляра редактора*.

5.2.7 Добавление новых изображений/шрифтов/и т. д. или создание новых пустых файлов

Вы можете добавить новое изображение, шрифт, таблицу стилей и т. д. со своего компьютера в книгу, нажав *Файл → Новый файл*. Это позволяет либо импортировать файл, нажав кнопку *Импортировать файл ресурса*, либо создать новый пустой файл HTML или таблицу стилей, просто введя имя файла в поле для нового файла.

Вы также можете импортировать несколько файлов в книгу одновременно, используя Файл->Импорт файлов в книгу.

5.2.8 Замена файлов

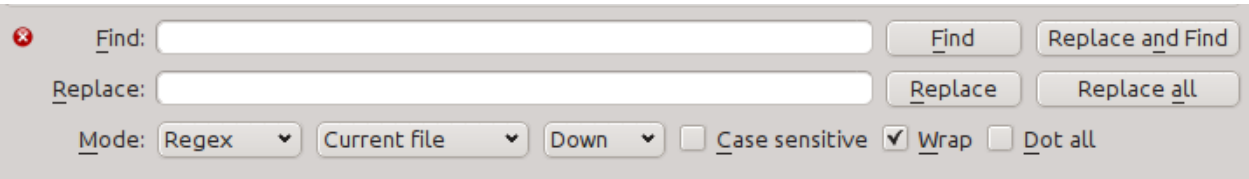
Вы можете легко заменить существующие файлы в книге, щёлкнув правой кнопкой мыши по файлу и выбрав заменить. Это автоматически обновит все ссылки и упоминания, если замещающий файл будет иметь другое имя, чем заменяемый файл.

5.2.9 Эффективное связывание таблиц стилей с файлами HTML

Для удобства вы можете выбрать несколько файлов HTML в Браузере файлов, щёлкнуть правой кнопкой мыши и выбрать ссылку на таблицы стилей, чтобы calibre автоматически вставлял <link> теги для этих таблиц стилей во все выбранные файлы HTML.

5.3 Поиск и замена

Редактир книг имеет очень мощный интерфейс поиска и замены, который позволяет вам искать и заменять текст в текущем файле, во всех файлах и даже в отмеченной области текущего файла. Вы можете искать с помощью обычного поиска или с помощью регулярных выражений. Чтобы узнать, как использовать регулярные выражения для расширенного поиска, смотрите *Всё об использования регулярных выражений в calibre* (страница 221).



Начните поиск и замените с помощью пункта меню *Поиск → Найти/заменить* (вы должны редактировать файл HTML или CSS).

Введите текст, который вы хотите найти, в поле «Найти» и его замену в поле «Заменить». Вы можете нажать соответствующие кнопки, чтобы найти следующее совпадение, заменить текущее совпадение и заменить все совпадения.

Используя раскрывающиеся списки в нижней части окна, вы можете настроить поиск по текущему файлу, всем текстовым файлам, всем файлам стилей или всем файлам. Вы также можете выбрать режим поиска: обычный (строковый) поиск или поиск по регулярному выражению.

Возможен подсчёт всех совпадений для выражения поиска *Поиск → Подсчитать все*. Счётчик работает с любыми файлами/регионами, выбранными в выпадающем списке.

Также можно перейти к определённой строке в текущем открытом редакторе с помощью *Поиск → Перейти к строке*.

Примечание

Все возможности поиска и замены реализуются регулярными выражениями. Смотрите regex tutorial.

5.3.1 Сохранённые поиски

Вы можете сохранять часто используемые выражения поиска/замены (включая выражения в функциональном режиме) и использовать их многократно. Чтобы сохранить поиск, просто щёлкните правой кнопкой мыши в поле поиска и выберите *Сохранить текущий поиск*.

Вы можете вызвать сохраненные поиски через *Поиск → Сохранённый поиск*. Вам предоставится список выражений поиска и замены, для применения. Вы сможете выбрать несколько записей в списке, удерживая нажатой клавишу Ctrl при щелчке, для запуска нескольких выражений поиска и замены выражений в одной операции.

5.3.2 Функциональный режим

Функциональный режим позволяет вам писать произвольно мощные функции Python, которые запускаются при каждом поиске/замене. Вы можете выполнять практически любые манипуляции с текстом в функциональном режиме . Для получения дополнительной информации см .:doc:*function_mode*.

5.3.3 Поиск, игнорирующий теги HTML

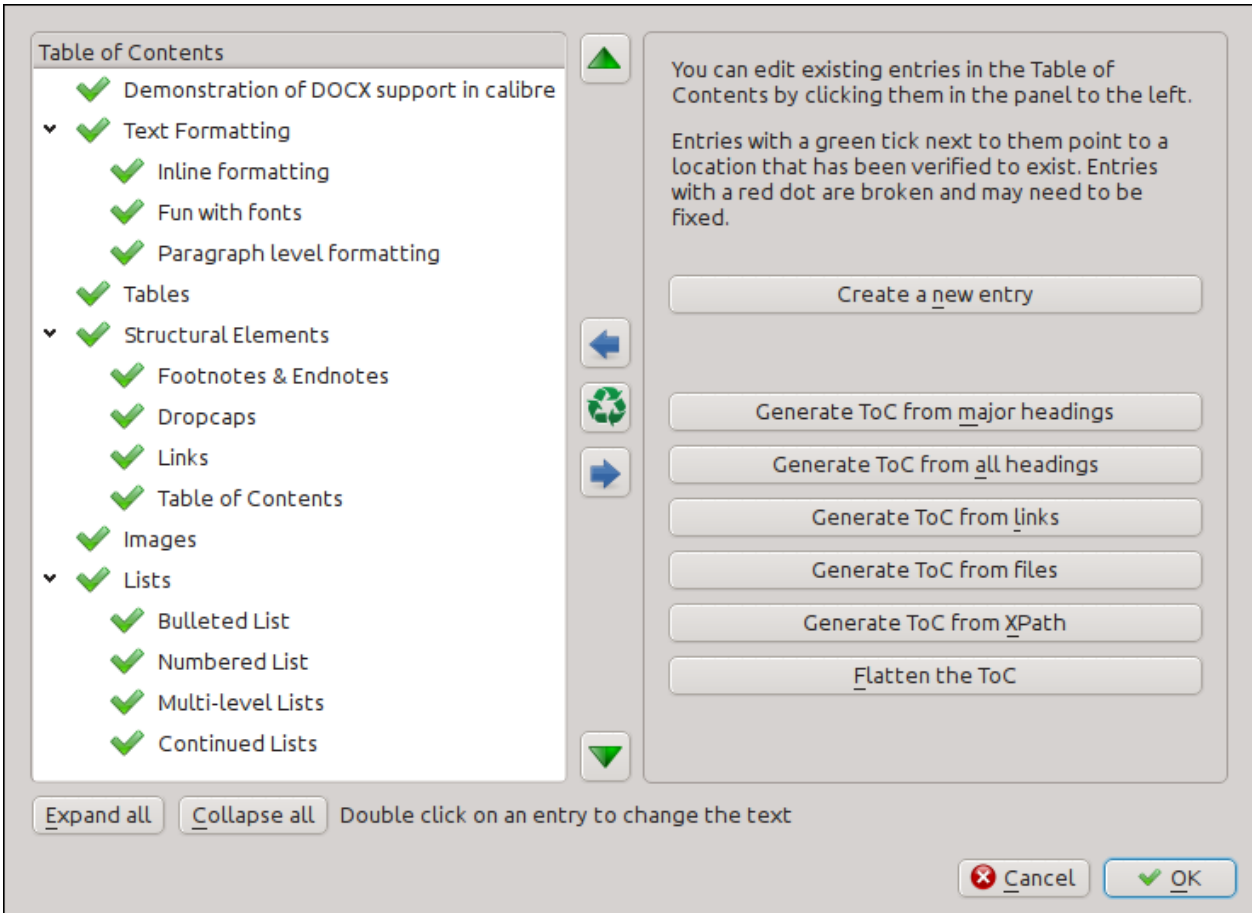
Существует также специальный инструмент для поиска текста, игнорирующий любые HTML-теги между ними. Например, если в книге есть HTML `Empahisis on a <i>word</i>`, вы можете искать `on a word`, и он будет найден, даже если в середине есть тег `<i>`. Используйте этот инструмент через пункт меню :guilabel: *Поиск->Поиск, игнорирующий HTML-разметку*.

5.4 Автоматизированные инструменты

Редактор книг имеет различные инструменты для помощи с общими задачами. Доступ к ним осуществляется через меню *Инструменты*.

5.4.1 Редактирование оглавления

Существует специальный инструмент для облегчения редактирования оглавления. Запустите его с помощью *Инструменты → Оглавление → Редактировать оглавление*.

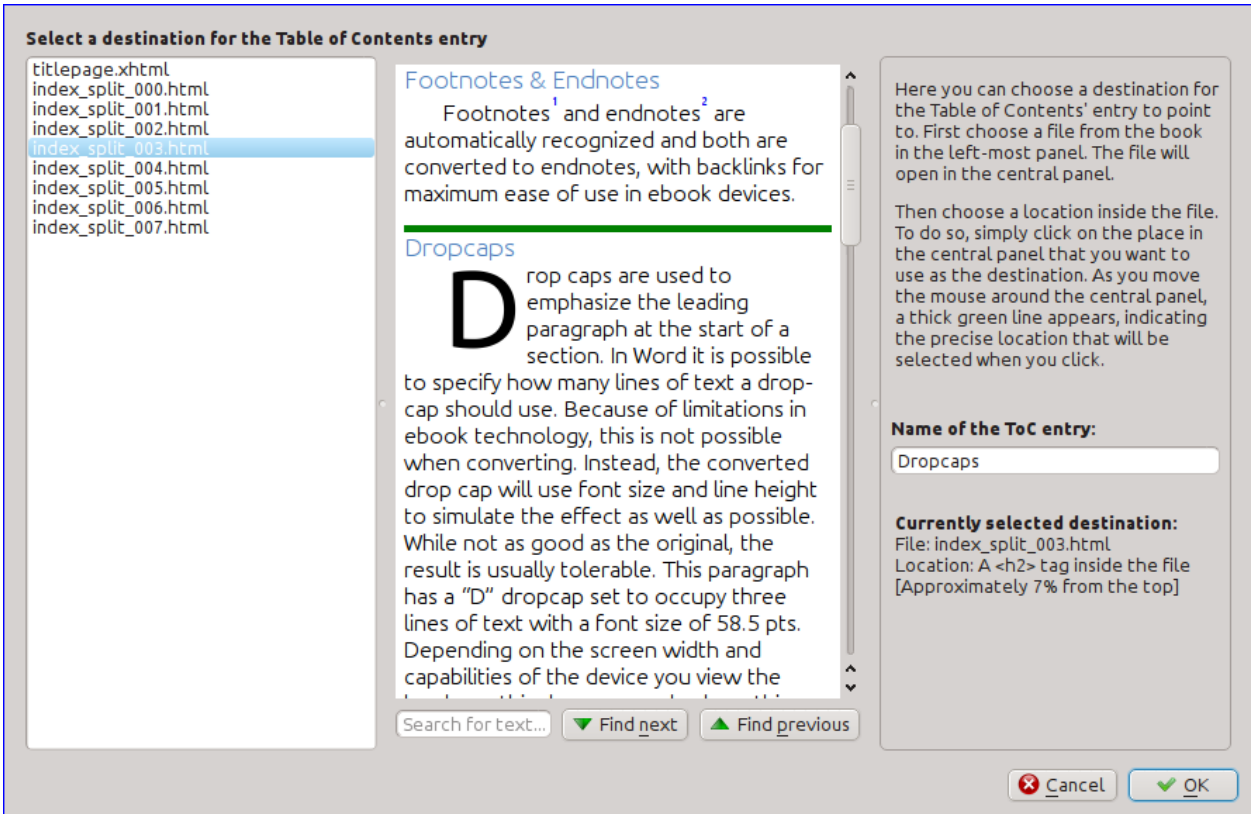


Инструмент Редактировать оглавление показывает вам текущее оглавление (если оно есть) слева. Просто дважды щелкните по любой записи, чтобы изменить её текст. Вы также можете изменить порядок записей путем перетаскивания или с помощью кнопок справа.

Для книг, которые не имеют существующего Оглавления, этот инструмент предоставляет различные варианты автоматической генерации Оглавления из текста. Вы можете создавать из заголовков в документе, из ссылок, из отдельных файлов и так далее.

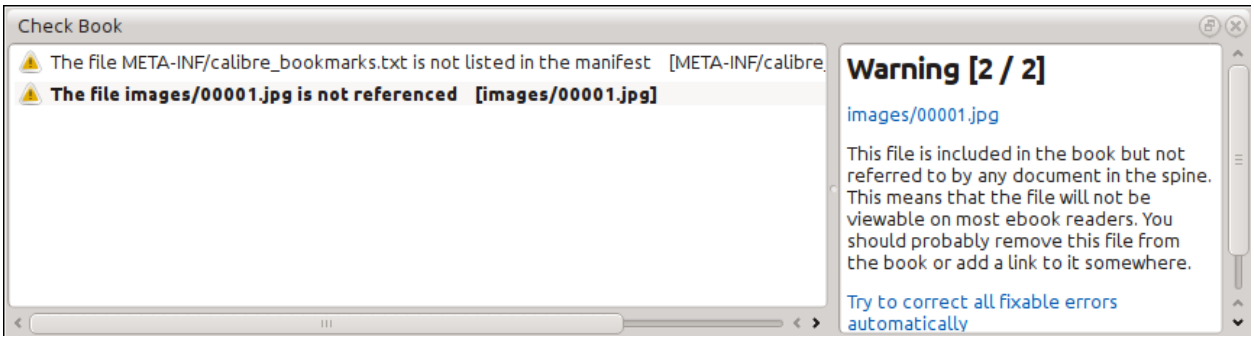
Вы можете редактировать отдельные записи, нажимая на них, а затем нажимая *Изменить местоположение, на*

которое указывает эта запись. Откроется предварительный мини-просмотр книги, просто наведите курсор мыши на панель просмотра книги и щёлкните там, куда должна указывать запись. Толстая зелёная линия покажет вам местоположение. Нажмите OK, когда вы будете довольны местоположением.



5.4.2 Проверка книги

Инструмент *Проверка книги* проверяет книгу на наличие проблем, которые могут помешать её чтению на реальных устройствах чтения. Активируйте его через *Инструменты → Проверить книгу*.



О найденных проблемах сообщается в простом и удобном списке. Нажав на любую запись в списке вы получите справку об этой ошибке, и сможете автоматически её исправить, если это возможно. Также можно дважды щёлкнув по ошибке открыть её местоположение в редакторе и исправить её вручную.

Некоторые из выполненных проверок:

- Неправильная HTML-разметка. Сообщается о любой разметке HTML, которая не анализируется как правильно сформированный XML. Исправив её, вы гарантируете, что ваша разметка работает так, как задумано во всех контекстах. calibre также может автоматически исправить эти ошибки, но иногда автоматическое

исправление может сбойнуть, поэтому используйте его с осторожностью. Как всегда, перед автоматическим исправлением создается контрольная точка, чтобы легко отменить все изменения. Автоматическое исправление работает анализируя разметку с использованием алгоритма HTML5, который очень устойчив к сбоям, а затем преобразовывает её в правильно сформированный XML.

- Неправильные или неизвестные стили CSS. Сообщается о любом недопустимом CSS или имеющем свойства, не определенные в стандарте CSS 2.1 (плюс некоторые из CSS 3). CSS проверяется во всех таблицах стилей, атрибутах встроенных стилей и <style>тегах в файлах HTML.
- Неработающие ссылки. Отчёт о ссылках, которые указывают на отсутствующие файлы в книге.
- Файлы без ссылок. Отчёт о файлах в книге, на которые не ссылаются никакие другие файлы или которые не находятся в корешке.
- Разные общие проблемы в файлах OPF, такие как дублирование элементов корешка или манифеста, неработающие идентификаторы или мета-покрывающие теги , пропущенные обязательные разделы и т. д.
- Разные проверки совместимости для известных проблем, которые могут вызвать сбой книги на читалках.

5.4.3 Добавление обложки

Вы можете легко добавить обложку в книгу через *Инструменты → Добавить обложку*. Это позволяет либо выбрать существующее изображение в книге в качестве обложки, либо импортировать новое изображение в книгу и сделать его обложкой. При редактировании файлов EPUB автоматически создается HTML-оболочка для обложки. Если в книге найдена существующая обложка, она заменяется. Инструмент также автоматически заботится о правильной маркировке файлов обложек как обложек в OPF.

5.4.4 Внедрение упоминаемых шрифтов

Доступ через *Инструменты → Внедрить упомянутые шрифты*, этот инструмент находит все шрифты, на которые есть ссылки в книге, и, если они еще не встроены, ищет их на вашем компьютере и встраивает их в книгу, если она найдена. Прежде чем делать это, убедитесь, что у вас есть необходимые авторские права для встраивания коммерчески лицензированных шрифтов.

5.4.5 Сокращение внедрённых шрифтов

Доступ к нему осуществляется через *Инструменты → Подмножество встроенных шрифтов*, этот инструмент усекает все шрифты в книге, чтобы они содержали только глифы для текста, фактически присутствующие в книге. Это обычно уменьшает размер файлов шрифтов на ~ 50%. Однако имейте в виду, что если шрифты являются подмножеством, и вы добавите новый текст, символы которого ранее не присутствовали в шрифте подмножества, шрифт не будет работать для нового текста. Так что делайте это только как последний шаг в вашем рабочем процессе.

5.4.6 Улучшение пунктуации

Конвертировать текстовые тире, многоточия, кавычки, множественные переносы и т.д. в их типографически верные эквиваленты. Помните, что алгоритм может иногда выдавать неверные результаты, особенно в случае одиночных кавычек в начале сокращений. Доступ через *Инструменты → Умная пунктуация*.

5.4.7 Преобразование свойств CSS

Создание правил для преобразования стиля книги. Например, создайте правило, чтобы преобразовать весь красный текст в зелёный или удвоить размер шрифта всего текста в книге, или сделать текст из определенного семейства шрифтов курсивом и т. д.

Создать правила просто, правила следуют формату естественного языка, который выглядит следующим образом:

- Если свойство *color* имеет значение *red change* его на *green*

- Если свойство *font-size* равно *any value multiply* значение на *2*

Доступ через *Инструменты → Трансформировать стили*.

5.4.8 Удаление неиспользуемых CSS правил

Удалить все неиспользуемые CSS правила из таблицы стилей и тегов. Некоторые книги, созданные с помощью технических шаблонов, могут иметь множество дополнительных CSS правил, не совпадающих с реальным содержанием. Это замедляет работу читалок при разборе правил. Доступ через *Инструменты → Удалить неиспользуемые CSS*.

5.4.9 Исправление HTML

Этот инструмент просто конвертирует HTML, который не может быть проанализирован как XML, в правильно сформированный XML. В электронных книгах очень часто бывает неправильно сформированный XML, поэтому этот инструмент просто автоматизирует процесс исправления такого HTML. Инструмент работает, анализируя HTML с использованием алгоритма HTML5 (алгоритм, используемый во всех современных браузерах), а затем преобразовывая результат в XML. Имейте в виду, что автоматическое исправление иногда может привести к противоречивым результатам. При желании вы можете использовать инструмент Проверить книгу, рассмотренный выше, чтобы найти и вручную исправить проблемы в HTML. Доступ через *Инструменты → Исправить HTML*.

5.4.10 Улучшение файлов

Этот инструмент используется для автоматического форматирования всех файлов HTML и CSS, чтобы они «выглядели красиво». Код автоматически выравнивается с отступами, вставляются пустые строки, при необходимости и т.д. Обратите внимание, что улучшение также автоматически исправляет поврежденный HTML/CSS. Поэтому, если вы не хотите, чтобы выполнялось автоматическое исправление, сначала воспользуйтесь инструментом Проверить книгу, чтобы исправить все проблемы, и только затем запустите улучшение. Доступ через *Инструменты → Улучшить все файлы*.

Примечание

В HTML любой текст может иметь значительный пробел через white-space директиву CSS. Следовательно, улучшение может потенциально изменить отображение HTML. Чтобы избежать этого, алгоритм beautify улучшает только теги уровня блока, которые содержат другие теги уровня блока. Так, например, текст внутри <p>тега не будет менять свой пробел. Но <body>тег, который содержит только другие <p> и теги, будет улучшен. Иногда это значит, что конкретный файл не будет затронут улучшением, поскольку у него нет подходящих тегов уровня блока. В таких случаях вы можете попробовать другие инструменты для улучшения, например: HTML Tidy[27].

5.4.11 Вставка встроенного Оглавления

Обычно в электронных книгах оглавление отделено от основного текста и доступно через специальную кнопку/меню оглавления в устройстве для чтения электронных книг. Вы также можете сделать так, чтобы calibre автоматически генерировал *inline* оглавление, которое становится частью текста книги. Оно генерируется на основе определенной в настоящее время Таблицы содержания.

Если вы используете этот инструмент несколько раз, каждый вызов приведет к замене ранее созданного встроенного оглавления. Доступ к инструменту можно получить через :guilabel: *Инструменты->Оглавление->Вставить встроенное оглавление*.

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

5.4.12 Установка семантики

Этот инструмент используется для установки *семантики* в файлах EPUB. Семантика - это просто ссылки в файле OPF, которые определяют определенные места в книге как имеющие особое значение. Вы можете использовать их для определения предисловия, посвящения, обложки, оглавления и т.д. Просто выберите тип семантической информации, которую вы хотите указать, а затем выберите место в книге, на которое должна указывать ссылка. Этот инструмент доступен через *Инструменты → Установить семантику*.

5.4.13 Фильтрация информации о стилях

Этот инструмент можно использовать для простого удаления указанных свойств стиля CSS из всей книги. Вы можете сказать ему, какие свойства вы хотите удалить, например, `` color, background-color, line-height``, и он удалит их везде, где они встречаются - таблицы стилей, теги `<style>` и инлайновые `style` атрибуты. После удаления информации о стиле отображается сводка с точным списком всех сделанных изменений. Доступ к инструменту можно получить через *Инструменты → Фильтровать информацию о стиле*.

5.4.14 Обновление внутренней структуры книги

Этот инструмент может быть использован для обновления внутренней структуры книги, по возможности. Например, он обновит книги EPUB 2 до книг EPUB 3. Доступ к инструменту можно получить через *Обновление внутренней структуры книги*.

5.5 Контрольные точки

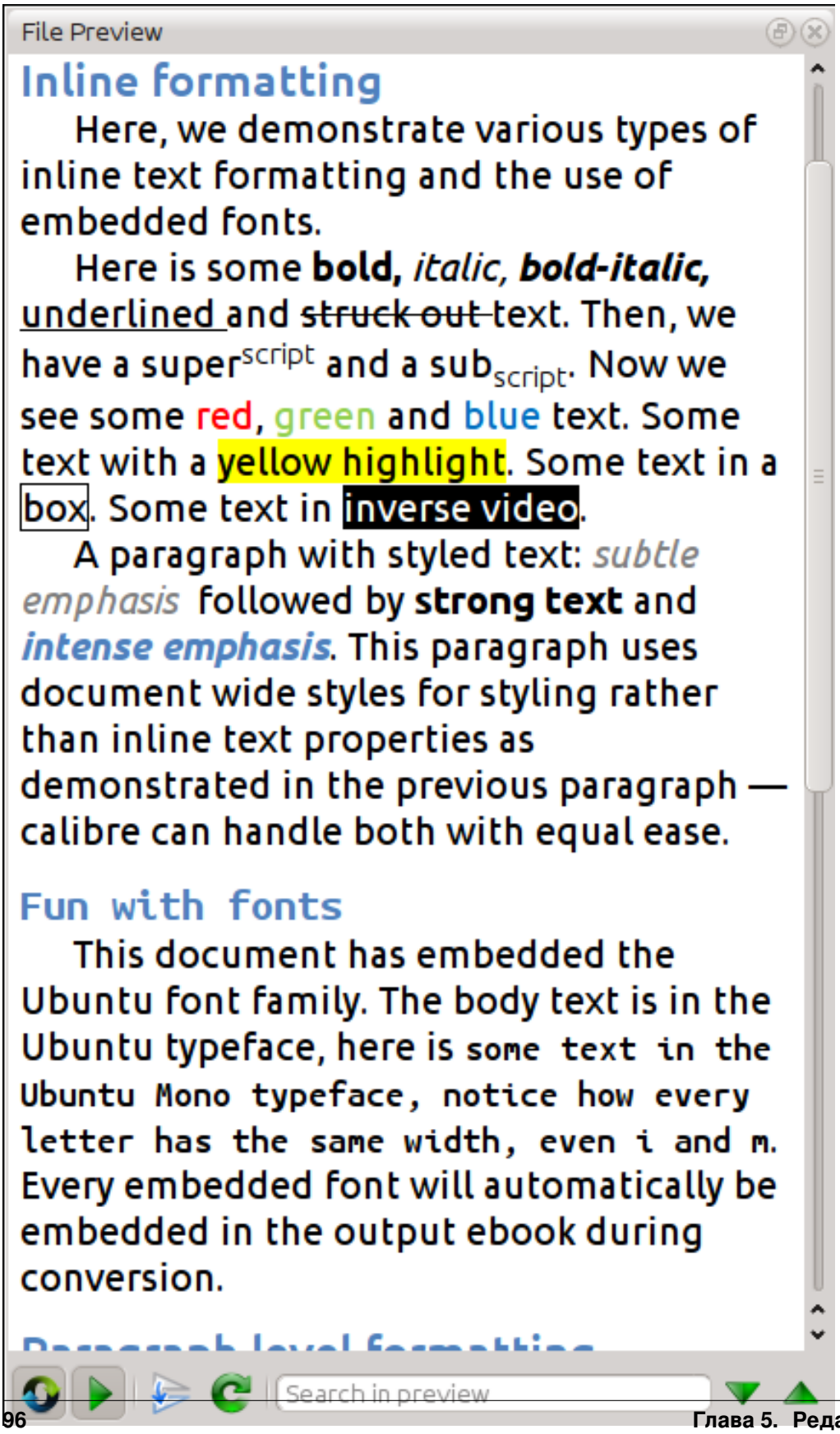
Контрольные точки - это способ пометить текущее состояние книги как «особенное». Затем вы можете продолжить вносить любые изменения в книгу и, если вам не нравятся результаты, вернитесь к состоянию контрольной точки. Контрольные точки создаются автоматически при каждом запуске любого из автоматизированных инструментов, описанных в предыдущем разделе.

Вы можете создать контрольную точку через *Правка → Создать контрольную точку*. И вернитесь к предыдущей контрольной точке с помощью *Правка → Вернуться к…*

Функциональность контрольных точек - дополнение к обычному механизму отмены/возврата при редактировании отдельных файлов. Контрольные точки особенно полезны, когда изменения распространяются на несколько файлов в книге или когда вы хотите иметь возможность отменить большую группу связанных изменений в целом.

Вы можете увидеть список доступных контрольных точек через *Вид → Контрольные точки*. Вы можете сравнить текущее состояние книги с указанной контрольной точкой, используя инструмент *Сравнение книг* (страница 123) - выбрав интересующую контрольную точку и нажав кнопку *Сравнить*. Кнопка *Вернуться к…* восстанавливает книгу до выбранной контрольной точки, отменяя все изменения с момента создания этой контрольной точки.

5.6 Панель живого предпросмотра



Просмотр файла дает обзор различных файлов внутри. Панель живого предпросмотра показывает изменения, которые вы вносите в реальном времени (с задержкой в одну или две секунды). При редактировании файлов HTML или CSS панель предварительного просмотра обновляется автоматически, чтобы отразить ваши изменения. Когда вы перемещаете курсор в редакторе, панель предварительного просмотра будет отслеживать его местоположение, показывая вам соответствующее местоположение в книге. Щелчок на панели предварительного просмотра приведет к тому, что курсор в редакторе будет расположен над элементом, который вы щелкнули. Если щелкнуть ссылку, указывающую на другой файл в книге, этот файл будет автоматически открыт в панели редактирования и предварительного просмотра.

Вы можете отключить автоматическую синхронизацию положения и предварительный просмотр изменений - с помощью кнопок под панелью предварительного просмотра. Обновление панели живого предпросмотра в реальном времени происходит только тогда, когда вы не печатаете в редакторе активно, чтобы не отвлекать и не замедлять работу в ожидании предварительного просмотра.

Панель предварительного просмотра показывает, как будет выглядеть текст при просмотре. Тем не менее, панель предварительного просмотра не является заменой для фактического тестирования вашей книги в реальном устройстве чтения. Он более и менее способен, чем настоящий читатель. Он будет лучше переносить ошибки и небрежную разметку, чем большинство устройств для чтения. Также не будут отображаться поля страницы, разрывы страниц и встроенные шрифты, которые используют псевдонимы имен шрифтов. Используйте панель предварительного просмотра, пока вы работаете над книгой, но как только вы закончите, просмотрите её в реальном устройстве чтения или программном эмуляторе.

Примечание

Панель предварительного просмотра не поддерживает встроенные шрифты, если имя шрифта внутри файла шрифта не совпадает с именем в правиле CSS @font-face. Вы можете использовать инструмент Проверить книгу для быстрого поиска и исправления таких проблемных шрифтов.

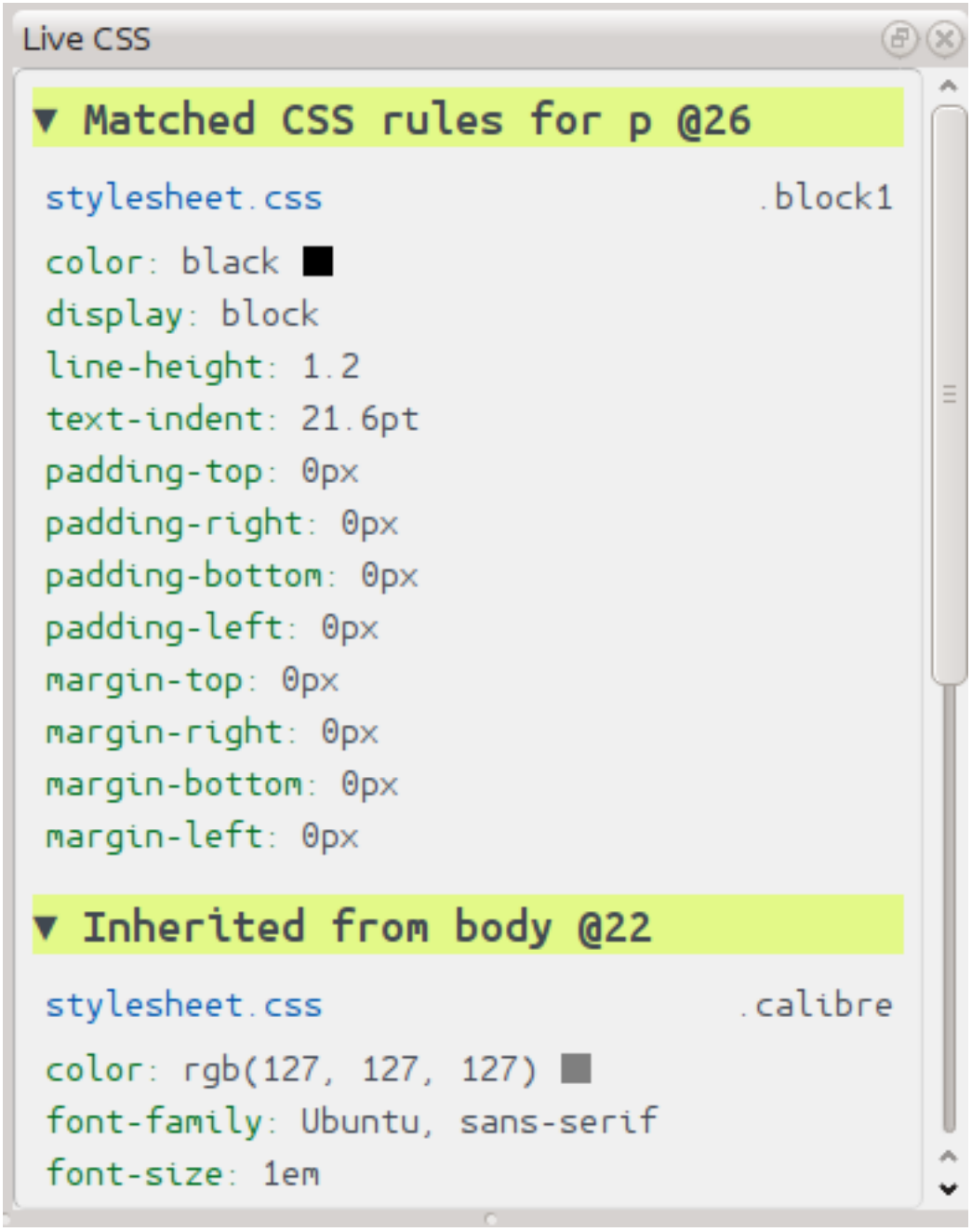
5.6.1 Разделение HTML файлов

Одним из, возможно, неочевидным, использованием панели предварительного просмотра является разделение длинных HTML-файлов. При просмотре файла, который вы хотите разделить, нажмите кнопку *Режим разделения* под панелью предварительного просмотра . Затем просто наведите указатель мыши на то место, где вы хотите разбить файл, и нажмите. Толстая зелёная линия покажет вам, где именно произойдет разделение при перемещении мыши. Как только вы нашли нужное место, просто нажмите и разделение будет выполнено.

Разделение файла автоматически обновит все ссылки и упоминания, указывающие на нижнюю половину файла, и откроет новый разделенный файл в редакторе.

Вы также можете автоматически разбить один HTML-файл на несколько, щёлкнув правой кнопкой мыши внутри файла в редакторе и выбрав *Разделить на несколько*. Это позволит вам легко разбить большой файл по всем тегам заголовков или всем тегам определённого класса и т.п.

5.7 Панель живого CSS



Панель *Live CSS* показывает вам все правила стиля, которые применяются к тегу, который вы сейчас редактируете. Отображается имя тега вместе с номером строки в редакторе, а затем список соответствующих правил стиля.

Это отличный способ быстро увидеть, какие правила стиля применимы к любому тегу. Представление также имеет интерактивные ссылки (синим цветом), которые ведут прямо к месту, где был определен стиль, на случай, если вы захотите внести какие-либо изменения в правила стиля. Отображаются правила стиля, которые применяются непосредственно к тегу, а также правила, унаследованные от родительских тегов.

Панель также показывает, каковы окончательно рассчитанные стили для тега. Свойства в списке, которые заменены правилами с более высоким приоритетом, показаны линией перечёркивания .

Вы можете включить панель Живого CSS через *Вид → Живой CSS*.

5.8 Разные инструменты

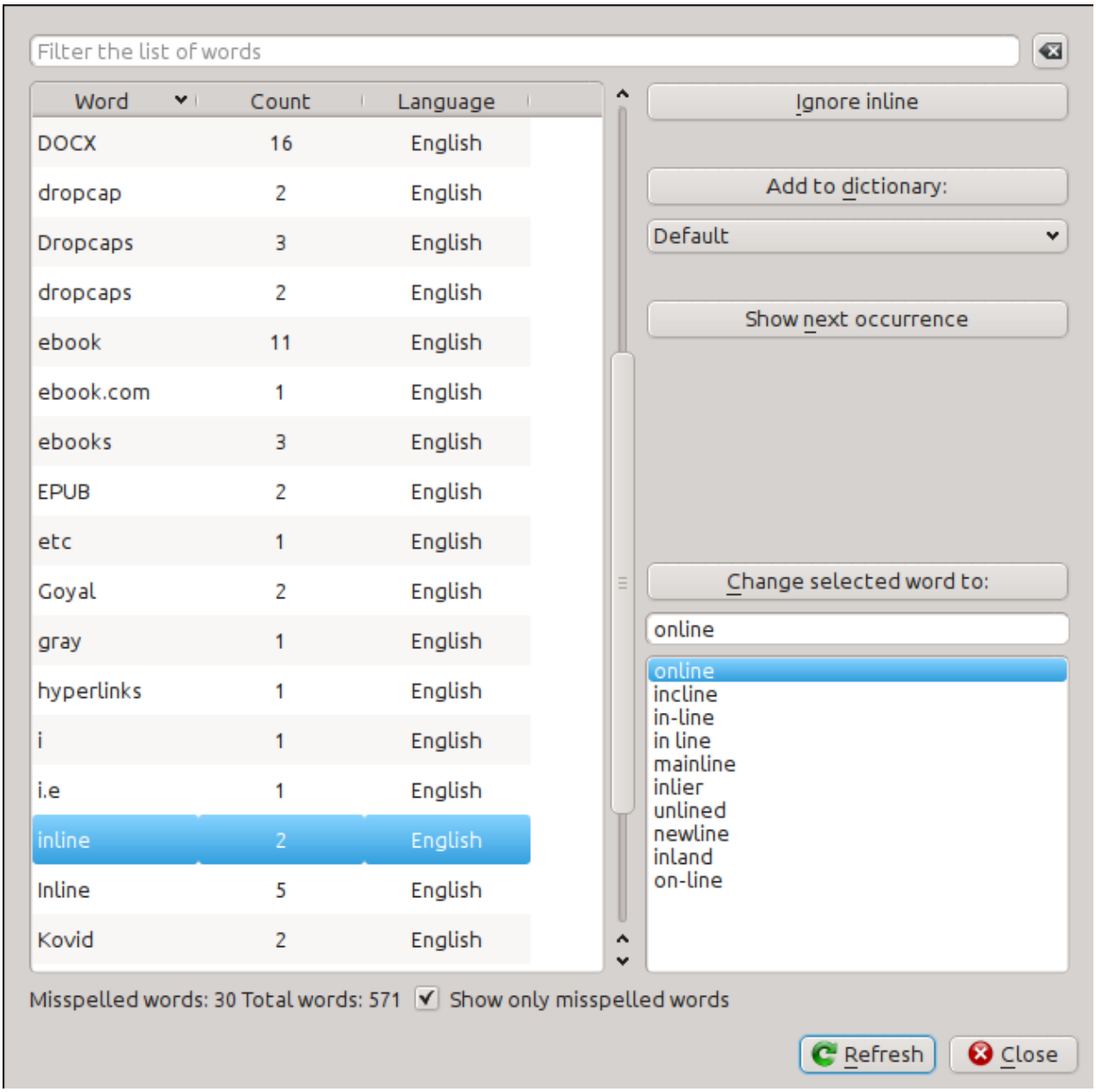
Есть еще несколько инструментов, которые могут быть полезны при редактировании книги.

5.8.1 Вид Оглавления

Представление «Оглавление» показывает текущее оглавление в книге. Двойной щелчок по любой записи открывает место, на которое указывает запись в редакторе. Вы можете щёлкнуть правой кнопкой мыши, чтобы отредактировать оглавление, обновить представление или развернуть/свернуть все элементы. Получите доступ к этому представлению через :guilabel: *Вид->Оглавление*.

5.8.2 Проверка правописания слов в книге

Вы можете запустить проверку орфографии через *Инструменты → Проверить орфографию*.



Слова показаны с количеством раз, когда они встречаются в книге, и языком, к которому относится слово. Информация о языке берется из метаданных книг и атрибутов `lang` в файлах HTML. Это позволяет программе проверки орфографии работать хорошо даже с книгами, которые содержат текст на нескольких языках. Например, в следующем фрагменте HTML слово color будет проверено с использованием американского английского, а слово colour - с использованием британского английского:

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

Примечание

Вы можете дважды щелкнуть слово, чтобы выделить следующее вхождение этого слова в редакторе. Это полезно, если вы хотите вручную отредактировать слово или посмотреть, в каком контексте оно находится.

Чтобы изменить слово, просто дважды щелкните по одному из предложенных альтернативных вариантов написания справа или введите собственное исправленное написание и нажмите кнопку *Заменить выбранное слово на*. Это заменит все вхождения слова в книгу. Вы также можете щелкнуть правой кнопкой мыши слово в главном списке слов, чтобы удобно заменить слово в меню правой кнопки мыши.

Вы можете заставить программу проверки правописания игнорировать слово для текущего сеанса, нажав кнопку *Игнорировать*. Вы также можете добавить слово в пользовательский словарь, нажав кнопку *Добавить в словарь*. Средство проверки правописания поддерживает несколько пользовательских словарей, поэтому вы можете выбрать словарь, в который хотите добавить слово.

Вы также можете настроить проверку орфографии на отображение всех слов в вашей книге, а не только неправильно написанных. Это полезно, чтобы увидеть, какие слова чаще всего встречаются в вашей книге, и выполнить простой поиск и заменить отдельные слова.

Примечание

Если вы вносите какие-либо изменения в книгу, редактируя файлы, когда инструмент проверки орфографии открыт, вы должны нажать кнопку *Обновить* в инструменте проверки орфографии. Если вы этого не сделаете и продолжите использовать инструмент проверки орфографии, вы можете потерять изменения, внесенные в редакторе.

Примечание

Чтобы исключить отдельный файл из проверки орфографии при запуске инструмента проверки орфографии, вы можете использовать кнопку *Исключить файлы* или добавить следующий комментарий сразу под открывающим тегом в файле:

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

Добавление словарей

Проверка орфографии поставляется со встроенными словарями для английского и испанского языков. Вы можете установить свои собственные словари через *Настройки → Редактор → Управлять орфографическими словарями*. Средство проверки орфографии может использовать словари из программы LibreOffice (в формате .oxt). Вы можете загрузить эти словари из репозитория «LibreOffice Extensions» <https://extensions.libreoffice.org/?Tags%5B%5D=50>`_.

5.8.3 Вставка специальных символов

Вы можете вставить символы, которые сложно набрать, используя инструмент *Правка → Вставить специальный символ*. Это показывает все символы Юникода, просто нажмите на символ, который вы хотите ввести. Если вы удерживаете нажатой клавишу `Ctrl` во время нажатия, окно закроется после вставки выбранного символа. Этот инструмент можно использовать для вставки специальных символов в основной текст или в любую другую область пользовательского интерфейса, например в инструмент поиска и замены.

Поскольку символов много, вы можете определить свои собственные *Избранные* символы, которые будут показаны первыми. Просто щёлкните правой кнопкой мыши по символу, чтобы пометить его как избранный. Вы также можете щёлкнуть правой кнопкой мыши на символе в избранном, чтобы удалить его из избранного. Наконец, вы можете изменить порядок следования символов в избранном, нажав кнопку *Переупорядочить избранное*, а затем перетащите символы в избранном.

Вы также можете напрямую вводить специальные символы с помощью клавиатуры. Для этого введите код Unicode для символа (в шестнадцатеричном формате) и затем нажмите клавишу `Alt+X`, которая преобразует ранее набранный код в соответствующий символ. Например, чтобы ввести ÿ, вы должны ввести ff, а затем Alt+X. Чтобы

ввести неразрывный пробел, нужно ввести a0, а затем `Alt+X`, для ввода горизонтального многоточия вы будете использовать 2026 и `Alt+X` и так далее.

Наконец, вы можете вводить специальные символы, используя именованные объекты HTML. Например, при вводе   ввод текста будет заменен неразрывным пробелом. Замена происходит только при вводе точки с запятой.

5.8.4 Вид инспектор кода

Это представление показывает HTML-код элемента и CSS, применяемые к нему. Вы открываете его, щёлкая правой кнопкой мыши место на панели предварительного просмотра и выбирая *Исследовать*. Это позволяет вам увидеть HTML-код для этого элемента и, что более важно, стили CSS, которые применяются к нему. Вы даже можете динамически редактировать стили и сразу же увидеть, как повлияют ваши изменения. Обратите внимание, что редактирование стилей на самом деле не вносит изменений в содержание книги, а только позволяет быстро экспериментировать. Возможность интерактивного редактирования внутри Инспектора находится в стадии разработки.

5.8.5 Проверка внешних ссылок

Вы можете использовать этот инструмент, чтобы проверить все ссылки в вашей книге, которые указывают на внешние сайты. Инструмент попытается посетить каждый веб-сайт с внешними ссылками, а в случае неудачи он сообщит обо всех неработающих ссылках в удобном для вас формате.

5.8.6 Скачать внешние ресурсы

Вы можете использовать этот инструмент для автоматической загрузки любых изображений/стилевых таблиц и т. д. в книгу, которые не связаны с книгой (т. е. у них есть URL-адреса, указывающие на местоположение в Интернете). Инструмент найдет все такие ресурсы и автоматически загрузит их, добавит их в книгу и заменит все ссылки на них для использования загруженных файлов.

5.8.7 Упорядочить файлы в папках по типу

Часто при редактировании файлов EPUB, которые вы получаете откуда-то, вы обнаружите, что файлы внутри EPUB расположены случайным образом в разных подпапках. Этот инструмент позволяет автоматически перемещать все файлы в подпапки в зависимости от их типа. Доступ к нему осуществляется через *Инструменты → Упорядочить по папкам*. Обратите внимание, что этот инструмент только изменяет порядок расположения файлов внутри EPUB, но не меняет способ их отображения в Браузере файлов.

5.8.8 Импорт файлов других форматов электронных книг в EPUB

Редактор включает в себя возможность импортировать файлы в некоторых других форматах электронных книг напрямую, как новый EPUB, без полной конвертации. Это особенно полезно для непосредственного создания файлов EPUB из ваших собственных отредактированных вручную файлов HTML. Вы можете сделать это через *Файл → Импортировать файл HTML или DOCX как новую книгу*.

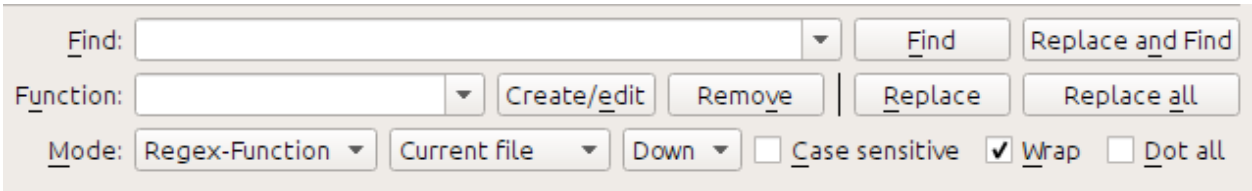
Функциональный режим для поиска и замены в редакторе

Инструмент *Search & replace* в редакторе поддерживает *функциональный режим*. В этом режиме вы можете комбинировать регулярные выражения (см. *Всё об использования регулярных выражений в calibre* (страница 221)) с произвольно мощными функциями Python для выполнения всех видов расширенной обработки текста.

В стандартном режиме *regexp* для поиска и замены вы указываете как регулярное выражение для поиска, так и шаблон, который используется для замены всех найденных совпадений. В функциональном режиме вместо

использования фиксированного шаблона вы указываете произвольную функцию на языке программирования Python[28]. Это позволяет вам делать много вещей, которые невозможны с простыми шаблонами.

Методы использования функционального режима и синтаксиса будут описаны с помощью примеров, показывающих, как создавать функции для выполнения более сложных задач.



Автоматическое исправление регистра заголовков в документе

Здесь мы будем использовать одну из встроенных функций в редакторе, чтобы автоматически изменять регистр всего текста внутри тегов заголовка на регистр заголовка:

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

Для функции просто выберите встроенную функцию *Заголовок (без тегов)*. Заголовки будут изменены следующим образом: `<h1>some titLE</h1>` на `<h1>Some Title</h1>`. Это будет работать, даже если внутри заголовочных тегов есть другие HTML-теги.

Ваша первая пользовательская функция - умные дефисы

Настоящая сила режима функций заключается в возможности создавать свои собственные функции для произвольной обработки текста. Инструмент Smarten Punctuation в редакторе оставляет индивидуальные дефисы в отдельности, поэтому вы можете использовать эту функцию, чтобы заменить их на тире.

Чтобы создать новую функцию, просто нажмите кнопку *Create/edit*, чтобы создать новую функцию и скопировать код Python снизу.

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

Каждая пользовательская функция *Search & replace* должна иметь уникальное имя и состоять из функции Python с именем replace, которая принимает все аргументы, показанные выше. На данный момент мы не будем беспокоиться о различных аргументах функции `replace()`. Просто сосредоточьтесь на аргументе `match`. Он представляет совпадение при выполнении поиска и замены. Полная документация доступна **`здесь https://docs.python.org/library/re.html#match-objects`_**. `match.group()` просто возвращает весь совпавший текст, и всё, что мы делаем, это заменяем дефисы в этом тексте длинными тире, сначала заменяя двойные дефисы, а затем одиночные дефисы.

Используйте эту функцию с регулярным выражением поиска:

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

И он заменит все дефисы на тире, но только в реальном тексте, а не в определениях тегов HTML.

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

Сила функционального режима - использование орфографического словаря для исправления неверно написанных слов

Часто электронные книги, созданные на основе отсканированных печатных книг, содержат неправильно написанные слова - слова, которые были разбиты в конце строки на печатной странице. Мы напишем простую функцию для автоматического поиска и исправления таких слов.

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

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

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

    # Search for words split by a hyphen
    text = replace_entities(match.group()[1:-1])  # Handle HTML entities like &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
```

Используйте эту функцию с тем же выражением поиска, что и раньше, а именно:

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

И это волшебным образом исправит все неправильно написанные слова в тексте книги. Основная хитрость заключается в использовании одного из полезных дополнительных аргументов для функции замены, `словарей`. Это относится к словарям, которые сам редактор использует для проверки правописания текста в книге. Эта функция ищет слова, разделенные дефисом, удаляет дефис и проверяет, распознает ли словарь составное слово; если да, то исходные слова заменяются составным словом без дефиса.

Обратите внимание, что одним из ограничений этой техники является то, что она будет работать только для одноязычных книг, потому что по умолчанию `dictionaries.recognized()` использует основной язык книги.

Авто нумерация разделов

Теперь мы увидим что-то немного другое. Предположим, ваш HTML-файл имеет много разделов, каждый из которых имеет заголовок в `<h2>` теге, который выглядит следующим образом :code: *<h2>Some text</h2>*. Вы можете создать пользовательскую функцию, которая будет автоматически нумеровать эти заголовки последовательными номерами разделов, чтобы они выглядели так `<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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

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

Используйте это с выражением поиска:

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

Поместите курсор вверху файла и нажмите *Заменить все*.

Эта функция использует еще один полезный дополнительный аргумент для `replace()`: аргумент `` number``. При выполнении *Заменить все* номер автоматически увеличивается для каждого последующего совпадения.

Другой новой функцией является использование `replace.file_order` - установка этого значения в `'spine'` означает, что если этот поиск выполняется по нескольким HTML-файлам, файлы обрабатываются в том порядке, в котором они отображаются в книга. Смотрите *Выберите порядок файлов при работе с несколькими файлами HTML* (страница 108) для деталей.

Автоматическое создание оглавления

Наконец, давайте попробуем что-нибудь более амбициозное. Предположим, что ваша книга имеет заголовки в тегах `h1` и `h2`, которые выглядят как `<h1 id="someid">Some Text</h1>`. Мы автоматически сгенерируем оглавление HTML на основе этих заголовков. Создайте пользовательскую функцию ниже:

```
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))
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        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'
```

И использовать это с выражением поиска:

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

Запустите поиск по *Все текстовые файлы* и в конце поиска появится окно Отладка вывода из вашей функции, которое будет содержать оглавление HTML, готовое для вставки в `toc.html`.

Вышеприведенная функция сильно прокомментирована, поэтому ей должно быть легко следовать. Ключевой новой возможностью является использование другого полезного дополнительного аргумента для функции `replace()` объекта `data`. Объект `data` - это Python *dict*, который сохраняется между всеми последовательными вызовами ``replace()``во время одной операции *Replace All*.

Еще одной новой возможностью является использование `call_after_last_match` - установка этого параметра в ``True`` в функции `replace()`` означает, что редактор будет вызывать `replace()` один раз после всех найденных совпадений. Для этого дополнительного вызова объектом сопоставления будет `None`.

Это была просто демонстрация мощи функционального режима: если вам действительно нужно сгенерировать оглавление из заголовков в вашей книге, вам лучше использовать специальный инструмент оглавления в :guilabel:`Инструменты->Оглавление.

API для function mode

Все функции function mode должны быть функциями Python с именем replace, со следующей сигнатурой:

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

Когда выполняется поиск/замена, для каждого найденного совпадения вызывается функция `replace()`, она должна возвращать строку замены для этого совпадения. Если никакие замены не должны быть сделаны, она должна вернуть `match.group()`, которая является исходной строкой. Различные аргументы функции `replace()` описаны ниже.

Аргумент `match`

Аргумент `match` представляет текущее найденное совпадение. Это объект Python Match[29]. Его наиболее полезным методом является `group()`, который можно использовать для получения сопоставленного текста, соответствующего отдельным группам захвата в регулярном выражении поиска.

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

Аргумент `number`

Аргумент `number` - это номер текущего совпадения. Когда вы запускаете *Replace All*, каждое последующее совпадение будет вызывать``replace()`` с возрастающим числом. Первое совпадение имеет номер 1.

Аргумент `file_name`

Это имя файла, в котором было найдено текущее совпадение. При поиске внутри помеченного текста `file_name` пусто. `file_name` находится в канонической форме, пути относительно корня книги, используя `/` в качестве разделителя пути.

Аргумент `metadata`

Это представляет метаданные текущей книги, такие как название, авторы, язык и т. д. Это объект класса *`calibre.ebooks.metadata.book.base.Metadata`* (страница 216). Полезные атрибуты включают `title`, `authors` (список авторов) и `language` (код языка).

Аргумент `dictionaries`

Это представляет собой словари, используемые для проверки орфографии текущей книги. Его наиболее полезным методом является `dictionaries.recognized(word)`, который будет возвращать `True`, если переданное слово распознается словарём для языка текущей книги.

Аргумент `data`

Это простой Python `dict`. Когда вы запускаете *Replace all*, каждое последующее совпадение будет вызывать``replace()`` с тем же `dict`, что и для данных. Таким образом, вы можете использовать его для хранения произвольных данных между вызовами `replace()` во время операции *Replace all*.

Аргумент `functions`

Аргумент `functions` предоставляет вам доступ ко всем другим пользовательским функциям. Это полезно для повторного использования кода. Вы можете определить служебные функции в одном месте и повторно использовать их во всех других ваших функциях. Например, предположим, что вы создаете имя функции `My Function` следующим образом:

```
def utility():
   # do something

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

Затем в другой функции вы можете получить доступ к функции `utility()` следующим образом:

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

Вы также можете использовать объект функций для хранения постоянных данных, которые могут быть повторно использованы другими функциями. Например, у вас может быть одна функция, которая при запуске с *Replace All* собирает некоторые данные, и другая функция, которая использует их при последующем запуске. Рассмотрим следующие две функции:

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

Отладка ваших функций

Вы можете отлаживать созданные вами функции, используя стандартную функцию `print()` из Python. Результат печати будет отображаться во всплывающем окне после завершения поиска/замены. Вы видели пример использования `print()` для вывода всего оглавления выше.

Выберите порядок файлов при работе с несколькими файлами HTML

Когда вы запускаете *Заменить все* для нескольких файлов HTML, порядок, в котором файлы обрабатываются, зависит от того, какие файлы вы открыли для редактирования. Вы можете заставить поиск обрабатывать файлы в том порядке, в котором они отображаются, установив атрибут `file_order` в вашей функции, например так:

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

replace.file_order = 'spine'
```

`file_order` принимает два значения, `spine` и `spine-reverse`, которые заставляют поиск обрабатывать несколько файлов в порядке их появления в книге, либо вперед, либо назад, соответственно.

Вызов вашей функции дополнительно после нахождения последнего совпадения

Иногда, как в приведенном выше примере с автоматическим созданием оглавления, полезно, чтобы ваша функция вызывалась дополнительно после того, как найдено последнее совпадение. Вы можете сделать это, установив атрибут `call_after_last_match` в вашей функции, например так:

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

replace.call_after_last_match = True
```

Добавление вывода из функции к выделенному тексту

При поиске и замене отмеченного текста иногда бывает полезно добавить такой текст в конец отмеченного текста. Вы можете сделать это, установив атрибут `append_final_output_to_marked` в своей функции (обратите внимание, что вам также необходимо установить `call_after_last_match`), например так:

```
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
```

Подавление диалогового окна результатов при выполнении поиска по выделенному тексту

Вы также можете отключить диалоговое окно результатов (которое может замедлить повторное применение поиска/замены во многих блоках текста), установив атрибут `suppress_result_dialog` в вашей функции, например так:

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

replace.suppress_result_dialog = True
```

Больше примеров

Больше полезных примеров от пользователей calibre можно найти на форуме редактора calibre[30].

Фрагменты

Редактор calibre поддерживает *фрагменты*. Фрагмент - это фрагмент текста, который часто используется повторно или содержит много избыточного текста. Редактор позволяет вставить фрагмент с помощью всего нескольких нажатий клавиш. Например, предположим, что вы часто вставляете теги ссылок при редактировании файлов HTML, тогда вы можете просто набрать `<a` в редакторе и нажать `Control+J`. Редактор расширит его до:

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

Кроме того, будет выбрано слово `filename` с курсором, расположенным над ним, чтобы вы могли легко ввести настоящее имя файла, используя отличную функцию редактора *Автозавершение* (страница 114). И как только вы закончите вводить имя файла, нажмите `Control+J` снова, и курсор переместится на позицию между тегами `<a>`, чтобы вы могли легко ввести текст для ссылки.

Система фрагментов в редакторе очень сложна, есть несколько встроенных фрагментов, и вы можете создать свой собственный в соответствии со своим стилем редактирования.

Следующее обсуждение встроенных фрагментов должно помочь проиллюстрировать мощь системы фрагментов.

Примечание

Вы также можете использовать фрагменты в полях ввода текста на панели *Search & replace*, однако заполнители (использующие `Control+J` для перемещения) не будут работать.

[30] https://www.mobileread.com/forums/showthread.php?t=237181

Встроенные фрагменты

Встроенные фрагменты описаны ниже. Обратите внимание, что вы можете переопределить их, создав собственные фрагменты с тем же текстом триггера.

Вставка заполнителя текста [Lorem]

Первый встроенный фрагмент и самый простой используется для вставки текста-заполнителя в документ. Текст заполнителя взят из De finibus bonorum et malorum[31] философского труда Цицерона (перевод на английский). Чтобы использовать его, просто наберите `Lorem` в HTML-файле и нажмите `Control+J`. Он будет заменен парой абзацев заполнителя.

Определение этого фрагмента очень просто, текст триггера определяется как `Lorem`, а шаблон определяется просто как текст, который нужно вставить. Вы можете легко настроить его, чтобы использовать вашу любимую форму текста заполнителя.

Вставка самозакрывающегося тега HTML [<>]

Теперь давайте рассмотрим простой пример мощной концепции *заполнителей*. Скажем, вы хотите вставить самозакрывающийся тег `<hr/>`. Просто введите `<>` и нажмите `Control+J`, редактор расширит фрагмент до:

```
<|/>
```

Здесь символ `|` представляет текущую позицию курсора. Затем вы можете набрать `hr` и нажать `Control+J` переместить курсор в конец тега. Этот фрагмент определен как:

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

Заполнители - это просто знак доллара ($), за которым следует число. Когда фрагмент раскрывается нажатием `Control+J` курсор помещается на первый заполнитель (заполнитель с наименьшим номером). Когда вы нажимаете `Control+J` снова курсор переходит к следующему заполнителю (заполнителю со следующим большим номером).

Вставить тег ссылка HTML [<a]

Все теги HTML-ссылок имеют общую структуру. У них есть атрибут `href` и некоторый текст между открывающим и закрывающим тегами. Фрагмент для более эффективного их набора познакомит нас с некоторыми функциями заполнителей. Чтобы использовать этот фрагмент, просто введите `<a` и нажмите `Control+J`. Редактор расширит это до:

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

Кроме того, будет выбрано слово `filename` с курсором, расположенным над ним, чтобы вы могли легко ввести настоящее имя файла, используя отличную функцию редактора *Автозавершение* (страница 114). И как только вы закончите вводить имя файла, нажмите `Control+J` снова, и курсор переместится на позицию между тегами `<a>`, чтобы вы могли легко ввести текст для ссылки. После того, как вы закончите ввод текста, нажмите `Control+J` снова, чтобы перейти к точке после закрывающего тега. Этот фрагмент определен как

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

Здесь есть пара новых функций. Сначала заполнитель `$1` стал более сложным. Теперь он включает в себя некоторый *текст по умолчанию* (слово `filename`). Если заполнитель содержит текст по умолчанию, то текст по

[31] https://en.wikipedia.org/wiki/De_finibus_bonorum_et_malorum

умолчанию заменяется заполнителем при раскрытии фрагмента. Также при переходе к заполнителю с текстом по умолчанию с помощью `Control+J` выбирается текст по умолчанию. Таким образом, вы можете использовать текст по умолчанию, чтобы напоминать вам о необходимости заполнения важных частей шаблона. Вы можете указать текст по умолчанию для заполнителя, используя синтаксис: `${<number>:текст по умолчанию}`.

Другая новая особенность заключается в том, что второй заполнитель имеет звездочку после него (`${2*}`). Это означает, что любой текст, который был выбран перед расширением шаблона, заменяется заполнителем. Чтобы увидеть это в действии, выделите текст в редакторе, нажмите `Control+J`, введите `<a` и нажмите `Control+J` снова шаблон будет расширен до:

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

Вставить тег изображение HTML [<i]

Это очень похоже на вставку HTML-ссылки, как мы видели выше. Он позволяет быстро вводить тег `<img src="filename" alt="description" />` и переходить между атрибутами `src` и `alt`:

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

Вставить произвольный тег HTML [<<]

Это позволяет вставлять произвольный полный тег HTML (или переносить ранее выделенный текст в теге). Чтобы использовать его, просто введите << и нажмите `Control+J`. Редактор расширит его до:

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

Введите имя тега, например: `span` и нажмите `Control+J`, что приведет к:

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

Вы заметите, что закрывающий тег был автоматически заполнен `span`. Это достигается с помощью еще одной функции заполнителей, *зеркальное отображение*. Зеркалирование просто означает, что если вы указали образец заполнителя более одного раза в шаблоне, вторая и все последующие позиции будут автоматически заполнены тем, что вы вводите в первой позиции, когда вы нажимаете `Control+J`. Определение этого фрагмента:

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

Как видите, первый заполнитель ($1) был указан дважды, второй раз в закрывающем теге, который просто скопирует всё, что вы введете в открывающий тег.

Вставка произвольного тега HTML с атрибутом класса [<c]

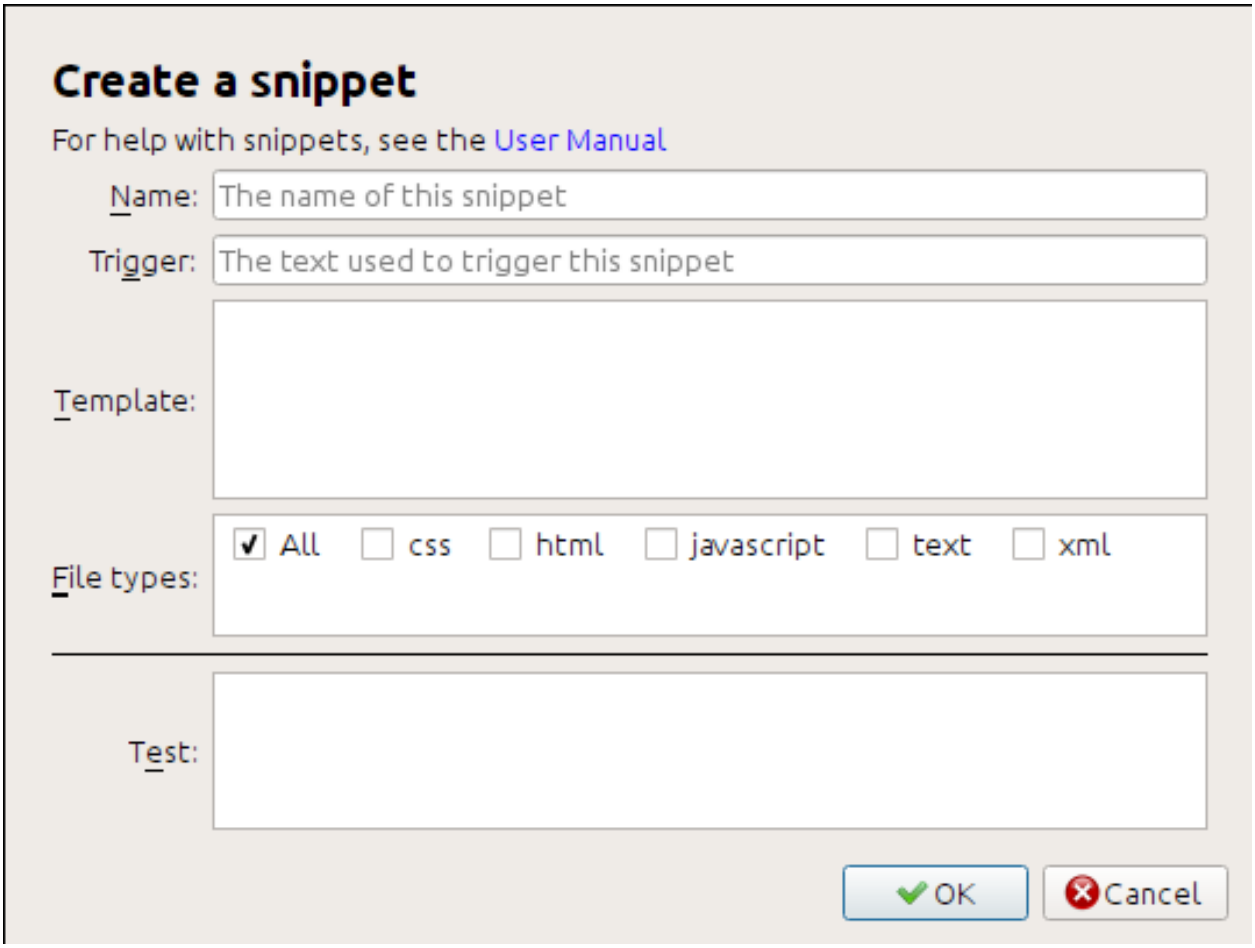
Это очень похоже на пример вставки произвольного тега выше, за исключением того, что предполагается, что вы хотите указать класс для тега:

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

Это позволит вам сначала ввести имя тега, нажать `Control+J`, ввести имя класса, нажать `Control+J`, ввести содержимое тега и нажав `Control+J` в последний раз, выпрыгнуть из тега. Закрывающий тег будет заполнен автоматически.

Создание собственных фрагментов

Фрагменты действительно блестящая вещь, потому что вы можете создать свой собственный, для соответствия своему стилю редактирования. Чтобы создать свои собственные фрагменты, перейдите в *Edit* → *Preferences* → *Settings Editor* → *Manage snippets* в редакторе. Появится простой в использовании диалог, который поможет вам создать свои собственные фрагменты. Просто нажмите кнопку *Добавить фрагмент*, и вы увидите диалоговое окно, которое выглядит следующим образом:



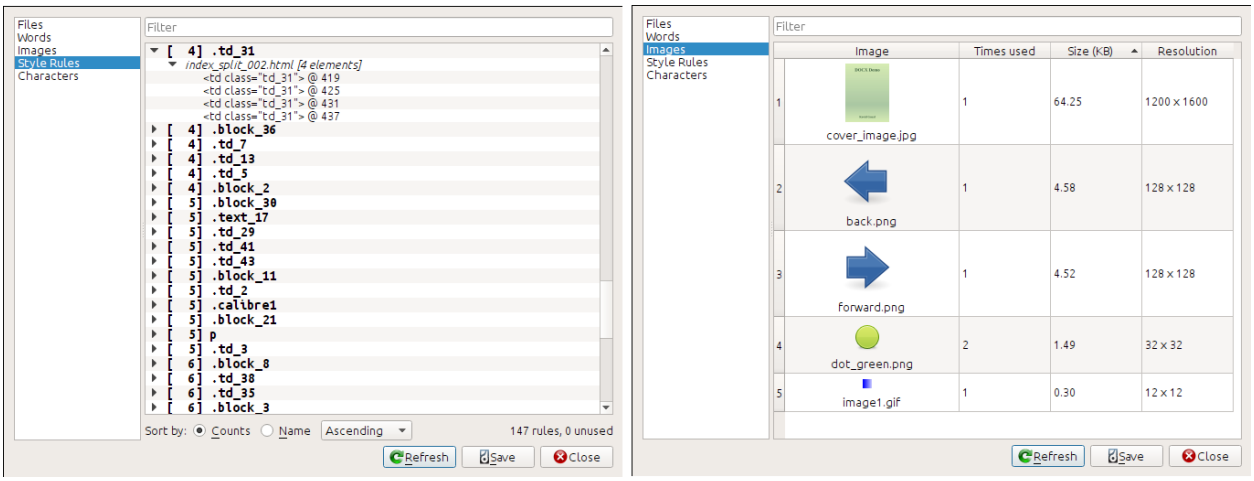
Сначала дайте вашему фрагменту имя, что-то описательное, чтобы помочь идентифицировать фрагмент в будущем. Затем укажите *триггер*. Триггер - это просто текст, который вы должны напечатать в редакторе, прежде чем нажимать `Control+J` чтобы расширить фрагмент.

Затем укажите шаблон фрагмента. Начните с одного из приведенных выше примеров и измените его в соответствии со своими потребностями. Наконец, укажите, для каких типов файлов фрагмент будет активен. Таким образом, у вас может быть несколько фрагментов с одним и тем же текстом триггера, которые по-разному работают в разных типах файлов.

Следующим шагом является проверка вашего вновь созданного фрагмента. Используйте поле *Test* внизу. Введите текст триггера и нажмите `Control+J` чтобы расширить фрагмент и перейти между заполнителями.

5.8.9 Инструмент Отчётов

Редактор включает в себя замечательный инструмент *Отчёты* (через *Инструменты → Отчёты*), который показывает сводку файлов, изображений, ссылок, слов, символов и стилей, используемых в книге. Каждая строка в отчете имеет горячую ссылку. Двойной щелчок по строке переходит к месту в книге, где этот элемент используется или определён (в зависимости от ситуации). Например, в представлении *Ссылки* можно дважды щелкнуть записи в столбце *Источник*, чтобы перейти туда, где определена ссылка, и записи в столбце *Цель*, чтобы перейти туда, где цель ссылки.



5.9 Специальные функции в редакторе кода

HTML-редактор calibre очень мощный. Он имеет много функций, которые облегчают редактирование HTML (и CSS).

5.9.1 Подсветка синтаксиса

Редактор HTML имеет очень сложную подсветку синтаксиса. Особенности включают в себя:

- Текст внутри тегов, выделенных жирным шрифтом, курсивом и заголовком, выделен жирным шрифтом/курсивом
- Когда вы перемещаете курсор по HTML-коду, подсвечиваются соответствующие теги HTML, и вы можете перейти к открывающему или закрывающему тегу с помощью сочетаний клавиш `Ctrl+{` и `Ctrl+}`. Точно так же вы можете выбрать содержимое тега с помощью `Ctrl+Alt+T` или `Ctrl+Shift+T`.
- Неправильный HTML выделяется красным подчёркиванием
- Орфографические ошибки в тексте внутри тегов HTML и атрибутов, таких как заголовок, выделены. Проверка орфографии выполняется с учетом языка на основе значения атрибута `lang` текущего тега и общего языка книги.
- CSS, встроенный в `<style>` теги, подсвечивается
- Подсвечиваются специальные символы, которые бывает трудно отличить, такие как неразрывные пробелы, различные виды дефисов и т. д.
- Ссылки на другие файлы в тегах `<a>`, `<img>` и `<link>` имеют выделенные имена файлов. Если имя файла, на которое они указывают, не существует, оно помечается красным подчеркиванием.

5.9.2 Контекстная помощь

Вы можете щёлкнуть правой кнопкой мыши по имени тега HTML или имени свойства CSS, чтобы получить справку по этому тегу или свойству.

Вы также можете удерживать нажатой клавишу `Ctrl` и щёлкнуть по любому имени файла внутри тега ссылки, чтобы автоматически открыть этот файл в редакторе. Аналогично `Ctrl` щёлкнув по имени класса, вы перейдёте к первому правилу стиля, которое соответствует тегу и классу.

Щелчок правой кнопкой мыши по имени класса в HTML-файле позволит вам переименовать класс, изменив все вхождения класса во всей книге и во всех её таблицах стилей.

5.9.3 Автозавершение

При редактировании электронной книги одной из самых утомительных задач является создание ссылок на другие файлы внутри книги, а также на таблицы стилей CSS или изображения. Вы должны выяснить правильное имя файла и относительный путь к файлу. Редактор имеет автозавершение, чтобы сделать это проще.

При вводе имени файла редактор автоматически выдаёт подсказки. Используйте клавишу `Tab`, для выбора правильного имени файла. Редактор также предлагает подсказки для ссылок, указывающих на привязку внутри другого HTML-файла. После ввода символа `#` будет выдан список всех якорей в целевом файле с небольшим фрагментом текста, помогающим выбрать правильный якорь.

Обратите внимание, что в отличие от большинства других систем завершения, система завершения редактора использует сопоставление подпоследовательностей. Это означает, что вы можете ввести всего две или три буквы из любого места в имени файла, чтобы завершить имя файла. Например, скажем, вы хотите имя файла `../images/arrow1.png`, вы можете просто набрать `ia1` и нажать Tab, чтобы завершить имя файла. При поиске совпадений система завершения определяет приоритетность букв, которые находятся в начале слова или сразу после разделителя пути. Как только вы привыкнете к этой системе, вы обнаружите, что она экономит ваше время и силы.

5.9.4 Фрагменты

Редактор calibre поддерживает *сниппеты*. Сниппет - это фрагмент текста, который часто используется повторно или содержит много избыточного текста. Редактор позволяет вставить сниппет с помощью всего нескольких нажатий клавиш. Сниппеты очень мощные, со многими функциями, такими как заполнители, между которыми вы можете перемещаться, автоматическое зеркалирование повторяющегося текста и так далее. Для получения дополнительной информации смотрите *Фрагменты* (страница 109).

ГЛАВА 6

Контент-сервер calibre

calibre *Контент-сервер* позволяет вам получать доступ к своим библиотекам calibre и читать книги прямо в браузере на мобильном телефоне или планшете. В результате вам не нужно устанавливать на телефон какие-либо специальные приложения для чтения и управления книгами. Просто используйте браузер. Сервер загружает и сохраняет книгу, которую вы читаете, в автономном кэше, чтобы вы могли читать ее, даже если нет подключения к Интернету.

Содержание

- *Доступ к контент-серверу с других устройств* (страница 116)
  - *Доступ к серверу с устройств в вашей домашней сети* (страница 116)
  - *Доступ к серверу из любой точки интернета* (страница 117)
- *Интерфейс сервера* (страница 118)
  - *Список книг* (страница 118)
  - *Просмотрщик книг* (страница 118)
- *Поддержка браузеров* (страница 118)
- *Включение режима автономной работы* (страница 119)
- *Управление пользовательскими аккаунтами с помощью командной строки* (страница 119)
- *Интеграция контент-сервера calibre с другими серверами* (страница 119)
  - *Использование полностью виртуального хоста* (страница 119)
  - *Использование префикса URL-адреса* (страница 120)
- *Создание сервиса для сервера calibre в современной системе Linux* (страница 121)

Чтобы запустить сервер, нажмите кнопку *Соединиться/поделиться* и выберите :guilabel:` Запустить контент-сервер`. Вы можете получить сообщение от брандмауэра вашего компьютера или антивирусной программы с

вопросом, можно ли разрешить доступ к `calibre.exe`. Нажмите кнопку «Разрешить» или «ОК». Затем откройте браузер (предпочтительно Chrome или Firefox) на своем компьютере и введите следующий адрес:

http://127.0.0.1:8080

Это откроет страницу в браузере, показывающую ваши библиотеки calibre, нажмите на любую и просмотрите книги в ней. Нажмите на книгу, и вы увидите все метаданные о книге, а также кнопки *Читать* и *Скачать*. Нажмите кнопку *Читать*, чтобы начать чтение книги.

Примечание

Адрес `http: //127.0.0.1: 8080` будет работать только на компьютере, на котором запущен calibre. Для доступа к серверу с других компьютеров/телефонов/планшетов и т. д. вам нужно будет проделать чуть больше работы, как описано в следующем разделе

6.1 Доступ к контент-серверу с других устройств

Существует два типа удаленного доступа к устройству, которые вам обычно необходимы. Первый, более простой вид - изнутри вашей домашней сети. Если вы используете calibre на компьютере в вашей домашней сети, и вы также подключили другие устройства к той же домашней сети, вы сможете легко получить доступ к серверу на этих устройствах.

6.1.1 Доступ к серверу с устройств в вашей домашней сети

После запуска сервера calibre, как описано выше, снова нажмите кнопку *Соединиться/поделиться*. Вместо действия *Запустить контент-сервер* вы должны увидеть действие *Остановить контент-сервер*. Справа от этого действия будет указан IP-адрес и номер порта. Они выглядят как набор чисел, разделенных точками. Например:

```
Stop Content server [192.168.1.5, port 8080]
```

Эти цифры говорят вам, какой адрес использовать для подключения к серверу на ваших устройствах. Следуя приведенному выше примеру, адрес становится:

```
http://192.168.1.5:8080
```

Первая часть адреса всегда `http://`, следующая часть - это IP-адрес, который представляет собой числа перед запятой, и, наконец, у нас есть номер порта, который необходимо добавить к IP-адресу с двоеточием (`:`). Если вам повезет, это все, что вам нужно, и вы будете смотреть на библиотеки calibre на вашем устройстве. Если нет, то читайте дальше.

Устранение неполадок подключения к домашней сети

Если вы не можете получить доступ к серверу с вашего устройства, попробуйте выполнить следующие действия:

1. Убедитесь, что сервер работает, открыв адрес `http: //127.0.0.1: 8080` в браузере, запущенном на том же компьютере, что и сервер.
2. Убедитесь, что ваш брандмауэр/антивирус разрешает подключения к вашему компьютеру через порт `8080` и к программе calibre. Самый простой способ устранить брандмауэр/антивирус как источник проблем - временно отключить их оба, а затем попробовать подключиться. Прежде чем отключать брандмауэр, сначала необходимо отключиться от Интернета, чтобы обеспечить безопасность компьютера.
3. Убедитесь, что ваше устройство и компьютер находятся в одной сети. Это означает, что они оба должны быть подключены к одному беспроводному маршрутизатору. В частности, ни один из них не должен использовать сотовую связь или интернет-провайдер с прямым подключением.

4. Если у вас нестандартные настройки сети, возможно, IP-адрес, указанный в меню *Соединиться/поделиться*, неверен. В таком случае вам придется самостоятельно определить, какой правильный IP-адрес использовать. К сожалению, учитывая бесконечное разнообразие возможных сетевых конфигураций, невозможно дать вам план действий для этого.
5. Если вы настроили имя пользователя и пароль, попробуйте сначала без них, чтобы увидеть, вызывает ли это проблемы. Некоторые устройства e-ink имеют браузеры, которые не поддерживают аутентификацию. Иногда это можно обойти, включив имя пользователя и пароль в URL, например: `http://username:password@192.168.1.2: 8080`.
6. Если вы застряли, вы всегда можете обратиться за помощью на форумы пользователей calibre[32].

6.1.2 Доступ к серверу из любой точки интернета

Предупреждение

Перед этим вы должны включить защиту на сервере с указанием имени пользователя/пароля, иначе любой человек в мире сможет получить доступ к вашим книгам. Перейдите по адресу *Настройки → Поделиться → Поделиться по сети* и включите опцию *Требуется имя пользователя и пароль для доступа к контент-серверу*.

Хотя конкретные сведения о настройке доступа в Интернет различаются в зависимости от конфигурации сети и типа используемого вами компьютера, основная схема выглядит следующим образом.

1. Узнайте внешний IP-адрес компьютера, на котором вы собираетесь запустить сервер. Вы можете сделать это, посетив сайт What is my IP address[33] в браузере, работающем на компьютере.
2. Если компьютер находится за маршрутизатором, включите переадресацию портов на маршрутизаторе, чтобы перенаправить порт `8080` (или любой другой порт, на котором вы хотите запустить контент-сервер calibre) на компьютер.
3. Убедитесь, что доступ к серверу calibre разрешен через любые брандмауэры/антивирусные программы на вашем компьютере.
4. Теперь вы сможете получить доступ к серверу на любом подключенном к Интернету устройстве, используя IP-адрес, который вы нашли в первом шаге. Например, если IP-адрес, который вы нашли, был `123.123.123.123`, а порт, который вы используете для сервера calibre - `8080`, адрес, который будет использоваться на вашем устройстве, станет: `http://123.123 .123.123:8080`.
5. При желании можно использовать службу типа no-ip <https://www.noip.com/free>`_, чтобы настроить легко запоминаемый адрес для использования вместо IP-адреса, найденного на первом шаге.

Примечание

Для максимальной безопасности вы также должны включить HTTPS на сервере содержимого. Вы можете сделать это непосредственно на сервере, указав путь к сертификату HTTPS для использования в дополнительных параметрах конфигурации для сервера, или вы можете настроить обратный прокси-сервер, как описано ниже, для использования существующей настройки HTTPS.

[32] https://www.mobileread.com/forums/forumdisplay.php?f=166
[33] https://www.whatismyip.com/

6.2 Интерфейс сервера

Интерфейс сервера - это упрощенная версия основного интерфейса calibre, оптимизированная для использования с сенсорными экранами. На главном экране отображаются книги, которые вы сейчас читаете, а также можно выбрать библиотеку calibre, которую вы хотите просмотреть. calibre сервер предоставляет вам доступ ко всем вашим библиотекам, а не только к одной, как раньше.

6.2.1 Список книг

Список книг сервера представляет собой простую сетку обложек. Нажмите на обложку, чтобы увидеть подробные метаданные для книги или прочитать книгу. Если вы предпочитаете более подробный список, вы можете изменить вид по умолчанию, нажав на три вертикальные точки в верхнем правом углу.

Сортировка и поиск в списке книг должны быть знакомы пользователям calibre. Доступ к ним можно получить, щелкнув их значки в правом верхнем углу. Они оба работают точно так же, как в основной программе calibre. Страница поиска даже позволяет создавать поисковые запросы, нажимая на авторов/ теги и т. д., так же, как и в Браузере тегов в основной программе.

Особо любимая особенность основной программы *Виртуальные библиотеки* также присутствует в интерфейсе сервера. Нажмите на три вертикальные точки в правом верхнем углу, чтобы выбрать виртуальную библиотеку.

6.2.2 Просмотрщик книг

Вы можете прочитать любую книгу в библиотеке calibre, просто нажав на неё, а затем нажав кнопку *Читать*. Просмотрщик книг очень прост в управлении. Вы можете нажать и провести пальцем, чтобы перевернуть страницы. Смахивание вверх/вниз пропускает главы. При нажатии на верхнюю четверть экрана вы получите подробные элементы управления и настройки просмотра.

Если вы оставите сервер содержимого включенным, вы даже сможете открыть одну и ту же книгу на нескольких устройствах, и она запомнит вашу последнюю прочитанную позицию. Если это не так, вы можете вызвать синхронизацию, нажав в верхней четверти и выбрав :guilabel: *Sync*.

6.3 Поддержка браузеров

В новом сервере calibre широко используются расширенные функции HTML 5 и CSS 3. Таким образом, для использования требуется современный браузер. Он был протестирован в Android Chrome и iOS Safari, а также в Chrome и Firefox на рабочем столе

Сервер использует функциональные возможности, которые либо уже стандартизированы, либо находятся в соответствии со стандартами. Если он в настоящее время не работает с вашим любимым браузером, но, вероятно, будет работать, когда этот браузер догонит его по возможностям.

Если вы используете особенно старый или ограниченный браузер или вам не нравится запускать JavaScript, вы можете использовать представление *mobile*, просто добавив `/mobile` к адресу сервера.

Примечание

В iOS Apple позволяет использовать только один движок браузера, поэтому Firefox, Chrome и Safari на самом деле являются одним и тем же браузером. Для нового серверного интерфейса требуется iOS 10.3.2 или новее. На Android сервер был протестирован с Chrome версии 58 и новее.

6.4 Включение режима автономной работы

Производители браузеров убеждают людей использовать SSL отключая расширенные функции в браузерах для простых HTTP-соединений. Одна из жертв этого - ApplicationCache для офлайн-поддержки в calibre. В результате, в настоящее время автономный режим работает только до тех пор, пока вы держите вкладку браузера открытой. Кроме того, в Firefox на Android вам нужно будет ввести `about: config`, создать настройку `browser.tabs.useCache` и переключить её на `true`.

6.5 Управление пользовательскими аккаунтами с помощью командной строки

У calibre неплохой раздел в *Preferences*, позволяющий управлять учётными записями пользователей на сервере. Однако, если вы хотите запустить автономный сервер и не можете запустить основную программу calibre на том же компьютере/учетной записи пользователя, то можете управлять пользователями, используя командную строку.

Вы можете управлять учетными записями пользователей, используя параметр `--manage-users` в автономной программе `calibre-server`. Предположим, вы хотите сохранить базу данных пользователей в папке `/srv/calibre`, затем вы создаете её, запустив:

```
calibre-server --userdb /srv/calibre/users.sqlite --manage-users
```

Просто следуйте инструкциям, чтобы создать учетные записи пользователей, установить их разрешения и т.д. Когда вы закончите, вы можете запустить сервер как:

```
calibre-server --userdb /srv/calibre/users.sqlite --enable-auth
```

Он будет использовать учетные записи пользователей, которые вы создали на предыдущем шаге.

6.6 Интеграция контент-сервера calibre с другими серверами

Здесь мы покажем вам, как интегрировать сервер контента calibre в другой сервер. Обычно для использования SSL или обслуживания библиотеки calibre как части более крупного сайта. Основной метод - запустить сервер calibre и настроить к нему обратный прокси с главного сервера.

Обратный прокси-сервер - это когда ваш обычный сервер принимает входящие запросы и передает их серверу calibre. Затем он считывает ответ от сервера calibre и пересылает его клиенту. Это означает, что вы можете просто запустить сервер calibre как обычно, не пытаясь тесно интегрировать его с вашим основным сервером.

6.6.1 Использование полностью виртуального хоста

Самая простая конфигурация - выделить полноценный виртуальный хост серверу calibre. В этом случае запустите сервер calibre как:

```
calibre-server
```

Теперь настройте виртуальный хост на вашем основном сервере, например, для nginx:

```
http {
    client_max_body_size 64M;  # needed to upload large books
}

server {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    listen [::]:80;
    server_name myserver.example.com;

    location / {
        proxy_pass http://127.0.0.1:8080;
    }
}
```

Или для 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 Использование префикса URL-адреса

Если вы не хотите выделять полный виртуальный хост для calibre, вы можете использовать префикс URL. Запустите сервер calibre как:

```
calibre-server --url-prefix /calibre --port 8080
```

Ключевым параметром здесь является `--url-prefix/calibre`. В результате сервер содержимого будет обслуживать все URL-адреса с префиксом `/calibre`. Чтобы увидеть это в действии, посетите в браузере `http://localhost:8080/calibre`. Вы должны увидеть обычный веб-сайт сервера контента, но теперь он будет работать под именем `/calibre`.

Для nginx необходимая конфигурация:

```
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;
}
```

Для Apache сначала включите прокси-модули в Apache, добавив следующее в `httpd.conf`:

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

Точный способ включения прокси-модулей будет зависеть от вашей установки Apache. После включения прокси-модулей добавьте следующие правила в `httpd.conf` (или, если вы используете виртуальные хосты, в файл conf для рассматриваемого виртуального хоста):

```
AllowEncodedSlashes On
RewriteEngine on
RewriteRule ^/calibre/(.*) http://127.0.0.1:8080/calibre/$1 [proxy]
RedirectMatch permanent ^/calibre$ /calibre/
```

Вот и всё, теперь вы сможете получить доступ к серверу содержимого calibre по URL-адресу `/calibre` на вашем основном сервере. Вышеупомянутые правила передают все запросы из раздела `/calibre` серверу calibre, работающему на порту 8080, и благодаря указанному выше параметру `--url-prefix` сервер calibre обрабатывает их прозрачно.

Примечание

При использовании обратного прокси-сервера вы должны указать серверу контента calibre прослушивать только локальный хост, используя параметр `--listen-on 127.0.0.1`. Таким образом, сервер будет прослушивать соединения только с того же компьютера, то есть с обратного прокси.

Примечание

Если вы настроили SSL для вашего основного сервера, вы должны указать серверу calibre использовать базовую аутентификацию вместо digest-аутентификации, так как это быстрее. Для этого передайте параметр `--auth-mode=basic` в `calibre-server`.

6.7 Создание сервиса для сервера calibre в современной системе Linux

Вы можете легко создать службу для запуска calibre при загрузке в современной системе Linux (systemd[34]). Просто создайте файл `/etc/systemd/system/calibre-server.service` с содержимым, показанным ниже:

```
[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
```

Измените `mylinuxuser` и `mylinuxgroup` на любого пользователя и группу, от имени которых будет работать сервер. Это должны быть тот же пользователь и группа, которым принадлежат файлы в папке библиотеки calibre. Обратите внимание, что обычно запускать сервер от имени пользователя root - не лучшая идея. Также измените путь к папке библиотеки calibre в соответствии с вашей системой. При необходимости вы можете добавить

[34] https://www.freedesktop.org/wiki/Software/systemd/

несколько библиотек. См. Справку по команде `calibre-server`.

Теперь запустите:

```
sudo systemctl start calibre-server
```

для запуска сервера. Проверьте его статус:

```
sudo systemctl status calibre-server
```

To make it start at boot, run:

```
sudo systemctl enable calibre-server
```

Примечание

Серверу calibre *не* нужен работающий X-сервер, но ему нужны установленные X-библиотеки, поскольку некоторые компоненты, которые он использует, связаны с ними.

Примечание

Сервер calibre также поддерживает активацию сокета systemd, так что вы также можете использовать его при необходимости.

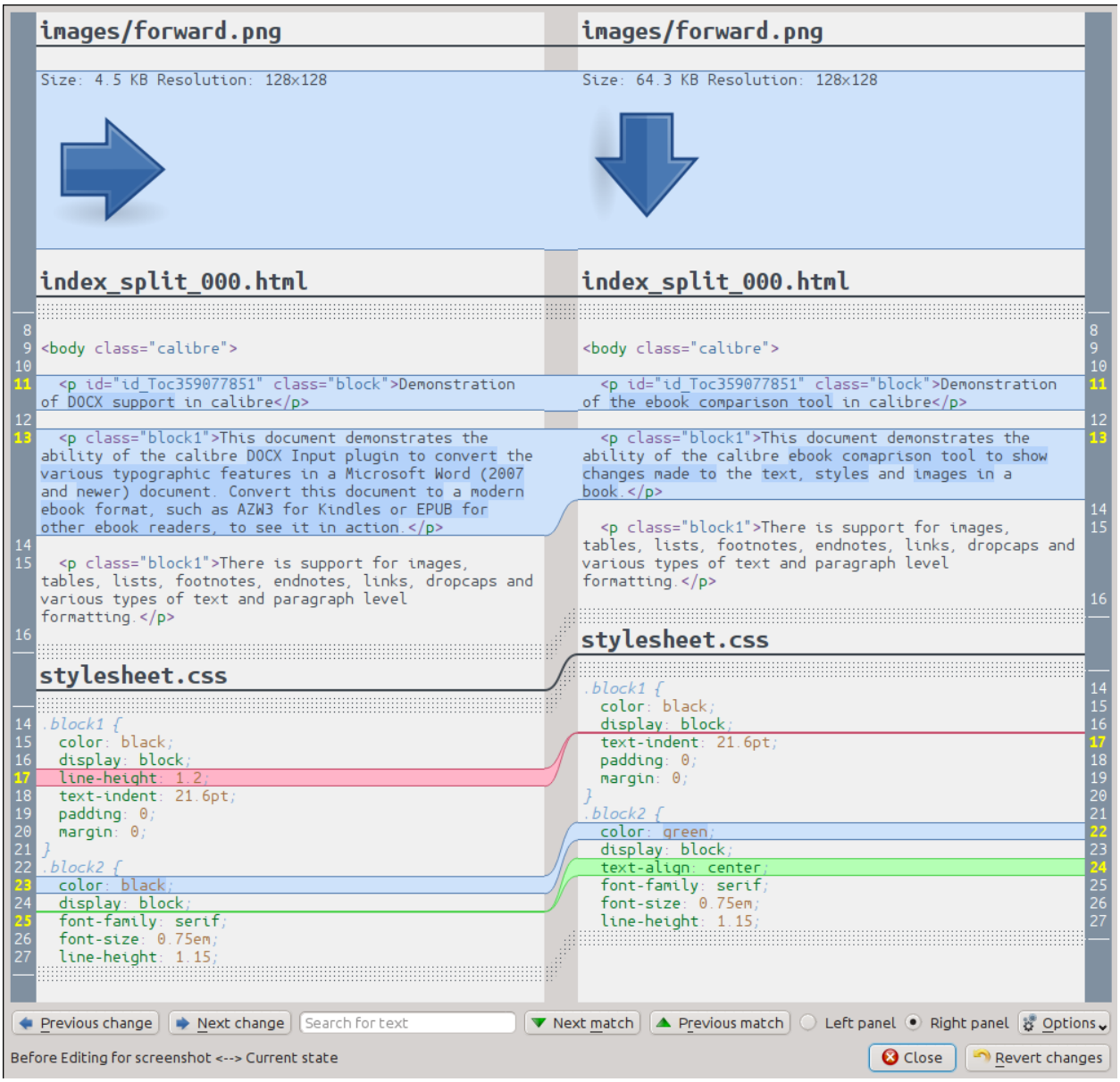
ГЛАВА 7

Сравнение книг

В calibre есть встроенный инструмент сравнения электронных книг, с помощью которого можно посмотреть, что изменилось внутри книги после редактирования или преобразования. Он может сравнивать книги в форматах EPUB и AZW3.

Чтобы работать с ним, либо откройте электронную книгу в инструменте *Редактирование книг* (страница 81), а затем нажмите *Файл → Сравнить с другой книгой*, либо воспользуйтесь панелью *Описание книги* (страница 21). Если вы выполните преобразование из EPUB в EPUB, исходный файл сохранится как ORIGINAL_EPUB. Просто щёлкните правой кнопкой мыши по записи ORIGINAL_EPUB на панели описания книги и выберите *Сравнить с форматом EPUB*.

Открывшийся инструмент сравнения будет выглядеть, как на скриншоте ниже. В нём отображаются различия в тексте, стилях и изображениях для выбранных книг.



7.1 Что и где показано в окне сравнения

Как видно на скриншоте выше, вид сравнение показывает различия между стороне две книги о бок. Только различия, с несколькими строками контекста вокруг них приведены. Это делает его легко увидеть с первого взгляда только то, что было изменено в большом документе, как книгу.

Добавил текст отображается на зеленом фоне, удалить текст на красном фоне и измененного текста с синим фоном.

Номера строк всего измененного текста являются шов по бокам, что дает ему легко перейти к определенной изменения в редакторе. Когда вы открываете инструмент сравнения в редакторе, вы также можете сделать двойной щелчок на линии в правой панели, чтобы перейти к этой линии в редакторе автоматически.

Один полезный метод при сравнении книги, чтобы рассказать инструмент сравнения, чтобы украсить текст и стиль файла перед вычислением разницы. Это часто может привести к чище и проще следовать различия. Чтобы сделать это, нажмите: guilabel: *options* кнопку в правом нижнем углу и выберите: guilabel:` Украсить файлы перед

comparing`. Следует отметить, что иногда может украшении есть нежелательные эффекты, так как это может привести к недействительным разметки должна быть изменена, чтобы сделать его действительным. Вы также можете изменить количество строк контекста, показанных вокруг различий через: guilabel: *options* кнопки.

Вы можете искать любой текст в различиях через панель поиска внизу. Вам нужно будет указать, в какой панели искать, *Left* или *Right*.

7.2 Запуск инструмента сравнения

Наиболее удобен инструмент сравнения в том случае, когда у Вас имеется два варианта одной и той же книги и Вы хотите понять разницу между ними. Для этого есть несколько возможностей запуска инструмента

7.2.1 Сравнение двух файлов электронных книг

Откройте первый файл, используя Редактировать. Теперь нажмите *Файл → Сравнить с другим файлом* и выберите второй файл (он должен быть в том же формате, что и первый). Просмотр сравнения откроется в редактируемом файле справа и слева во втором файле.

7.2.2 Сравнение ORIGINAL_FMT и FMT

Когда вы в calibre преобразовываете из FMT в этот же формат, исходный файл сохраняется как ORIGINAL_FMT. Увидеть, что изменилось после преобразования, можно, щёлкнув правой кнопкой мыши по записи ORIGINAL_FMT на панели «*Описание книги* (страница 21)» в главном окне calibre и выбрав «*Сравнить с FMT*». Откроется окно сравнения с ORIGINAL_FMT слева и FMT справа.

7.2.3 Сравнение контрольной точки с текущим состоянием книги при редактировании

Инструмент *Редактирование книг* (страница 81) имеет очень полезную функцию *Контрольные точки* (страница 94). Это позволяет сохранить текущее состояние книги в качестве именованной контрольной точки *, на которую вы можете вернуться, если вам не нравятся изменения, сделанные вами после создания контрольной точки. Контрольные точки также создаются автоматически при выполнении различных автоматических действий в редакторе. Вы можете просмотреть список контрольных точек, перейдя по ссылке *View → Checkpoints*, а затем используйте кнопку *Compare*, чтобы сравнить книгу с выбранной контрольной точкой с текущим состоянием. Инструмент сравнения покажет контрольную точку слева и текущее состояние справа.

ГЛАВА 8

Редактирование метаданных книги

Содержание

- *Редактирование метаданных одной книги за раз* (страница 127)
  - *Загрузка метаданных* (страница 128)
  - *Управление форматами книг* (страница 128)
  - *Всё об обложках* (страница 128)
- *Редактирование метаданных нескольких книг одновременно* (страница 128)
  - *Поиск и замена* (страница 129)
  - *Массовая загрузка метаданных* (страница 130)
- *Добавить дополнительные файлы данных в книгу* (страница 130)

Электронные книги бывают разных форм и размеров, и чаще всего их метаданные (такие как название/автор/серия/издатель) являются неполными или неправильными. Самый простой способ изменить метаданные в calibre - это просто дважды щелкнуть по записи и ввести правильную замену. Для более сложного «мощного редактирования» используйте инструменты редактирования метаданных, описанные ниже.

8.1 Редактирование метаданных одной книги за раз

Нажмите на книгу, которую хотите отредактировать, а затем нажмите кнопку *Редактировать метаданные* или нажмите клавишу `E`. Откроется диалоговое окно, которое позволяет редактировать все аспекты метаданных. В нём есть различные функции для быстрого и эффективного редактирования. Список часто используемых советов:

- Вы можете нажать кнопку между заголовком и авторами, чтобы поменять их местами автоматически.
- Вы можете нажать кнопку рядом с сортировкой по автору, чтобы calibre автоматически заполнил её, используя значения сортировки, сохранённые у каждого автора. Используйте *Управление авторами*, чтобы

увидеть и изменить значения сортировки авторов. Это диалоговое окно можно открыть, нажав и удерживая кнопку рядом с сортировкой автора.

- Вы можете нажать кнопку рядом с тегами, чтобы использовать *Редактор тегов* для управления тегами, связанными с книгой.
- Поле «Идентификаторы» можно использовать для ввода ISBN (и многих других типов идентификаторов), красный фон в поле, если вы введете неправильный ISBN. Зелёный фон в поле для действительных номеров ISBN.
- Поле сортировки автора будет красным, если значение сортировки автора отличается от того, что, по мнению calibre, должно быть.

8.1.1 Загрузка метаданных

Самая приятная особенность диалога редактирования метаданных — это возможность автозаполнения полей метаданными, полученными с различных веб-сайтов. Сейчас calibre использует Google Книги и Amazon. Загрузка метаданных может заполнить название, автора, серии, теги, оценку, описание и ISBN.

Чтобы использовать загрузку, заполните поля заголовка и автора и нажмите кнопку :guilabel: *Получить метаданные*. calibre представит вам список книг, наиболее соответствующих названию и автору. Если вы сначала заполните поле ISBN, оно будет использовано вместо названия и автора. Если совпадений не найдено, попробуйте сделать поиск немного менее конкретным, включив в него только некоторые ключевые слова и только фамилию автора.

8.1.2 Управление форматами книг

В calibre с одной записью книги может быть связано много разных *форматов*. Например, вы, возможно, получили Полное собрание сочинений Шекспира в формате EPUB, а затем преобразовали его в MOBI для чтения на Kindle. calibre автоматически управляет несколькими форматами для вас. В разделе :guilabel: *Доступные форматы* диалога Редактировать метаданные вы можете управлять этими форматами. Можно добавить новый формат, удалить существующий формат, а также попросить calibre задать метаданные и заполнить запись книги метаданными в одном из форматов.

8.1.3 Всё об обложках

Вы можете попросить calibre загрузить обложки для вас, при условии, что книга имеет известный номер ISBN. Кроме того, вы можете указать файл на вашем компьютере для использования в качестве обложки. calibre может даже создать обложку по умолчанию с основными метаданными для вас. Вы можете перетащить изображение на обложку, чтобы изменить её, а также щелкнуть правой кнопкой мыши, чтобы скопировать/вставить изображение обложки.

Кроме того, есть кнопка для автоматической обрезки границ обложки, если изображение на обложке имеет уродливую рамку.

8.2 Редактирование метаданных нескольких книг одновременно

Сначала выберите книги для редактирования, удерживая `Ctrl` или `Shift` и щёлкая по ним. При выборе более одной книги нажатие кнопки *Редактировать метаданные* откроет диалоговое окно *Массового* редактирования метаданных. В этом окне вы можете быстро установить для набора книг одно и то же значение автора/издателя/рейтинга/тега/серии и т. д. Это особенно полезно, если вы только что импортировали несколько книг с общими метаданными. Это очень мощное диалоговое окно, например, оно имеет вкладку «Поиск и замена», которую можно использовать для выполнения массовых операций с метаданными и даже для копирования метаданных из одного столбца в другой.

В обычном диалоговом окне редактирования метаданных также есть кнопки *Следующая* и *Предыдущая*, которые можно использовать для редактирования метаданных нескольких книг друг за другом.

8.2.1 Поиск и замена

Диалог *Массовое редактирование метаданных* позволяет произвольно выполнять мощные операции поиска и замены выбранных книг. По умолчанию он использует простой текстовый поиск и замену, но также поддерживает *регулярные выражения*. Для получения дополнительной информации о регулярных выражениях см. *Всё об использования регулярных выражений в calibre* (страница 221).

Как отмечалось выше, существует два режима поиска и замены: сопоставление символов и регулярное выражение. Сопоставление символов - это символы, вводимые в поле «Поиск», и замена их теми символами, которые вы вводите в поле «Заменить на». Каждое вхождение символов поиска в поле будет заменено. Например, предположим, что искомое поле содержит *a bad cat*. Если вы ищете *a* для замены на *HELLO*, то результатом будет *HELLO bHELLOd cHELLOt*.

Если поле, по которому вы ищете, является «множественным» полем, подобным тегам, то каждый тег обрабатывается отдельно. Например, если ваши теги содержат *Horror, Scary*, поисковое выражение *r,* не будет ни с чем совпадать, потому что выражение сначала будет применено к *Horror*, а затем к *Scary*.

Если вы хотите, чтобы при поиске игнорировались различия в верхнем/нижнем регистре, снимите флажок «Учитывать регистр».

Вы можете указать calibre изменить регистр в результате (информация после замены), выбрав одну из функций из поля «Применить после замены». Доступные операции:

- *Нижний регистр* - изменить все символы в поле на нижний регистр
- *Верхний регистр* - изменить все символы в поле на верхний регистр
- «Регистр Заголовков» - капитализировать каждое слово в результате.

Поле «Ваш тест» предназначено для ввода текста, чтобы проверить, выполняет ли поиск/замена то, что вы хотите. В большинстве случаев достаточно блоков с книжными тестами, но возможно, что есть случай, который вы хотите проверить, который не показан в этих полях. Введите этот случай в «Ваш тест».

Режим регулярных выражений имеет некоторые отличия от режима символов, помимо (конечно) использования регулярных выражений. Во-первых, функции применяются к частям строки, совпадающим со строкой поиска, а не ко всему полю. Во-вторых, функции применяются к строке замены, а не ко всему полю.

Третье и самое важное - замена строки может ссылаться на части строки поиска с помощью обратных ссылок. Обратной ссылкой является `\\n`, где n - целое число, которое ссылается на n-ную группу в скобках в поисковом выражении. Например, если использовать тот же пример, что и выше, *a bad cat*, поисковое выражение *a (…) (…) `и выражение замены `a \2 \1*, результат будет *a cat bad*. Пожалуйста, смотрите *Всё об использования регулярных выражений в calibre* (страница 221) для получения дополнительной информации о обратных ссылках.

Один полезный шаблон: предположим, что вы хотите изменить регистр всего поля. Самый простой способ сделать это - использовать символьный режим, но давайте предположим, что вы хотите использовать режим регулярных выражений. Выражение поиска должно быть *(^.*$)*, выражение замены должно быть *\1*, и должна быть выбрана нужная функция изменения регистра.

Наконец, в режиме регулярных выражений вы можете копировать значения из одного поля в другое. Просто измените поле источника и назначения. Копия может заменить поле назначения, добавить его к полю (добавить в начало) или добавить в поле (добавить в конце). Флажок «использовать запятую» указывает calibre (или не добавлять) запятую между текстом и полем назначения в режимах prepend и append. Если пунктов назначения несколько (например, теги), вы не можете снять этот флажок.

Поиск и замена выполняется после применения всех других изменений метаданных в других вкладках. Это может привести к некоторой путанице, потому что тестовые окна будут показывать информацию до других изменений, но операция будет применена после других изменений. Если у вас есть какие-либо сомнения относительно того, что произойдет, не смешивайте поиск/замену с другими изменениями.

8.2.2 Массовая загрузка метаданных

Если вы хотите загрузить метаданные для нескольких книг одновременно, щелкните правой кнопкой мыши кнопку :guilabel: *Изменить метаданные* и выберите *Загрузить метаданные*. Вы можете загружать только метаданные, только обложки или и то, и другое.

8.3 Добавить дополнительные файлы данных в книгу

calibre может хранить любое количество дополнительных файлов данных, связанных с книгой. Это могут быть альтернативные обложки, дополнительный материал и т. д. Их нельзя просматривать напрямую или использовать в качестве источников конверсии. Они также не индексируются полнотекстовой поисковой системой в calibre. Чтобы просмотреть/добавить/удалить их, выберите книгу и щелкните правой кнопкой мыши кнопку *Редактировать метаданные* и выберите *Управление файлами данных*. Появится всплывающее окно, в котором вы можете выполнять операции с этими файлами. Кроме того, вы можете щелкнуть правой кнопкой мыши по кнопке *Добавить книги* и выбрать *Добавить файлы данных в выбранные записи книг*, чтобы быстрее добавить файлы данных.

ГЛАВА 9

Часто задаваемые вопросы

Содержание

- *Конвертация форматов электронных книг* (страница 131)
- *Интеграция устройств* (страница 134)
- *Управление библиотекой* (страница 142)
- *Разное* (страница 147)

9.1 Конвертация форматов электронных книг

Содержание

- *Какие форматы конвертации поддерживает calibre?* (страница 132)
- *Каковы оптимальные исходные форматы для конвертации?* (страница 132)
- *Я преобразовал PDF файл, но в результате имеются разные проблемы?* (страница 132)
- *Как конвертировать мой файл, если он включает не английские символы или «умные» кавычки?* (страница 132)
- *Что случилось с оглавлением MOBI-файлов?* (страница 133)
- *Каким образом можно преобразовывать коллекцию HTML-файлов в определенной последовательности?* (страница 133)
- *Если EPUB, созданный в calibre, не работает?* (страница 134)

- *Как я могу использовать некоторые из расширенных возможностей инструментов конвертации?* (страница 134)

9.1.1 Какие форматы конвертации поддерживает calibre?

calibre поддерживает конвертацию множества форматов из одного в другой. Возможно конвертировать каждый входной формат в приведенном ниже списке в каждый формат вывода.

Входные форматы: 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

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

Примечание

PRC - общий формат, calibre поддерживает файлы PRC с заголовками TextRead и MOBIBook. PDB также - общий формат, calibre поддерживает файлы eReader, Plucker, PML и zTxt PDB. DJVU формат поддерживается только для преобразования DJVU-файлов, содержащих встроенный текст. Они, как правило, генерируются с помощью программного обеспечения для OCR. MOBI-книги могут быть двух типов: Mobi6 и KF8. calibre полностью поддерживает оба. MOBI-файлы часто имеют расширение .azw или .azw3. Файлы .DOCX поддерживаются начиная с Microsoft Word 2007 и все более новые версии.

9.1.2 Каковы оптимальные исходные форматы для конвертации?

В порядке убывания предпочтения: LIT, MOBI, AZW, EPUB, KEPUB, AZW3, FB2, FBZ, DOCX, HTML, PRC, ODT, RTF, PDB, TXT, PDF

9.1.3 Я преобразовал PDF файл, но в результате имеются разные проблемы?

PDF — это ужасный формат для начальной конвертации. Для списка различных вопросов, с которыми придется столкнуться при конвертировании PDF, см.: *Преобразование PDF документов* (страница 76).

9.1.4 Как конвертировать мой файл, если он включает не английские символы или «умные» кавычки?

Существует два аспекта этой проблемы:

1. Знание кодировки исходного файла: calibre пробует предположить кодировку, которая используется в вашем исходном файле, но это зачастую невозможно, так что вы должны сообщить, какую кодировку использовать. В графическом интерфейсе это указывается через поле *Исходная кодировка символов* в *Оформление → Текст* в Общих настройках диалога конвертации. В утилитах командной строки - параметр `ebook-convert-txt-input --input-encoding`

2. При добавлении файлов HTML в calibre вам может потребоваться указать calibre, в какой кодировке находятся файлы. Для этого перейдите в *Preferences-Advanced-Plugins-File type* и настройте плагин *HTML to ZIP* , указав кодировку ваших HTML-файлов. Теперь, когда вы добавляете HTML-файлы в calibre, они будут правильно обрабатываться. HTML-файлы из разных источников часто имеют разные кодировки, поэтому вам, возможно, придется менять этот параметр несколько раз. Обычная кодировка для многих некириллических файлов из Интернета - `cp1252`, сначала попробуйте её. Заметьте, что при преобразовании файлов HTML нужно оставить указанную выше настройку кодировки ввода пустым. Это связано с тем, что плагин *HTML to ZIP* автоматически преобразует файлы HTML в стандартную кодировку (UTF-8).

9.1.5 Что случилось с оглавлением MOBI-файлов?

Первое, что нужно понимать, - в большинстве электронных книг два оглавление. Одно - традиционное оглавление, как в бумажных книгах. Это Оглавление - часть главного потока документа и может быть стилизовано как вам нравится. Это Оглавление называется *Оглавление содержимого*.

Кроме того есть *Оглавление метаданных*. Оглавление метаданных не является частью текста книги и обычно доступно при помощи специальной кнопки в читалке. Например в calibre viewer используется кнопка Показать оглавление. Это Оглавление не может быть стилизовано создателем книги. Его представление - задача просмотрщика.

В формате MOBI ситуация немного запутаннее. Т. к. формат MOBI *не* слишком хорошо поддерживает Оглавления метаданных. Книга MOBI эмулирует наличие Оглавления метаданных, добавляя *дополнительное* Оглавление содержимого в конце книги. Когда вы нажимаете Перейти к Оглавлению на Kindle, он показывает вам дополнительное Оглавление содержимого.

Сейча хорошо видно, что эта книга MOBI содержит два идентичных Оглавления. Помните, что одно из них - семантически Оглавление содержимого, а другое - Оглавление метаданных, даже если обма содержат одинаковые записи и выглядят одинаково. Одно может быть вызвано прямо из меню Kindle, другое -нет.

При конвертировании в MOBI, calibre находит *Оглавление метаданных* в исходом документе и генерирует Оглавление в-конце-файла в выходном файле MOBI. Вы можете отключить этот параметр в Выходных настройках MOBI. Также calibre можно указать добавлять его в начало или в конец книги через параметр в Выходных настройках MOBI. Помните, что семантически оно - Оглавление метаданных, в любом формате кроме MOBI оно *не может быть частью текста*. То, что это часть текста в MOBI - случайность, связанная с ограничениями формата. Если вы хотите получить Оглавление в определённом месте текста вашего документа, создайте его вручную. Также настоятельно рекомендуется оставить всё по умолчанию, то есть помещать Оглавление метаданных в конце книги. Заметьте, что если вы отключаете создание Оглавления в-конце-файла, получившийся файл MOBI может неправильно функционировать в Kindle, поскольку Kindle использует Оглавление метаданных для многих вещей, включая переворот страницы.

Если вам неоходимо вручную редактировать оглавление в исходном документе, вы можете использовать настройки определения Оглавления в calibre для автоматической генерации Оглавления метаданных. Смотрите раздел Конвертация в Руководстве пользователя для подробной информации о том, как использовать эти настройки.

В конце, я призываю вас убрать Оглавление содержимого и оставить в ваших электронных книгах только Оглавление метаданных. Оглавление метаданных даст читателям наилучшую навигацию (исключая Kindle, где оно, по существу, совпадает с Оглавлением содержимого).

Примечание

Новый формат AZW3 имеет соответствующую поддержку Оглавления метаданных. Однако, прошивка Kindle может функционировать неправильно, если вы отключаете генерацию Оглавления в-конце-файла. Так что рекомендуется не менять сгенерированное Оглавление. Если вы создаёте файл AZW3 с Оглавлением метаданных и без Оглавления в-конце-файла, некоторые возможности, такие как Page Flip, в Kindle

9.1.6 Каким образом можно преобразовывать коллекцию HTML-файлов в определенной последовательности?

Чтобы преобразовать коллекцию HTML файлов в определенном порядке, вам нужно создать файл со списком содержимого. Это еще один файл HTML, содержащий ссылки на все другие файлы в желаемом порядке. Подобный файл выглядит так:

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

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

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

Затем просто добавьте этот файл HTML в графическом интерфейсе и нажмите кнопку *Convert*, чтобы создать электронную книгу. Вы можете использовать опцию в разделе Оглавление диалога конвертации, чтобы управлять созданием Оглавления.

Примечание

По умолчанию, при добавлении файлов HTML, calibre следует по ссылкам в файлах в порядке *поиск в глубину*. Это означает, что если файл A.html ссылается на B.html, C.html и D.html, но B.html также ссылается на D.html, то файлы будут иметь следующий порядок: A.html, B.html, D. .html, C.html. Если вместо этого вы хотите, чтобы порядок был A.html, B.html, C.html, D.html, тогда вы должны указать calibre на добавление файлов в порядке *поиск в ширину*. Сделайте это, перейдя в *Preferences-Advanced-Plugins-File type* и настроив плагин *HTML to ZIP*.

9.1.7 Если EPUB, созданный в calibre, не работает?

calibre не гарантирует, что созданный им EPUB получится валидным. Единственное, в чём можно быть уверенным - это то, что если вы скормите ему валидные XHTML 1.1 + CSS 2.1, он выдаст валидный EPUB. calibre старается убедиться, что файлы EPUB, которые он производит действительно работают как задумано на широком разнообразии устройств, что несовместимо с созданием валидных EPUB, но преследует цель, которая намного важнее для большинства пользователей. Если вам нужен инструмент, который всегда должен производить валидные EPUB, то calibre не для вас. Это означает, что если вы хотите отправить созданный calibre EPUB в онлайн магазин, использующий проверку валидности EPUB, вам следует убедиться в валидности файла самостоятельно, calibre не сделает это за вас. Другими словами, вы должны скармливать на вход calibre валидные XHTML + CSS документы.

9.1.8 Как я могу использовать некоторые из расширенных возможностей инструментов конвертации?

Вы можете получить помощь по любым возможностям конверторов, подведя на них курсор мыши в графическом интерфейсе или выполнив команду `ebook-convert dummy.html .epub -h` в командной строке. Лучше всего начать с просмотра следующего демо файла, который демонстрирует некоторые из расширенных возможностей html-demo.zip[35].

9.2 Интеграция устройств

Содержание

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

- *Какие устройства поддерживает calibre?* (страница 135)
- *Как я мог помочь получению поддержки моего устройства в calibre?* (страница 135)
- *Моё устройство не определяется в calibre?* (страница 136)
- *Моё устройство нестандартное или необычное. Что я могу сделать, чтобы подключить его?* (страница 136)
- *Как я могу использовать calibre с моим iPad/iPhone/iPod touch?* (страница 136)
- *Как использовать calibre с телефоном/планшетом Android или Kindle Fire?* (страница 137)
- *Могу ли я получить доступ к своим книгам calibre с помощью веб-браузера на Kindle или другом устройстве для чтения?* (страница 138)
- *Я не могу посылать электронную почту при помощи calibre?* (страница 139)
- *Мое устройство монтируется в Linux только для чтения, поэтому calibre не может подключиться к нему?* (страница 140)
- *Почему calibre не поддерживает коллекции на Kindle или полку на Nook?* (страница 140)
- *Я получаю сообщение об ошибке, когда пытаюсь использовать calibre с моим Kobo Touch/Glo и т. д.?* (страница 140)
- *Обложки для книг, которые я отправляю на свой Kindle для электронных чернил, появляются на мгновение, а затем заменяются универсальной обложкой* (страница 141)
- *Обложки для книг, отправленных на мой Kindle ColorSoft и новее, не отображаются в списке книг?* (страница 141)
- *Обложки моих MOBI-файлов перестали отображаться в Kindle для PC/Kindle, Android/iPad и т.п.* (страница 141)
- *Я перекинул несколько книг в мой Kindle при помощи calibre и их не видно?* (страница 142)

9.2.1 Какие устройства поддерживает calibre?

calibre может напрямую подключаться ко всем основным устройствам чтения электронных книг (и большинству второстепенных), смартфонам, планшетам и т. д. Кроме того, с помощью функции в Тонкой настройке *Connect to folder* можно использовать подключать любую читалку, представляющуюся как USB-диск. Наконец, можно подключиться по беспроводной сети к любому устройству с веб-браузером, используя сервер контента calibre.

9.2.2 Как я мог помочь получению поддержки моего устройства в calibre?

Если ваше устройство определяется операционной системой как USB диск, то добавить его поддержку в calibre не составит труда. Нам только нужно немного информации от вас:

- Полный список форматов электронных книг, которые поддерживает ваше устройство.
- Есть ли на устройстве специальная папка, в которую должны быть помещены все файлы электронных книг? Устройство обнаруживает файлы, помещенные в подпапки?
- Также нам потребуется информация о вашем устройстве, которую calibre соберёт автоматически. Сначала, если ваше устройство поддерживает карты SD, вставьте их. Затем подключите устройство к компьютеру. В calibre перейдите в *Параметры → Разное* и нажмите кнопку «Обнаружение устройства в режиме отладки». Это создаст некий отладочный вывод. Скопируйте его в файл и повторите процесс, на этот раз с отключенным от вашего компьютера устройством.

- Отправьте оба упомянутых вывода нам с прочей информацией и мы напишем драйвер для вашего устройства.

Как только вы отправите нам выходные данные для конкретной операционной системы, поддержка устройства в этой операционной системе появится в следующем выпуске calibre. Чтобы отправить нам результат, откройте отчет об ошибке и прикрепите к нему вывод. Смотрите how to report bugs[36].

9.2.3 Моё устройство не определяется в calibre?

Следуйте этим инструкциям, чтобы найти проблему:

- Убедитесь, что одновременно к вашему компьютеру вы присоединяете только одно устройство. Не держите подключенными в это время другие поддерживаемые calibre устройства, такие как iPhone/iPad и прочие.
- Если вы подключаете Apple iDevice (iPad, iPod Touch, iPhone), Apple больше не позволяет стороннему программному обеспечению подключаться к своим устройствам с помощью кабеля USB. Вместо этого используйте беспроводное соединение через сервер контента calibre.
- Если вы подключаете 2024 Kindle или другое устройство Android, и они управляются macOS или Linux, прочтите примечание *Используя кабель USB* (страница 137).
- В macOS, если вы получаете ошибки разрешений при подключении устройства к calibre, можно исправить это: guilabel: *Системные настройки -> безопасность и конфиденциальность -> Файлы и папки*.
- Убедитесь, что у вас установлена последняя версия calibre (текущая 8.5.0). Последнюю версию всегда можно загрузить с website:*the calibre website <download>*. Вы можете узнать, какую версию calibre вы используете в данный момент, посмотрев на нижнюю строку главного окна calibre.
- Убедитесь, что ваша операционная система видит устройство. То есть устройство должно отображаться в Проводнике (в Windows) или Finder (в macOS).
- В calibre перейдите по ссылке *Параметры → Общий доступ к файлам → Игнорируемые устройства* и убедитесь, что ваше устройство не игнорируется.
- Если все вышеперечисленные шаги завершились неудачно, перейдите в *Параметры → Разное* и нажмите *Обнаружение устройства в режиме отладки*, подключив ваше устройство, после этого опубликуйте результат в виде тикета на <https://bugs.launchpad.net/calibre>`_.

9.2.4 Моё устройство нестандартное или необычное. Что я могу сделать, чтобы подключить его?

В дополнение к функции: guilabel: *Connect to folder*, находящейся под кнопкой: guilabel:` Connect/share`, calibre предоставляет подключаемый модуль устройства, определяемый пользователем, который можно использовать для подключения к любому обнаруженному USB-устройству, как к диску в операционной системе. Примечание: в Windows устройство должно иметь букву диска, чтобы calibre мог его использовать. См. в `Настройки -> Плагины -> Интерфейс устройства -> Folder Device Interface` и `Настройки -> Разное -> Получить информацию для установки пользовательского устройства`. Обратите внимание, что если используется пользовательский плагин для устройства, обычно определяемого встроенным плагином calibre, сначала нужно отключить встроенный плагин, для использования вместо него пользовательского.

9.2.5 Как я могу использовать calibre с моим iPad/iPhone/iPod touch?

Лёгкий способ просмотреть коллекцию calibre с устройства Apple - использовать *Контент-сервер calibre* (страница 115), для доступа к коллекции из сети. Сначала выполните следующие шаги в calibre

- Установить предпочитаемый формат вывода как EPUB (формат вывода можно установить в *Настройки → Интерфейс → Поведение*)

[36] https://calibre-ebook.com/ru/bugs

- Установить профиль вывода на iPad (также работает для iPhone/iPod) в разделе *Параметры → Конвертация → Общие настройки → Настройка страницы*
- Конвертируйте книги для чтения на i-Pad/Phone/Pod в формат EPUB выделив их и нажав *Преобразовать книги*.
- Включите сервер содержимого, нажав кнопку *Подключить/поделиться*, и оставьте calibre запущенным. Также можно настроить автоматический запуск сервера контента с помощью *Настройки → Сетевой доступ → Запускать сервер автоматически при запуске calibre*.

Сервер содержимого позволяет читать книги непосредственно в Safari. Кроме того, для вашего iDevice есть множество приложений, которые могут подключаться к серверу содержимого calibre. Примеры: Marvin, Mapleread, iBooks.

Использование сервера контента

Запустите браузер Safari и введите IP-адрес и порт компьютера, на котором запущен сервер calibre, например:

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

Замените `192.168.1.2` локальным IP-адресом компьютера с запущенной calibre. Подробнее о том, как запустить сервер и узнать правильный IP-адрес, читайте в разделе «*Контент-сервер calibre* (страница 115)».

Вы увидите список книг в Safari, нажмите на любую книгу, и вы сможете либо загрузить её, либо прочитать в самом браузере. Если вы решите загрузить её, Safari спросит вас, хотите ли вы открыть файл с помощью iBooks.

Многие приложения для чтения поддерживают просмотр библиотеки calibre напрямую через поддержку OPDS . В таких приложениях вы можете перейти на экран онлайн-каталога и добавить IP-адрес сервера calibre для просмотра и загрузки книг из библиотеки calibre в приложении.

9.2.6 Как использовать calibre с телефоном/планшетом Android или Kindle Fire?

Есть два способа подключить Android-устройство к calibre. С помощью кабеля USB или по беспроводной сети. Сначала установите на Android-устройство приложение для чтения электронных книг. Есть множество бесплатных и платных приложений для чтения электронных книг для Android. Некоторые примеры (в произвольном порядке): **`FBReader https://play.google.com/store/apps/details?id=org.geometerplus.zlibrary.ui .android & hl = ru`_**, **`Moon + https://play.google.com/store/apps/details?id=com.flyersoft.moonreader&hl=en`_**,` Mantano https://play.google.com/store /apps/details?id=com.mantano.reader.android.lite&hl=en`_, **`Aldiko https://play.google.com/store/apps/details?id=com.aldiko.android&hl=en`_** , **`Kindle https://play.google.com/store/apps/details?id=com.amazon.kindle&feature=related_apps`_**.

Используя кабель USB

Просто подключите устройство к компьютеру с помощью кабеля USB. calibre должен автоматически определить устройство, после чего вы сможете перенести на него книги, нажав кнопку *Отправить на устройство*. Обратите внимание, что в macOS и Linux только одна программа может подключаться к устройству Android одновременно, поэтому убедитесь, что устройство не открыто в диспетчере файлов ОС или в утилите передачи файлов Android и т.п.

Примечание

С новыми устройствами Android вам, возможно, придётся попрыгать с бубном, чтобы соединение заработало, т.к. Google действительно не хочет, вашей независимости от его облака. Сначала разблокируйте экран перед подключением кабеля USB. При подключении кабеля USB вы получите всплывающее уведомление.

Убедитесь, что написано что-то вроде «Передача файлов мультимедиа» или «MTP (режим передачи мультимедиа)». Если это не так, коснитесь уведомления и измените режим на Media Transfer (MTP). На этом этапе возможно потребоваться перезапустить calibre, чтобы ваше устройство было распознано. Если на вашем устройстве всплывающее окно с запросом разрешения появляется при каждой попытке подключения calibre или операционной системы к нему - жмите OK.

Примечание

В Kindle Fire 8 или новее при подключении USB-кабеля появляется значок, показывающий, что устройство заряжается. Нажмите на него и переключите устройство в режим передачи данных, а затем запустите calibre, после чего он должен быть обнаружен.

По беспроводному соединению

в calibre есть встроенный веб-сервер :doc:`calibre Content server<server> ` для доступа к коллекции calibre из сети. Для просмотра используйте браузер или специальное приложение. Вначале выполните следующие шаги в calibre:

- Установите *Предпочтительный выходной формат* в EPUB для обычных устройств Android или MOBI для Kindles (формат вывода можно установить в *Параметры → Интерфейс → Поведение*)
- Преобразуйте книги, которые вы хотите читать на своем устройстве, в формат EPUB/MOBI, выбрав их и нажав кнопку *Преобразовать книги*.
- Включите *Content server* в настройках calibre и оставьте calibre включенным.

Теперь на вашем устройстве Android откройте браузер и перейдите на:

http://192.168.1.2:8080/

Замените `192.168.1.2` локальным IP-адресом компьютера с запущенной calibre. Подробнее о том, как запустить сервер и узнать правильный IP-адрес, читайте в разделе «*Контент-сервер calibre* (страница 115)».

Теперь вы можете просматривать свою коллекцию книг и загружать книги с любого calibre на свое устройство, чтобы открывать их с помощью любого программного обеспечения для чтения электронных книг, установленном на Android-устройстве.

Многие приложения для чтения поддерживают просмотр библиотеки calibre напрямую через поддержку OPDS . В таких приложениях вы можете перейти на экран онлайн-каталога и добавить IP-адрес сервера calibre для просмотра и загрузки книг из библиотеки calibre в приложении.

9.2.7 Могу ли я получить доступ к своим книгам calibre с помощью веб-браузера на Kindle или другом устройстве для чтения?

В calibre есть *контент-сервер*, который представляет книги в calibre как веб-страницы. Подробнее см. «*Контент-сервер calibre* (страница 115)».

Некоторые устройства, такие как Kindle (1/2/DX), не позволяют получить доступ к порту 8080 (порт по умолчанию, на котором работает сервер контента). В этом случае измените порт в настройках calibre на 80. (В некоторых операционных системах вы не сможете запустить сервер на порте с номером меньше 1024 из-за настроек безопасности. В этом случае самым простым решением является настройка ваш маршрутизатор для пересылки запросов с порта 80 на порт 8080).

Также некоторые устройства не имеют браузеров, достаточно продвинутых для запуска интерфейса, подобного приложениям, используемого сервером содержимого. Для таких устройств вы можете просто добавить `/mobile` к URL-адресу сервера, чтобы получить упрощённый интерфейс без использования JavaScript.

9.2.8 Я не могу посылать электронную почту при помощи calibre?

Из-за большого количества спама в электронной почте отправка электронной почты может быть сложной, поскольку разные почтовые серверы используют разные стратегии для блокировки электронной почты. Наиболее распространенная проблема - если вы отправляете электронную почту напрямую из calibre (без почтового ретранслятора). Многие серверы (например, Amazon) блокируют электронную почту, которая не исходит от известного ретранслятора. Самый надежный способ настроить отправку электронной почты в calibre - это сделать следующее:

- Создать бесплатный GMX аккаунт в GMX[37].
- Перейти к *Настройки → Общий доступ к файлам → Пересылка книг по электронной почте* в calibre, нажать кнопку *Использовать GMX* и ввести запрашиваемую информацию.
- Войти в свою учётную запись GMX на веб-сайте и включить отправку SMTP (*Настройки-POP3 и IMAP-Отправлять и получать электронные письма через внешнюю программу*)
- После этого calibre сможет использовать GMX для отправки почты.
- Если вы отправляете книги на Kindle, не забудьте обновить настройки электронной почты на странице Amazon Kindle, чтобы разрешить отправку электронной почты с вашего адреса электронной почты GMX. Также обратите внимание, что Amazon не разрешает доставку по электронной почте файлов MOBI формата AZW3 и нового формата (KF8). Наконец, Amazon недавно начал отправлять электронные письма с подтверждением, которые вы должны нажать обратно в свою учетную запись GMX, прежде чем книга будет фактически доставлена.

Даже после этого у вас могут возникнуть проблемы. Одним из распространенных источников проблем является то, что отдельные плохо спроектированные антивирусы блокируют открытие соединения calibre-ом для отправки электронной почты. Попробуйте добавить исключение для calibre в антивирус.

Примечание

Microsoft / Google / GMX может отключить вашу учетную запись, если вы используете её для отправки большого количества писем. Таким образом, при использовании этих сервисов для отправки почты calibre автоматически ограничивается отправкой одной книги каждые пять минут. Если вы не возражаете против того, чтобы ваша учётная запись была заблокирована, вы можете сократить этот интервал ожидания, перейдя по ссылке *Preferences-Advanced-Tweaks* в calibre.

Примечание

Google недавно умышленно отключили поддержку протокола рассылки сообщений (SMTP) чтобы заставить всех использовать их веб-интерфейс с показом вам рекламы. Они объявили SMTP незащищённым, что некорректно. Используйте дроугого провайдера электропочты.

Примечание

Если вас беспокоит предоставление доступа к своей учётной записи электронной почты для calibre, просто создайте новую бесплатную учётную запись электронной почты с помощью GMX или Outlook и используйте её только для calibre.

[37] https://www.gmx.com

9.2.9 Мое устройство монтируется в Linux только для чтения, поэтому calibre не может подключиться к нему?

Ядра Linux монтируют устройства только для чтения, если в их файловых системах есть ошибки. Вы можете восстановить файловую систему с помощью:

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

Замените /dev/sdc на путь к узлу устройства вашего устройства. Вы можете найти узел устройства вашего устройства, которое всегда будет в /dev, проверив вывод:

```
mount
```

9.2.10 Почему calibre не поддерживает коллекции на Kindle или полку на Nook?

Ни Kindle, ни Nook не позволяют управлять коллекциями через USB-соединение. Если вы действительно заботитесь об использовании коллекций, я бы посоветовал вам продать свой Kindle/Nook и получить Kobo. Кажется, только Кобо понимает, что жизнь слишком коротка, чтобы вводить коллекции одну за другой на экране e-ink

Обратите внимание, что в случае с Kindle есть способ управлять коллекциями через USB, но для этого требуется, чтобы Kindle перезагружался *каждый раз* при отключении от компьютера, чтобы изменения в коллекциях были распознаны. Таким образом, маловероятно, что разработчики calibre когда-либо почувствуют себя достаточно мотивированными, чтобы поддержать его. Однако существует плагин calibre, который позволяет создавать коллекции на Kindle из метаданных calibre. Он доступен **`здесь https://www.mobileread.com/forums/showthread.php?t=244202`_**.

Примечание

Amazon полностью удалила возможность управления коллекциями в своих новых моделях, таких как Kindle Touch и Kindle Fire, что сделало даже вышеуказанный плагин бесполезным, если вы не рутируете свой Kindle и не устанавливаете пользовательскую прошивку.

9.2.11 Я получаю сообщение об ошибке, когда пытаюсь использовать calibre с моим Kobo Touch/Glo и т. д.?

У Kobo очень глючная прошивка. Известно, что подключение к нему происходит случайно. Определенные комбинации материнской платы, USB-портов/кабелей/концентраторов могут усугубить эту тенденцию к отказу. Если вы получаете сообщение об ошибке при подключении с помощью calibre, попробуйте следующее, каждое из которых решает проблему для *некоторых* пользователей calibre.

- Подключите Kobo напрямую к компьютеру, а не через USB-концентратор.
- Попробуйте использовать другой USB-кабель и другой USB-порт на компьютере.
- Выйдите из Kobo и войдите снова, это заставит его перестроить базу данных, исправив ошибки.
- Попробуйте обновить прошивку на Kobo Touch до последней версии
- Попробуйте перезагрузить Kobo (иногда это решает проблему на некоторое время, но затем она появляется снова, и в этом случае вам придется перезагружать его снова и снова)
- Старайтесь класть на Kobo только одну или две книги за раз и не храните большие коллекции на Kobo.

9.2.12 Обложки для книг, которые я отправляю на свой Kindle для электронных чернил, появляются на мгновение, а затем заменяются универсальной обложкой

Это происходит из-за ошибки Amazon. Они пытаются загрузить обложку для книги со своих серверов, и когда это не удается, они заменяют существующую обложку на созданную универсальную обложку. Подробности см. в этой ветке форума https://www.mobileread.com/forums/showthread.php?t=329945`_. Начиная с версии 4.17, у calibre есть обходной путь: если вы подключите Kindle к calibre после того, как Amazon уничтожит обложки, calibre восстановит их автоматически. Итак, чтобы увидеть обложки на Kindle, вам необходимо:

1) Отправьте книгу на Kindle из calibre
2) Отключите Kindle и подождите, пока Amazon уничтожит обложку.
3) Подключите Kindle к calibre.

Обратите внимание, что этот обходной путь работает только для книг, отправленных из calibre 4.17 или новее. В качестве альтернативы, просто оставьте Kindle в режиме полёта, если вы действительно не хотите, чтобы Amazon знал о каждой книге, которую вы читаете. Я рекомендую вам обратиться в службу поддержки Amazon и громко пожаловаться на эту ошибку. Может быть, Amazon прислушается.

Примечание

Если обходной путь у вас не работает, убедитесь, что у вас прошивка Kindle как минимум версии 5.12.5, выпущенной в апреле 2020 года.

9.2.13 Обложки для книг, отправленных на мой Kindle ColorSoft и новее, не отображаются в списке книг?

Amazon намеренно нарушила эту функциональность в своих устройствах ColorSoft и более новых версиях, чтобы отбить у вас охоту читать книги не от Amazon, на их устройствах. Подробности смотрите в „этой теме форума https://www.mobileread.com/forums/showthread.php?t=364350`__. Единственный известный обходной путь - отправить книги как „Личные документы“ на Kindle, что исправит отсутствие обложек, но нарушит работу других функций, таких как Whispersync, и книги будут отображаться на устройстве в разделе „Личные документы“, а не „Книги“. Чтобы включить это в calibre, перейдите по ссылке: guilabel: `Настройки-Параметры вывода-Вывод MOBI“ и установите флажок с надписью: guilabel: „Разрешить общий доступ к содержимому книги“. В результате все будущие книги, отправляемые на Kindle из calibre, будут помечены как личные документы.

9.2.14 Обложки моих MOBI-файлов перестали отображаться в Kindle для PC/Kindle, Android/iPad и т.п.

Причиной этому ошибка в программном обеспечении Amazon. Вы можете преодолеть эту проблему, перейдя в *Параметры → Настройки вывода → MOBI вывод* и изменить параметр «Включить обмен книжным контентом». Если вы переконвертируете преобразованную ранее книгу, вам также нужно включить параметр в диалоге конвертации для этой конкретной книги (так как настройки конвертации книги сохраняются и имеют преимущество).

Заметьте, что в таком случае созданный файл MOBI будет показыаться в списке персональных документоы вместо Книг на Kindle Fire и Amazon whispersync работать не будет, однако будут обложки. Вам решать, что для вас важнее. Я призываю вас связаться с Amazon и попросить их исправить эту ошибку.

Ошибка в программном обеспечении Amazon заключается в следующем. Когда вы передаёте файл MOBI в Kindle, если файл не отмечен как Персональный документ, Amazon предполагает, что вы купили книгу у них и пробует скачать миниатюру обложки с их серверов. Когда загрузка не удаётся, Amazon отказывается возвратиться к обложке, определённой в файле MOBI. Это похоже на преднамеренную попытку Amazon принудить

авторов продавать книги только через них. Другими словами, Kindle показывает обложки только для книг, отмеченных как Персональные документы или купленных напрямую у Amazon.

Если вы передаёте файл MOBI в Kindle на электронных чернилах через USB соединение при помощи calibre, то calibre преодолевает проблему, загружая миниатюру обложки самостоятельно. Однако, такое решение возможно только при использовании соединения USB и передачи через calibre. Заметьте, что если вы посылаете файл через электронную почту, Amazon автоматически отмечает файл MOBI как Персональный документ и обложка будет показана, но и книга будет в разделе Персональных документов.

9.2.15 Я перекинул несколько книг в мой Kindle при помощи calibre и их не видно?

Книги, отправленные на Kindle, отображаются на Kindle только после того, как они были *проиндексированы* Kindle. Это может занять некоторое время. Если книга по-прежнему не появляется через некоторое время, вероятно, произошел сбой индексатора Kindle. Иногда определённая книга может вызвать сбой индексатора. К сожалению, Amazon не предоставила никакого способа определить, какая книга вызывает сбой на Kindle. Единственный выход - либо перезагрузить Kindle, либо удалить все файлы из его памяти с помощью проводника Windows (или любого другого файлового менеджера), а затем снова отправить ему книги, одну за другой, пока вы не обнаружите проблемную книгу. Как только вы найдете проблемную книгу, удалите её с Kindle и выполните преобразование файла этой книги MOBI в MOBI или MOBI в AZW3 в calibre, а затем загрузите файл обратно в Kindle. Это, скорее всего, решит проблему.

9.3 Управление библиотекой

Содержание

- *Где хранятся файлы книг?* (страница 142)
- *Как calibre управляет именами авторов и сортировкой?* (страница 143)
- *Почему calibre не позволяет мне хранить книги в собственной структуре папок?* (страница 144)
- *Почему calibre не имет какого-либо необходимого мне столбца?* (страница 145)
- *Могу ли я добавить столбец с форматами или ISBN?* (страница 145)
- *Как перенести мою копию calibre с одного компьютера на другой?* (страница 145)
- *Список книг в calibre пуст!* (страница 146)
- *Я получаю ошибки, когда моя библиотека calibre находится на сетевом диске/NAS?* (страница 146)

9.3.1 Где хранятся файлы книг?

Когда вы впервые запускаете calibre, она попросит вас указать папку для хранения ваших книг. Каждый раз, когда вы добавляете книгу в calibre, она копирует книгу в эту папку. Книги в папке организованы в подпапки по автору и названию. Обратите внимание, что содержимое этой папки автоматически управляется calibre, **не** добавляйте файлы/папки вручную в эту папку, так как они могут быть автоматически удалены. Если вы хотите добавить файл, связанный с определённой книгой, используйте для этого правую верхнюю область диалогового окна *Редактировать метаданные*. Затем calibre автоматически поместит этот файл в нужную папку и переместит его при изменении названия/автора.

Метаданные о книгах хранятся в файле `metadata.db` на верхнем уровне папки библиотеки. Этот файл представляет собой базу данных sqlite. При резервном копировании вашей библиотеки убедитесь, что вы скопировали всю папку и все её подпапки.

Папка библиотеки и все её содержимое составляют так называемую библиотеку calibre. У вас может быть несколько таких библиотек. Для управления библиотеками щелкните значок calibre на панели инструментов. Вы можете создавать новые библиотеки, удалять/переименовывать существующие и легко переключаться между библиотеками.

Вы можете копировать или перемещать книги между разными библиотеками (если у вас настроено несколько библиотек), щелкнув книгу правой кнопкой мыши и выбрав действие *Копировать в библиотеку*.

9.3.2 Как calibre управляет именами авторов и сортировкой?

Имена авторов сложны, особенно в разных культурах, см. Это примечание[38], чтобы узнать о некоторых сложностях. calibre имеет очень гибкую стратегию управления именами авторов. Первое, что нужно понять, - это то, что книги и авторы - отдельные сущности по размеру. У книги может быть более одного автора, а у автора может быть более одной книги. Вы можете управлять авторами книги с помощью диалогового окна редактирования метаданных. Вы можете управлять отдельными авторами, щёлкнув правой кнопкой мыши автора в браузере тегов слева от главного окна калибра и выбрав: guilabel: *Управление авторами*. Используя этот диалог, вы можете изменить имя автора, а также способ сортировки этого имени. Имя автора автоматически изменится во всех книгах этого автора. Если у книги несколько авторов, разделите их имена с помощью символа &.

Теперь переходим к сортировке имён авторов:

- Когда в calibre добавляется новый автор (это происходит всякий раз, когда добавляется книга нового автора), calibre автоматически вычисляет строку сортировки как для книги, так и для автора.
- Авторы в Браузере тегов сортируются по значению сортировки для **авторов**. Помните, что это отличается от поля сортировки по авторам для книги.
- По умолчанию этот алгоритм сортировки предполагает, что имя автора находится в формате `Имя Фамилия` и генерирует значение сортировки `Фамилия, Имя`.
- Вы можете изменить этот алгоритм, перейдя в *Preferences-Advanced-Tweaks* и установив настройку *author_sort_copy_method*.
- Вы можете заставить calibre пересчитывать значения сортировки по авторам для каждого автора, щёлкнув правой кнопкой мыши по любому автору и выбрав *Управление авторами*, затем нажав кнопку *Пересчитать все значения сортировки по авторам*. Сделайте это после того, как вы установили желаемый твик author_sort_copy_method.
- Вы можете заставить calibre пересчитать значения сортировки по авторам для всех книг с помощью диалогового окна массового редактирования метаданных (выберите все книги и щёлкните Редактировать метаданные, установите флажок *Автоматически устанавливать сортировку по авторам*, затем нажмите OK.)
- При пересчете значений сортировки по авторам для книг calibre использует значения сортировки по авторам для каждого отдельного автора. Поэтому убедитесь, что значения сортировки отдельных авторов верны, прежде чем пересчитывать значения сортировки авторов книг.
- Вы можете управлять отображением в Браузере тегов авторов с использованием их имён или значений сортировки, установив настройку *Categories_use_field_for_author_name* в *Preferences-Advanced-Tweaks*

Обратите внимание, что вы можете установить любое значение сортировки для отдельного автора, используя *Manage authors*. Это полезно при работе с именами, которые calibre не может получить правильно, такими как сложные, состоящие из нескольких частей имена, такие как Miguel de Cervantes Saavedra, или при работе с азиатскими именами, такими как Sun Tzu.

При всей этой гибкости можно настроить calibre для управления именами авторов, как вам нравится. Например, один из распространенных запросов - указать имена авторов отображения калибра LN, FN. Для этого, и если приведенное ниже примечание к вам не относится, то:

- Установите `author_sort_copy_method` в `copy` как описано выше.

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

- Перезапустите calibre. Не меняйте никаких метаданных книги до выполнения оставшихся шагов.
- Измените все имена авторов на LN, FN, используя диалог Управление авторами.
- После того как вы изменили авторов, нажмите кнопку *Пересчитать все значения сортировки авторов*
- Нажмите OK, в этот момент calibre изменит авторов всех ваших книг. Это может занять какое-то время.

Примечание

При изменении с FN LN на LN, FN, часто бывает, что значения в author_sort уже в формате LN, FN. Если это ваш случай, то следайте следующее:

- Установите `author_sort_copy_method` в `copy` как описано выше.
- Перезапустите calibre. Не меняйте никаких метаданных книги до выполнения оставшихся шагов.
- Откройте диалог Управление авторами. Нажмите кнопку ``копировать все данные сортировки автору `
- Пройдитесь по авторам, чтобы быть уверенным, что всё в порядке. Вы всё ещё можете нажать Отмена, чтобы отменить изменения. Если вы нажали OK, то вернуть изменения будет невозможно.
- Нажмите OK, в этот момент calibre изменит авторов всех ваших книг. Это может занять какое-то время.

9.3.3 Почему calibre не позволяет мне хранить книги в собственной структуре папок?

Весь смысл функций управления библиотекой calibre заключается в том, что они предоставляют интерфейс на основе поиска и сортировки для поиска книг, который *намного* более эффективен, чем любая возможная схема папок, которую вы могли бы придумать для своей коллекции. Действительно, как только вы освоите интерфейс calibre для поиска, сортировки и просмотра своей коллекции, вы больше никогда не почувствуете необходимости рыться в файлах на вашем диске, чтобы снова найти книгу. Управляя книгами в собственной структуре папок, состоящей из файлов „Автор - Название - Книга“, calibre может достичь высокого уровня надежности и стандартизации. Чтобы проиллюстрировать, почему интерфейс на основе поиска / тегов превосходит папки, рассмотрим следующее. Предположим, ваша коллекция книг красиво отсортирована по папкам по следующей схеме:

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

Теперь это позволяет очень легко найти, например, все научно-фантастические книги Айзека Азимова из серии Foundation. Но предположим, что вы хотите найти все непрочитанные научно-фантастические книги. Нет простого способа сделать это с помощью этой схемы папок, вместо этого вам понадобится схема папок, которая выглядит следующим образом:

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

В calibre вместо этого вы должны использовать теги для отметки жанра и статуса чтения, а затем просто использовать простой поисковый запрос, например, `tag:scifi, and not tag:read`. calibre даже имеет красивый графический интерфейс, поэтому вам не нужно изучать его язык поиска, вместо этого вы можете просто щелкнуть теги, чтобы включить или исключить их из поиска.

Для тех из вас, кто утверждает, что вам нужен доступ к файловой системе, чтобы вы могли иметь доступ к своим книгам по сети, у calibre есть отличный контент-сервер, который дает вам доступ к библиотеке calibre через сеть.

Если вы беспокоитесь о том, что когда-нибудь calibre перестанет развиваться, оставив все ваши книги брошенными в своей структуре папок, изучите мощную функцию *Сохранить на диск* в calibre, которая позволяет вам экспортировать все ваши файлы в структуру папок произвольной сложности на основе их метаданных.

Наконец, причина, по которой в конце каждой папки заголовков есть числа, - это *надежность*. Этот номер является идентификационным номером книжной записи в базе данных calibre. Наличие номера позволяет иметь несколько записей с одинаковым названием и именами авторов. Это также часть того, что позволяет calibre волшебным образом регенерировать базу данных со всеми метаданными, если файл базы данных поврежден. Учитывая, что миссия calibre состоит в том, чтобы заставить вас перестать хранить метаданные в именах файлов и перестать использовать файловую систему для поиска вещей, повышенная надёжность, обеспечиваемая номерами идентификаторов, лучше уродливых имён папок.

Если вы всё ещё не уверены, то, боюсь, calibre не для вас. Поищите что-нибудь другое для каталогизации ваших книг. Просто чтобы было ясно, **это не изменится**. Пожалуйста, не пытайтесь заставить изменить это.

9.3.4 Почему calibre не имет какого-либо необходимого мне столбца?

В calibre предусмотрены столбцы для наиболее часто используемых и широко используемых полей. Кроме того, вы можете добавлять любые понравившиеся столбцы. Столбцы могут быть добавлены через *Preferences-Interface-Add your own columns*. Посмотрите учебное пособие UI Power tips[39], чтобы узнать, как создавать свои собственные столбцы, или прочтите это сообщение в блоге[40].

Вы также можете создать «виртуальные столбцы», которые содержат комбинации метаданных из других столбцов. В диалоге добавления столбцов используйте ссылки на *Быстрое создание* для лёгкого создания столбцов для отображения ISBN или формата книги. Вы можете использовать мощный язык шаблонов calibre, чтобы сделать намного больше столбцов. Для подробностей смотрите *Язык шаблонов calibre* (страница 164).

9.3.5 Могу ли я добавить столбец с форматами или ISBN?

Да, вы можете. Следуйте инструкциям в ответе выше для добавления особых столбцов.

9.3.6 Как перенести мою копию calibre с одного компьютера на другой?

Вы можете экспортировать все данные calibre (книги, настройки и плагины), а затем импортировать их на другой компьютер. Сначала посмотрим, как экспортировать данные:

- Щелкните правой кнопкой мыши значок calibre на главной панели инструментов calibre и выберите *Экспорт / импорт всех данных calibre*. Обратите внимание, что если в данный момент подключено устройство, этот пункт меню будет недоступен, поэтому отключите все подключенные устройства. Затем нажмите кнопку с надписью *Экспортировать все данные calibre*. Вы увидите список всех ваших библиотек calibre. Нажмите ОК и выберите пустую папку где-нибудь на вашем компьютере. Экспортированные данные будут сохранены в этой папке. Просто скопируйте эту папку на свой новый компьютер и следуйте инструкциям ниже, чтобы импортировать данные.
- Установите calibre на свой новый компьютер и запустите *Welcome wizard*, неважно, что вы там делаете, так как на следующем шаге вы импортируете свои старые настройки. Теперь у вас будет пустой calibre с только руководством *Getting Started* в библиотеке. Ещё раз щёлкните правой кнопкой мыши кнопку calibre и выберите *Экспорт / импорт всех данных calibre*. Затем нажмите кнопку с надписью :guilabel: *Импортировать ранее экспортированные данные*. Выберите папку с экспортированными данными, которые вы скопировали ранее. Теперь у вас есть список библиотек для импорта. Просмотрите список поочерёдно и выберите новое расположение для каждой библиотеки (это просто пустая папка где-то на вашем компьютере). Щелкните ОК. После завершения импорта calibre перезапустится со всеми вашими старыми библиотеками, настройками и плагинами calibre.

Примечание

[39] https://calibre-ebook.com/ru/demo#tutorials
[40] https://blog.calibre-ebook.com/calibre-custom-columns/

Эта функция импорта / экспорта доступна только начиная с версии 2.47 калибра. Если у вас более старая версия calibre или у вас возникли проблемы с импортом / экспортом, вы можете просто скопировать папку библиотеки calibre вручную, как описано в следующем абзаце.

Просто скопируйте папку библиотеки calibre со старого компьютера на новый. Вы можете узнать, что это за папка библиотеки, щелкнув значок calibre на панели инструментов. Выберите действие *Переключить / создать библиотеку calibre*, и вы увидите путь к текущей библиотеке calibre.

Теперь на новом компьютере запустите калибр в первый раз. Он запустит *Welcome wizard* и запросит у вас расположение библиотеки calibre. Наведите его на ранее скопированную папку. Если на целевом компьютере уже установлен calibre, то *Welcome wizard* не запустится. В этом случае щелкните правой кнопкой мыши значок calibre на панели инструментов и укажите на только что скопированную папку. Теперь у вас будет две библиотеки calibre на компьютере, и вы можете переключаться между ними, щелкая значок calibre на панели инструментов. При переносе библиотеки сохраняются все ваши метаданные, теги, настраиваемые столбцы и т. д.

9.3.7 Список книг в calibre пуст!

Чтобы понять, почему это произошло, вы должны понимать, что такое библиотека calibre. На самом базовом уровне библиотека calibre - это просто папка. Всякий раз, когда вы добавляете книгу в calibre, файлы этой книги копируются в эту папку (сгруппированные по подпапкам по автору и названию). Внутри папки библиотеки calibre на верхнем уровне вы увидите файл с именем metadata.db. В этом файле calibre хранит метаданные, такие как название/автор/рейтинг/теги, для *каждой* книги в вашей библиотеке calibre. Список книг, который отображает calibre, создается путём чтения содержимого этого файла metadata.db.

Существует две причины, по которым Calibre показывает пустой список книг:

- Папка вашей библиотеки calibre изменила свое местоположение. Это может произойти, если он был на внешнем диске и буква диска для этого диска изменилась. Или если вы случайно переместили папку. В этом случае calibre не может найти свою библиотеку и вместо этого запускается с пустой библиотекой. Для исправления щёлкните правой кнопкой мыши значок calibre на панели инструментов calibre и выберите «Переключить / создать библиотеку». Щёлкните по маленькому синему значку, чтобы выбрать новое расположение библиотеки calibre, и после выбора расположения нажмите OK. Если вы не знаете новое местоположение, найдите на своем компьютере файл `metadata.db`.
- Ваш файл metadata.db был удален / поврежден. В этом случае вы можете указать calibre восстановить файл metadata.db из его резервных копий. Щёлкните правой кнопкой мыши значок calibre на панели инструментов calibre и выберите «Обслуживание библиотеки - Восстановление базы данных». calibre автоматически перестроит файл metadata.db.

9.3.8 Я получаю ошибки, когда моя библиотека calibre находится на сетевом диске/NAS?

Не помещайте вашу библиотеку calibre на сетевой диск.

Файловая система - это сложный зверь. В большинстве сетевых файловых систем отсутствуют различные функции файловой системы, которые использует calibre. Некоторые не поддерживают блокировку файлов, некоторые не поддерживают жесткую привязку, некоторые просто нестабильны. Кроме того, calibre - это однопользовательское приложение, и если вы случайно запустите две копии calibre в одной сетевой библиотеке, произойдут неприятности. Наконец, разные операционные системы налагают разные ограничения на файловые системы, поэтому, если вы используете общий сетевой диск в разных операционных системах, опять же, *могут случиться неприятности*.

Рассмотрите возможность использования контент-сервера calibre, чтобы ваши книги были доступны на других компьютерах. Запустите calibre на одном компьютере и получите доступ к нему через контент-сервер или решение для удаленного рабочего стола.

Если вам необходимо предоставить общий доступ к реальной библиотеке, используйте инструмент синхронизации файлов, такой как DropBox или rsync, вместо сетевого диска. Если вы используете инструмент синхронизации файлов, **важно** убедиться, что и calibre, и инструмент синхронизации файлов не пытаются одновременно получить доступ к библиотеке calibre. Другими словами, **не** запускайте инструмент синхронизации файлов и calibre одновременно.

Даже при использовании этих инструментов существует опасность повреждения / потери данных, поэтому делайте это только в том случае, если вы готовы нести этот риск. В частности, имейте в виду, что **Google Диск** несовместим с calibre, если вы поместите свою библиотеку calibre в Google Диск, **вы потеряете данные**. Подробности смотрите в этой теме[41].

9.4 Разное

Содержание

- *Amazon прекращает доставку файлов MOBI по электронной почте?* (страница 148)
- *Я хочу, чтобы calibre загружал новостис моих любимых сайтов новостей.* (страница 148)
- *Почему программа называется calibre?* (страница 148)
- *Почему calibre показывает только некоторые из моих шрифтов в macOS?* (страница 149)
- *calibre не запускается в Windows?* (страница 149)
- *calibre время от времени тормозит/вылетает?* (страница 150)
- *Просмотрщик электронных книг calibre и инструменты редактирования книг не работают в Windows?* (страница 150)
- *Использование просмотрщика или выполнение любых конвертаций приводит к ошибке нехватки разрешений в Windows* (страница 151)
- *calibre не запускается/вылетает на macOS?* (страница 151)
- *Я получаю только чёрный или белый экран при запуске просмотрщика электронных книг Calibre?* (страница 151)
- *Я загрузил установщик, но он не работает?* (страница 151)
- *Мой антивирус утверждает, что calibre - вирус/троян?* (страница 152)
- *Как я могу сделать резервную копию calibre?* (страница 152)
- *Как я могу использовать купленные EPUB книги с calibre (или что делать с файлами .acsm)?* (страница 153)
- *Я получаю ошибку «Доступ запрещён»?* (страница 153)
- *Могу ли я показать метаданные комментария на моем ридере?* (страница 153)
- *Как я могу указать calibre использовать мой HTTP прокси?* (страница 154)
- *Я хочу, чтобы в calibre были добавлены некоторые возможности. Что я могу сделать?* (страница 154)
- *Почему у calibre нет автоматического обновления?* (страница 154)
- *Как лицензирована calibre?* (страница 155)
- *Как я могу запустить calibre с USB-флешки?* (страница 155)

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

- *Как мне запустить такие части calibre, как загрузка новостей и сервер содержимого, на моём собственном сервере Linux?* (страница 155)

9.4.1 Amazon прекращает доставку файлов MOBI по электронной почте?

Amazon «объявила https://blog.the-ebook-reader.com/2022/05/03/amazon-dropping-mobi-support-on-send-to-kindle-apps/__, *что прекратит принимать MOBI. файлы, отправленные по электронной почте на адреса электронной почты ``@kindle.com`*. Вы можете настроить calibre на отправку EPUB вместо MOBI, перейдя в *Настройки-Обмен книгами по электронной почте*, а затем удалив MOBI из списка форматов для отправки на ваш адрес электронной почты `@kindle.com` и добавив вместо этого EPUB.

Однако обратите внимание, что прием EPUB в Amazon очень несовершенен, они отклонят ряд файлов EPUB, которые работают везде. В таких случаях можно попробовать следующий трюк:

1. Конвертируйте файл EPUB в MOBI
2. Затем конвертируйте файл MOBI обратно в EPUB и отправьте полученный файл EPUB.

Это удалит всё расширенное форматирование, встроенные шрифты и т. д., но значительно повысит шансы на то, что Amazon примет EPUB.

Примечание

Если вы ранее использовали доставку по электронной почте периодических изданий, загруженных calibre, вам будет лучше отправлять их по USB-кабелю или загружать их с сервера контента калибровки через встроенный браузер Kindle. Однако, если вы хотите продолжить использовать доставку по электронной почте, вы можете попробовать изменить формат вывода в Preferences-Behavior на EPUB, тогда caliber будет загружать новости в формате EPUB. Примет ли Amazon EPUB или нет — это совсем другой вопрос.

9.4.2 Я хочу, чтобы calibre загружал новостис моих любимых сайтов новостей.

Если вы достаточно опытный с компьютерами, вы можете научить calibre скачивать новости с любого сайта по вашему выбору. Чтобы узнать, как это сделать смотрите news`.

В противном случае вы можете запросить конкретный новостной сайт, разместив сообщение на форуме calibre Recipes[42].

9.4.3 Почему программа называется calibre?

Выбирайте:

- Converter And LIBRary for E-books
- Продукт высокого качества (*calibre*)
- Дань SONY Librie, первой читалке электронных книг с экраном на основе электронных чернил
- Моя жена выбрала название ;-)

calibre произносится как кал-и-бер, не ка-ли-бре. Если вы удивлены, калибре - произношение слова caliber в Британии/Содружестве. Как для индийца, это для меня естественное произношение.

[42] https://www.mobileread.com/forums/forumdisplay.php?f=228

9.4.4 Почему calibre показывает только некоторые из моих шрифтов в macOS?

calibre встраивает шрифты в файлы электронных книг, которые он создает. Файлы электронных книг поддерживают встраивание только шрифтов TrueType и OpenType (.ttf и .otf). Большинство шрифтов в системах MacOS имеют формат .dfont, поэтому они не могут быть встроены. calibre показывает только шрифты TrueType и OpenType, имеющиеся в вашей системе. Вы можете найти множество таких шрифтов в Интернете. Просто загрузите файлы .ttf / .otf и добавьте их в папку Library/Fonts в своей домашней папке.

9.4.5 calibre не запускается в Windows?

Этому может быть несколько причин:

- Если вы не получаете ошибок, но окно calibre не появляется, возможно, оно появилось за пределами экрана. Вы можете собрать все окна на текущем экране, используя один из методов, описанных здесь[43].
- Известно, что некоторое программное обеспечение вмешивается в работу calibre, попробуйте перезагрузиться в безопасном режиме и посмотреть, работает ли оно. Известным виновником является Sunshine[44] программное обеспечение для совместного использования экрана.
- Если вы видите ошибку о том, что calibre не может открыть файл, потому что он используется другой программой, сделайте следующее:
  - Удалите программу calibre
  - Перезагрузите ваш компьютер.
  - Переустановите calibre , но не запускайте calibre из мастера установки.
  - Временно отключите ваш антивирус (для безопасности перед этим отключитесь от интернета)
  - Загляните в директорую, которую вы выбрали для библиотеки calibre . Если вы видите файл с названием metadata.db, удалите его.
  - Запустите calibre
  - С этого моменты calibre должна запускаться нормально.
- Если вы видите ошибку о том, что функция Python неожиданно завершилась после обновления calibre, сначала удалите calibre, потом удалите директории (если они существуют):file:*C:\Program Files\Calibre* and `C:\Program Files\Calibre2` . Теперь переустановите calibre и всё должно быть хорошо.
- Если вы получаете ошибку в *Welcome wizard* при первом запуске calibre, попробуйте выбрать папку, например `C:\library`, в качестве библиотеки calibre (у calibre иногда есть проблемы с расположением библиотек, если путь содержит неанглийские символы или только цифры и т. д.)
- Запустите его от имени Администратора (Щёлкните правой кнопкой мыши по значку и выберите *Запустить как Администратор*)

Если он по-прежнему не запускается, запустите командную строку (нажмите клавишу Windows и R; затем введите: command: *cmd.exe* в появившемся диалоговом окне *Выполнить*). В командной строке введите следующую команду и нажмите Enter:

```
calibre-debug -g
```

Публикуйте любые результаты, которые вы видите в справочном сообщении, на Форуме[45].

[43] https://www.wikihow.com/Bring-an-Off-Screen-Window-Back-on-Windows
[44] https://github.com/LizardByte/Sunshine
[45] https://www.mobileread.com/forums/forumdisplay.php?f=166

9.4.6 calibre время от времени тормозит/вылетает?

Я знаю несколько вещей, которые могут быть этому причиной:

- Недавно вы подключили внешний монитор или телевизор к компьютеру. В этом случае, при открытии calibre новых окон, например, окна редактирования метаданных или диалогового окна преобразования, они появляются на втором экране, где вы их не замечаете и думаете, что calibre завис. Отключите второй монитор и перезапустите calibre.
- Сообщалось, что следующие программы вызывают сбои в calibre: Если вы используете любую из них, закройте их перед запуском calibre или удалите их: *RoboForm*, *Logitech SetPoint Settings*, *Constant Guard Protection от Xfinity*, *Spybot*, *Killer Network Manager*, *Интерфейс пользовательского интерфейса Nahimic*, *Acronis True Image*.
- Вы используете USB мышь/планшет Wacom. Присутствует несовместимость между драйверами Wacom и графическим инструментарием, используемым calibre. Попробуйте использовать мышь не от Wacom.
- В некоторых 64-разрядных версиях Windows есть программное обеспечение/параметры безопасности, которые препятствуют нормальной работе 64-разрядного calibre. Если вы используете 64-разрядную версию calibre, попробуйте перейти на 32-разрядную версию.
- Если сбой происходит, когда вы пытаетесь скопировать текст из программы просмотра электронных книг calibre, скорее всего, это вызвано каким-то запущенным приложением для мониторинга/управления буфером обмена. Выключите его, и всё будет в порядке.
- Если сбои происходят именно тогда, когда вы используете диалоговое окно с файлом, например, нажимаете кнопку *Добавить книги* или *Сохранить на диск*, значит, у вас есть программное обеспечение, которое установило неработающие расширения оболочки на вашем компьютере. компьютер. Известные виновники: *SpiderOak*, *odrive sync* и *Dell Backup and Recovery* и *NetDrive*. Если у вас есть один из них, удалите его, и всё будет в порядке. Вы также можете использовать NirSoft Shell Extension Viewer[46], чтобы узнать, какие расширения оболочки установлены в вашей системе, и отключить их по отдельности, если вы не хотите удалять программу. Не забудьте перезагрузить проводник или перезагрузить весь компьютер после отключения расширений оболочки.

Если ничего из вышеперечисленного не относится к вам, значит, на вашем компьютере есть какая-то другая программа, которая мешает calibre. Сначала перезагрузите компьютер в безопасном режиме, чтобы у вас было как можно меньше запущенных программ, и посмотрите, происходят ли сбои по-прежнему. Если этого не происходит, то вероятно проблема вызвана какой-то программой. Наиболее вероятным таким виновником является программа, которая изменяет поведение других программ, например антивирус, драйвер устройства, что-то вроде RoboForm (приложение для автоматического заполнения форм) или вспомогательные технологии, такие как голосовое управление или средство чтения с экрана.

Единственный способ найти виновника - удалить программы одну за другой и посмотреть, какая из них вызывает проблему. Вкратце: остановите программу, запустите calibre, проверьте, нет ли сбоев. Если они всё ещё возникают, остановите другую программу и повторите.

9.4.7 Просмотрщик электронных книг calibre и инструменты редактирования книг не работают в Windows?

Эти две программы используют аппаратное ускорение, поскольку они встраивают версию браузера Chrome для рендеринга HTML. Если они не работают, это будет из-за несовместимости с драйверами графического процессора вашей системы. Попробуйте сначала обновить их и перезагрузитесь. Если это не поможет, вы можете установить для переменной среды `QTWEBENGINE_CHROMIUM_FLAGS` значение `--disable-gpu`, чтобы отключить аппаратное ускорение. Подробности смотрите на этой странице[47].

[46] https://www.nirsoft.net/utils/shexview.html
[47] https://doc.qt.io/qt-6/qtwebengine-debugging.html

9.4.8 Использование просмотрщика или выполнение любых конвертаций приводит к ошибке нехватки разрешений в Windows

Что-то на вашем компьютере мешает calibre получить доступ к собственным временным файлам. Скорее всего, права доступа к вашей папке `Temp` неверны. Перейдите в папку `C:\Users\USERNAME\AppData\Local` в проводнике Windows, а затем щелкните правой кнопкой мыши папку`Temp`, выберите *Properties* и перейдите во вкладку *Безопасность*. Убедитесь, что ваша учётная запись пользователя имеет полный контроль над этой папкой.

У некоторых пользователей выполнение следующей команды в командной строке с правами Администратора исправило разрешения. Для запуска командной строки от Администратора, найдите cmd.exe в меню «Пуск», затем щёлкните правой кнопкой мыши по найденному пункту и выберите «Запуск от имени администратора». В командной строке введите следующую команду и нажмите Enter:

```
icacls "%appdata%\..\Local\Temp" /reset /T
```

Или вы можете запустить calibre от имени Администратора, но при этом некоторые функции, такие как перетаскивание мышью, перестанут работать.

В итоге, некоторые пользователи сообщили, что отключение UAC решает проблему.

9.4.9 calibre не запускается/вылетает на macOS?

Одной из распространенных причин сбоев в macOS является использование технологий специальных возможностей, несовместимых с графическим инструментарием, используемым calibre. Попробуйте отключить VoiceOver, если он у вас включен. Также перейдите в Системные настройки-Система-Универсальный доступ и отключите настройку для включения доступа для вспомогательных устройств на всех вкладках. Другой причиной могут быть некоторые сторонние приложения, которые изменяют поведение системы, такие как Smart Scroll и Default Folder X.

Вы можете получить отладочную информацию о том, почему calibre не запускается, запустив *Console.app*. В нём будет напечатан вывод отладки. Если выходные данные отладки содержат строку, которая выглядит как:

```
Qt: internal: -108: Error ATSUMeasureTextImage text/qfontengine_mac.mm
```

тогда проблема, вероятно, в поврежденном кэше шрифтов. Вы можете очистить кэш, следуя этим инструкциям[48]. Если это не решит проблему, поищите повреждённый файл шрифта в вашей системе в ~/Library/Fonts или подобном месте. Простой способ проверить наличие поврежденных шрифтов в macOS - запустить приложение «Font Book» , выбрать все шрифты, а затем в меню File выбрать «Validate fonts».

9.4.10 Я получаю только чёрный или белый экран при запуске просмотрщика электронных книг Calibre?

Это происходит из-за несовместимости между Qt WebEngine, который средство просмотра использует для рендеринга, и драйверами графического процессора в вашей системе. Сначала попробуйте обновить драйверы графического процессора. Если это не поможет, вы можете попробовать отключить аппаратное ускорение в Qt WebEngine, установив для переменной среды `QTWEBENGINE_CHROMIUM_FLAGS` значение `--disable-gpu`. См. *Переменные окружения* (страница 298), чтобы узнать, как изменить переменные среды.

9.4.11 Я загрузил установщик, но он не работает?

При скачивании из Интернета загруженные файлы иногда могут оказаться поврежденными. Если скачанный установочный файл calibre не запускается, попробуйте скачать его снова. Если это не помогает, скачайте его из другого расположения[49]. Если установщик по-прежнему не работает, значит что-то на вашем компьютере мешает его запуску.

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

- Попробуйте временно отключить вашу антивирусную программу (Microsoft Security Essentials, Антивирус Касперского, Norton, McAfee и проч.). Скорее всего, она и является причиной зависания процесса обновления на полпути.
- Точно так же, если установщик дает сбой/запускается и у вас работает Microsoft PowerToys, выйдите из него.
- Если вы установили программу в нестандартное расположение, попробуйте запустить программу установки от имени администратора
- Попробуйте перезагрузить компьютер и запустить программу очистки реестра, например Wise registry cleaner[50].
- Попробуйте чистую установку. Деинсталлируйте calibre, удалите каталог `C:\Program Files\Calibre2` (или тот, который вы указывали при установке). Переустановите calibre. Имейте в виду, что деинсталляция не затронет ваши книги или настройки.
- Попробуйте загрузить установщик в другом браузере. Например, если вы используете Microsoft Edge, попробуйте вместо этого использовать Firefox или Chrome.
- Если вы получаете сообщение об отсутствующей DLL в Windows, то, скорее всего, права доступа к вашей временной папке неверны. Перейдите в папку `C:\Users\USERNAME\AppData\Local` в проводнике Windows, а затем щелкните правой кнопкой мыши по папке `Temp` и выберите *Свойства* и перейдите в вкладку *Безопасность*. Убедитесь, что у вашей учётной записи пользователя есть полные права на эту папку.

Если вы по-прежнему не можете заставить работать установщик и работаете в Windows, вы можете использовать calibre portable install[51], для которого не требуется установщик (это просто ZIP файл).

9.4.12 Мой антивирус утверждает, что calibre - вирус/троян?

Первое, что нужно проверить, это то, что вы загружаете calibre с официального сайта: https://calibre-ebook.com/ru/<download>. Убедитесь, что вы нажимаете ссылки для скачивания слева, а не рекламу справа. calibre - очень популярная программа, и недобросовестные люди создают фальшивые веб-сайты, якобы предлагающие её для загрузки, с целью обмана неосторожных посетителей.

Если у вас есть официальная загрузка, а ваша антивирусная программа всё ещё утверждает, что calibre - это вирус, значит, ваша антивирусная программа ошибается. Антивирусные программы используют эвристику, шаблоны кода, которые «выглядят подозрительно» для обнаружения вирусов. Это скорее похоже на расовое профилирование. calibre - это продукт с полностью открытым исходным кодом. Вы можете самостоятельно просмотреть исходный код (или нанять кого-нибудь, кто сделает это за вас), чтобы убедиться, что это не вирус. Сообщите о ложной идентификации любой компании, у которой вы покупаете антивирусное программное обеспечение. Если антивирусная программа не позволяет вам загрузить/установить calibre, временно отключите её, установите calibre и снова включите.

9.4.13 Как я могу сделать резервную копию calibre?

Самая важная вещь для резервного копирования - это папка библиотеки calibre, которая содержит все ваши книги и метаданные. Это папка, которую вы выбрали для своей библиотеки calibre при первом запуске calibre. Вы можете получить путь к папке библиотеки, щелкнув значок calibre на главной панели инструментов. Вы должны сделать полную резервную копию этой папки со всеми её файлами и подпапками.

Вы можете переключить calibre на использование резервной папки библиотеки, просто щелкнув значок calibre на панели инструментов и выбрав папку резервной библиотеки. Резервная копия библиотеки папок создает резервные копии ваших пользовательских столбцов и сохраненных результатов поиска, а также всех ваших книг и метаданных.

[50] https://www.wisecleaner.com
[51] https://calibre-ebook.com/ru/download_portable

Если вы хотите сделать резервную копию конфигурации/плагинов calibre, вам необходимо сделать резервную копию папки config. Вы можете найти эту конфигурационную папку через *Preferences-Miscellaneous*. Обратите внимание, что восстановление папок конфигурации официально не поддерживается, но в большинстве случаев должно работать. Просто скопируйте содержимое резервной папки в текущую папку конфигурации для восстановления.

9.4.14 Как я могу использовать купленные EPUB книги с calibre (или что делать с файлами .acsm)?

В большинстве купленных EPUB-книг имеется *DRM* (страница 387), что не даёт calibre их открыть. Хотя вы всё ещё можете использовать calibre для их хранения и передачи на своё устройство чтения. Для начала нужно авторизовать устройство на компьютере с Windows при помощи Adobe Digital Editions. Как только это будет сделано, EPUB-книги, переданные calibre, будут отлично работать на вашей читалке. Когда вы покупаете EPUB-книгу на веб-сайте, то получаете «.acsm» файл. Его необходимо открыть в Adobe Digital Editions, и программа загрузит актуальную «.epub» книгу. Файл электронной книги будет сохранён в папке «My Digital Editions», откуда вы сможете добавить его в calibre.

9.4.15 Я получаю ошибку «Доступ запрещён»?

Ошибки доступа могут появляться по ряду причин, никаким образом не связанных с calibre.

- Ошибки доступа могут появляться по причине включенной защиты от записи на SD карте.
- В macOS, если вы получаете ошибки разрешений при подключении устройства к calibre, можно исправить это: guilabel: *Системные настройки -> безопасность и конфиденциальность -> Файлы и папки*.
- Если вы или какая-либо используемая вами программа изменили разрешения проблемных файлов на «только чтение».
- Если на устройстве произошла ошибка файловой системы, из-за которой ваша операционная система смонтировала файловую систему в режиме только для чтения или пометила конкретный файл атрибутом «только для чтения» ожидая его восстановления.
- Если владельцами файлов выставлен другой пользователь.
- Если ваш файл открыт в другой программе.
- Если файл находится на устройстве, возможно, вы достигли ограничения в 256 файлов в корневом каталоге устройства. В этом случае вам необходимо переформатировать устройство/SD-карту, указанную в сообщении об ошибке, в файловую систему FAT32 или удалить некоторые файлы с SD-карты/памяти устройства.

Вам нужно будет устранить основную причину ошибки разрешений, прежде чем возобновить использование calibre. Внимательно прочтите сообщение об ошибке, посмотрите, на какой файл оно указывает, и исправьте разрешения для этого файла или содержащих его папок.

9.4.16 Могу ли я показать метаданные комментария на моем ридере?

Большинство читателей этого не поддерживают. Вы должны пожаловаться на это производителю, и, надеюсь, если достаточно людей будет жаловаться, все изменится. Тем временем вы можете вставить метаданные, включая комментарии, на «страницу обложки» в начале электронной книги, используя опцию «Вставить метаданные как страницу в начале книги» во время преобразования. Опция находится в разделе *Определение структуры* настроек преобразования. Обратите внимание, что для этого вам необходимо *конвертировать* книгу. Если ваша книга уже имеет формат, который не требует преобразования, вы можете преобразовать этот формат в тот же формат.

Другой альтернативой является создание каталога в форме электронной книги, содержащего список всех книг в вашей библиотеке calibre с их метаданными. Нажмите и удерживайте кнопку *Convert*, чтобы получить доступ к инструменту создания каталога. И прежде чем вы спросите, нет, у вас не может быть каталог с «прямой ссылкой на книги» на вашем читателе.

9.4.17 Как я могу указать calibre использовать мой HTTP прокси?

По умолчанию calibre использует любые настройки прокси, установленные в вашей ОС. Иногда они неверны, например, в Windows, если вы не используете Microsoft Edge, настройки прокси-сервера могут быть устаревшими. Вы можете указать calibre использовать конкретный прокси-сервер, установив переменные среды http_proxy и https_proxy. Формат переменной: `http://username:password@servername`, вы должны попросить вашего сетевого администратора дать вам правильное значение для этой переменной. Обратите внимание, что calibre поддерживает только HTTP-прокси, но не SOCKS-прокси. Вы можете увидеть текущие прокси, используемые calibre, в Настройках-Разное.

9.4.18 Я хочу, чтобы в calibre были добавлены некоторые возможности. Что я могу сделать?

У вас два выбора:

1. Создать патч путём изменения calibre и послать его мне для рассмотрения и включения. См. Разработка[52].
2. Откройте ошибку, запрашивающую функцию[53]. Помните, что, хотя вы можете думать, что ваш запрос функции чрезвычайно важен / необходим, разработчики calibre могут не согласиться. К счастью, calibre имеет открытый исходный код, что означает, что у вас всегда есть возможность реализовать свою функцию самостоятельно или нанять кого-то, кто сделает это за вас. Кроме того, calibre имеет комплексную архитектуру плагинов, поэтому вы можете разработать свою функцию как плагин, см. *Создание собственных плагинов для расширения функциональности calibre* (страница 232).

9.4.19 Почему у calibre нет автоматического обновления?

По многим причинам:

- *Нет необходимости в еженедельном обновлении*. Если вас устраивает работа calibre, отключите уведомление об обновлении и будьте сами по себе. Возвращайте уведомления раз в год или около того, чтобы посмотреть, не захотите ли обновиться. Чекбокс для отключения уведомлений об обновлении есть в самом уведомлении.
- Скачивания calibre в настоящее время используют около 150 ТБ полосы пропускания в месяц <dynamic/downloads>`. Внедрение автоматических обновлений значительно увеличит это и в конечном итоге обойдется в тысячи долларов в месяц, которые кому-то придется платить.
- Если я реализую диалог, которые скачивает и запускает обновление, вместо перехода на веб-сайт как сейчас, это могло бы сохранить наиболее пылким любителям обновлений calibre *не более пяти кликов в неделю*. В разработке calibre есть намного более приоритетные вещи.
- Если вы действительно очень ненавидите загружать calibre, но всё ещё хотите использовать последнюю версию, я поддерживаю запуск из исходного кода, что делает обновление тривиальным. Инструкции *доступны здесь* (страница 355).
- На форуме calibre <https://www.mobileread.com/forums/forumdisplay.php?f=238>`_ имеются сторонние средства автоматического обновления для calibre, созданные пользователями calibre.
- Кроме того, некоторые программы обновления сторонних производителей, такие как Norton или Avast, могут обновлять программное обеспечение за спиной пользователя. Если вы обнаружите, что calibre неожиданно обновился, проверьте их наличие.

[52] https://calibre-ebook.com/ru/get-involved
[53] https://calibre-ebook.com/ru/bugs

9.4.20 Как лицензирована calibre?

calibre под лицензией GNU General Public License v3 (лицензия с открытым исходным кодом). Это означает, что вы можете свободно распространять calibre до тех пор, пока вы делаете исходный код доступным. Поэтому, если вы хотите поместить calibre на компакт-диск с вашим продуктом, вы также должны поместить исходный код calibre на компакт-диск. Исходный код доступен для скачивания на[54]. Вы можете использовать результаты конвертации из calibre, как хотите. Вы не можете использовать ни код, ни библиотеки от calibre в своем программном обеспечении, не сделав свое программное обеспечение открытым. Подробнее см. GNU GPL v3[55].

9.4.21 Как я могу запустить calibre с USB-флешки?

Портативная версия calibre доступна здесь[56].

9.4.22 Как мне запустить такие части calibre, как загрузка новостей и сервер содержимого, на моём собственном сервере Linux?

Во-первых, вы должны установить calibre на свой Linux-сервер. Если на вашем сервере используется современный дистрибутив Linux, у вас не должно возникнуть проблем с установкой на него calibre.

Примечание

calibre требуется GLIBC >= 2.31 и libstdc++ >= 6.0.28. Если у вас более старый сервер, вам нужно либо скомпилировать их из исходного кода, либо использовать calibre 3.48, для которого требуется GLIBC >= 2.17, или 2.85.1, для которого требуется GLIBC >= 2.13, или calibre 1.48, для которого требуется только GLIBC >= 2.10. Кроме того, хотя утилитам командной строки calibre не требуется работающий X-сервер, некоторые из них требуют, чтобы в вашей системе были установлены библиотеки X-сервера. Это из-за Qt, который используется для различных задач обработки изображений, и ссылок на эти библиотеки. Если вы получаете сообщение ImportError о некоторых модулях Qt, вероятно не хватает некоторых X-библиотек. Типичными кандидатами являются: `libxcb-cursor0`, `libxcb-xinerama0`, `libegl1`, `libopengl0`.

Вы можете запустить сервер calibre при помощи команды:

```
/opt/calibre/calibre-server /path/to/the/library/you/want/to/share
```

Вы можете скачать новости и преобразовать их в электронную книгу с помощью команды:

```
/opt/calibre/ebook-convert "Title of news source.recipe" outputfile.epub
```

Если вы хотите сгенерировать MOBI, используйте `outputfile.mobi` вместо этого и используйте `--output-profile kindle`.

Вы можете послать загруженные новости по электронной почте командой:

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

Я оставляю создание точной командной строки в качестве упражнения для читателя.

В результате вы можете добавить загруженные новости в библиотеку calibre при помощи:

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

[54] https://download.calibre-ebook.com
[55] https://www.gnu.org/licenses/gpl.html
[56] https://calibre-ebook.com/ru/download_portable

Не забудьте прочитать раздел *Командная строка* (страница 313) в руководстве пользователя calibre, чтобы узнать больше об этих и других командах.

ГЛАВА 10

Руководства

Здесь вы найдёте руководства по более сложным функциям, например, XPath или шаблонам.

10.1 Управление подгруппами книг, например «genre»

Некоторые люди хотят организовать книги в своей библиотеке в подгруппы, похожие на подпапки. Наиболее распространенной причиной является создание жанровой иерархии, но есть и многие другие. Один пользователь попросил способ организовать учебники по предмету и номеру курса. Другой хотел отслеживать подарки по предмету и получателю. Этот учебник будет использовать жанр как пример для остальной части этого сообщения.

Прежде чем продолжить, обратите внимание, что мы не говорим о папках на жестком диске. Подгруппы не являются файловыми папками. Книги не будут никуда копироваться. Это не влияет на файловую структуру библиотеки. Вместо этого мы представляем способ организации и отображения подгрупп книг в библиотеке.

- *Установка* (страница 159)
- *Поиск* (страница 161)
- *Ограничения* (страница 161)
- *Полезные шаблонные функции* (страница 162)

Обычно для подгрупп, таких как жанры, требуются:

- Подгруппа (например, жанр) должна содержать (указывать) книги, а не категории книг. Это то, что отличает подгруппы от пользовательских категорий calibre.
- Книга может быть в нескольких подгруппах (жанрах). Это отличает подгруппы от физических файловых папок.
- Подгруппы (жанры) должны образовывать иерархию; подгруппы могут содержать подгруппы.

Теги дают вам первые два. Если вы тегируете книгу жанром, вы можете использовать браузер тегов (или поиск) для поиска книг с этим жанром, с выдачей первого. Многие книги могут иметь один и тот же тег (-и), предоставляя

вам второй. Проблема в том, что теги не удовлетворяют третьему требованию. Они не обеспечивают иерархию.

Функция иерархии в calibre дает вам третье - способность видеть жанры в «дереве» и возможность легко искать книги в жанре или поджанре. Например, предположим, что ваша структура жанра похожа на следующую:

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

Используя функцию иерархии, вы можете увидеть эти жанры в браузере тегов в виде дерева, как показано на изображении на экране. В этом примере внешний уровень (жанр) является настраиваемым столбцом, который содержит жанры. Жанры, содержащие поджанры, появляются с небольшим треугольником рядом с ними. Нажатие на этот треугольник откроет элемент и покажет суб-жанры, как вы можете видеть в Истории и Научной фантастике.

Нажав на жанр, вы можете искать все книги с этим жанром или потомками этого жанра. Например, нажатие на Science Fiction может дать все три дочерних жанра: Alternate History, Military и Space Opera. Нажатие на Alternate History выдаст книги в этом жанре, игнорируя те, которые находятся в Military и Space Opera . Конечно, книга может иметь несколько жанров. Если в книге есть жанры как Space Opera, так и Military, вы увидите эту книгу,

если вы нажмете на любой жанр. Подробнее поиск обсуждается ниже.

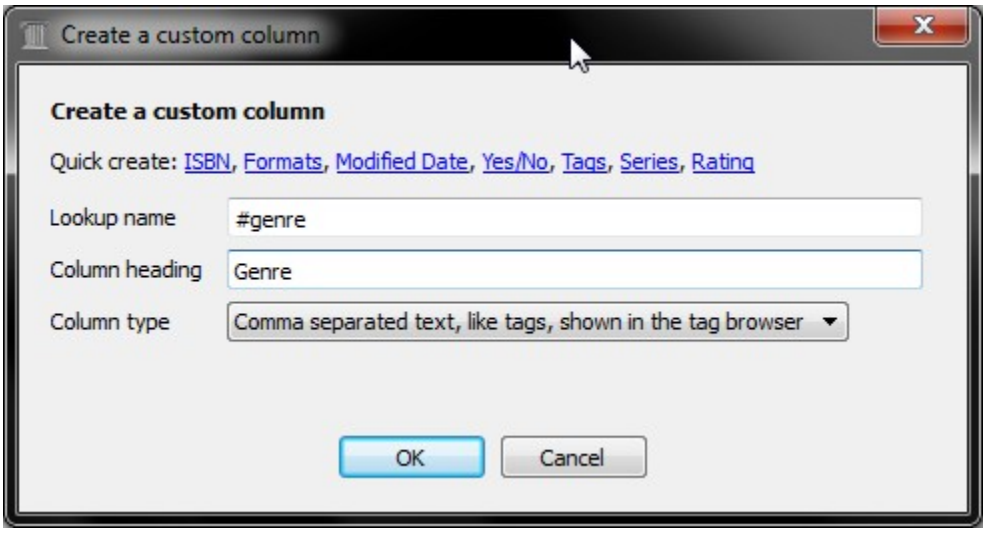
Ещё одна вещь, которую вы можете видеть по изображению, состоит в том, что жанр Military появляется дважды, один раз в Истории и один раз под Научной фантастикой. Поскольку жанры находятся в иерархии, это два отдельных жанра. Книга может быть в одном, другом или (сомнительно в этом случае) и в том, и другом. Например, книга Уинстона Черчилля «Вторая мировая война» могут быть в «History.Military». Книги почёта Хэрингтона Дэвида Вебера можно найти в «Science Fiction.Military», и в этом отношении также в «Science Fiction.Space Opera».

Если жанр существует, это значит, что, по крайней мере, одна книга имеет этот жанр. Вы можете легко применить его к другим книгам, перетащив книги из представления библиотеки в желаемый для них жанр. Вы также можете применять жанры в редакторах метаданных; подробнее об этом ниже.

10.1.1 Установка

К настоящему моменту ваш вопрос может быть следующим: «Как всё это настроить?» Всего три шага: 1) создать пользовательский столбец, 2) сообщить calibre, что новый столбец должен рассматриваться как иерархия, и 3) добавить жанры.

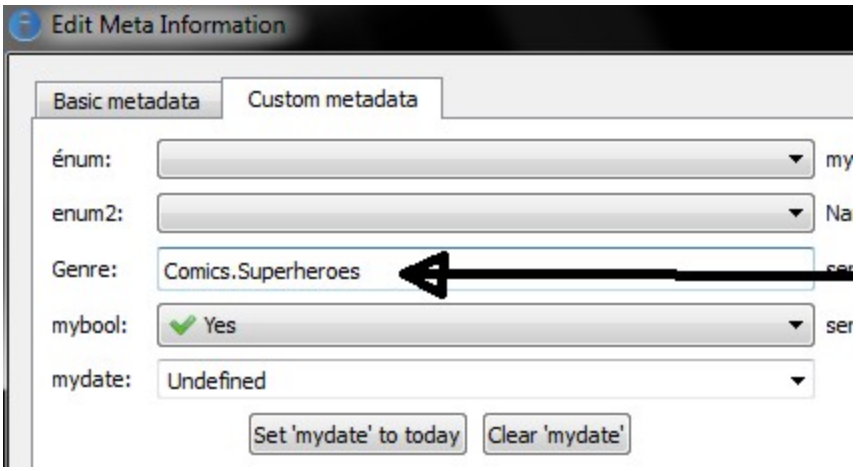
Вы создаете пользовательский столбец обычным способом, используя «Настройки»->«Добавить свои собственные столбцы». Этот пример использует «#genre» в качестве имени поиска и «Genre» в качестве заголовка столбца. Тип столбца - это *Текст с разделителями-запятыми, напр. теги, отображаемые в Браузере тегов*.



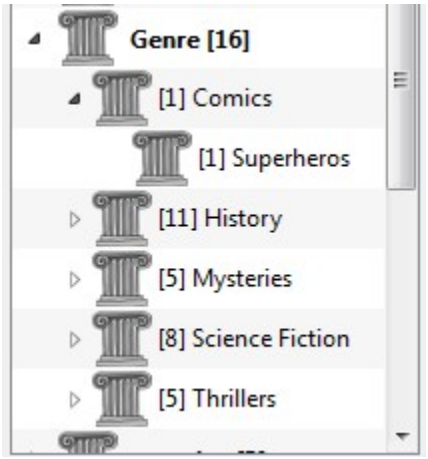
Затем, после рестарта, вы должны сказать calibre, что столбец должен рассматриваться как иерархия. Перейдите по ссылке *Настройки → Look & feel → Tag browser → Иерархия и поиск*» и выберите новый столбец Жанр как содержащий иерархические элементы.

На данный момент в столбце нет жанров. Остаётся последний шаг: как применить жанр к книге. Жанр не существует в calibre, пока он не появится, по крайней мере, в одной книге. Чтобы узнать, как применить жанр в первый раз, мы должны подробно рассказать о том, как выглядит жанр в метаданных для книги.

Иерархия «вещей» строится путем создания элемента, состоящего из фраз, разделённых точками. Продолжая пример жанра, эти предметы будут «History.Military», «Mysteries.Vampire», «Science Fiction.Space Opera» и т.д. Таким образом, чтобы создать новый жанр, вы выбираете книгу, которая должна иметь этот жанр, редактируете его метаданные и вводите новый жанр в созданный вами столбец. Продолжая наш пример, если вы хотите присвоить книге новый жанр «Комиксы» с поджанром «Супергерои», вы должны «редактировать метаданные» для этой книги (комикса), выбрать вкладку «Пользовательские метаданные», а затем ввести «Comics.Superheroes», как показано ниже (игнорируйте другие настраиваемые столбцы):



После выполнения вышеизложенного вы увидите в Браузере тегов:

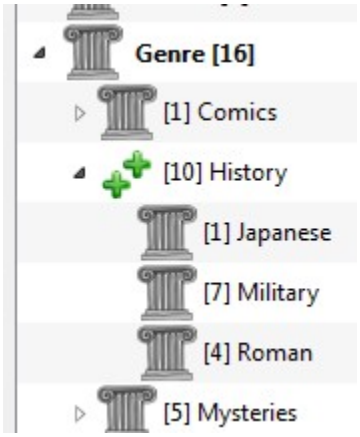


С этого момента, чтобы применить этот новый жанр к книге (предположительно, комикс), вы можете перетащить книгу в жанр или добавить его в книгу, используя редактирование метаданных, точно так же, как это было сделано выше.

Примечание

Иерархическое отображение работает только в том случае, если браузер тегов настроен на сортировку элементов по имени. Это значение по умолчанию, и его можно проверить, нажав кнопку *Configure* в нижней части браузера тегов.

10.1.2 Поиск

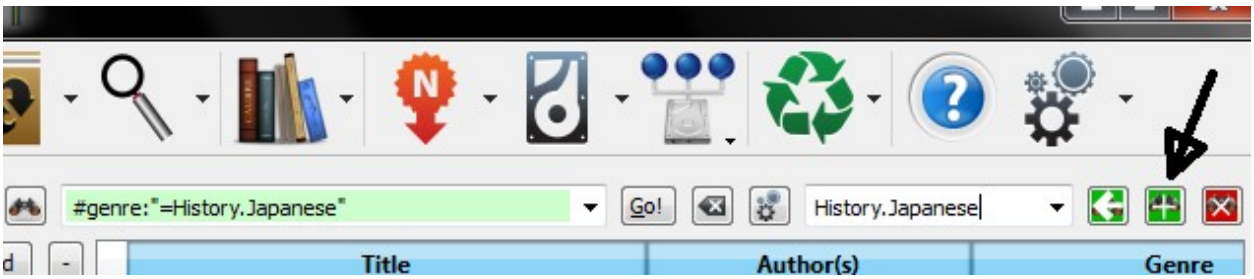


Простейший способ поиска жанров - использовать Браузер тегов, нажав на жанр, который вы хотите увидеть. Нажав на жанр с потомками, вы увидите книги с этим жанром и всеми жанрами-потомками. Однако это может вызвать вопрос. Просто потому, что у жанра есть потомки, это не значит, что это не собственный жанр. Например, книга может иметь жанр «History», но не «History.Military». Как вы ищете книги только с «History»?

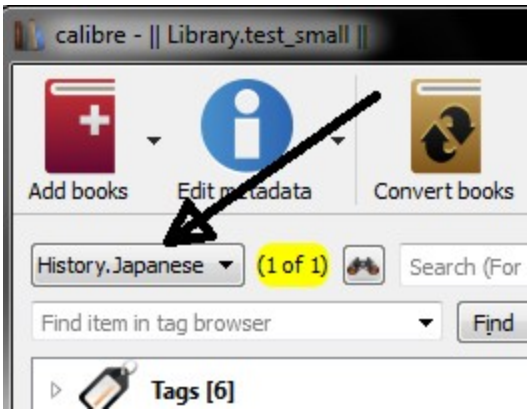
Механизм поиска Браузера тегов знает, имеет ли элемент дочерние элементы. Если это так, щелчок по элементу циклически проходит через 5 поисковых запросов вместо обычных трёх. Первый - обычный зелёный плюс, который показывает только книги с этим жанром (например, «История»). Второй - двойной плюс (показан выше), который показывает вам книги с этим жанром и всеми суб-жанрами (например, History and History.Military). Третий - это нормальный красный минус, который показывает вам книги без этого точного жанра. Четвёртый - это удвоенный минус, который показывает вам книги без этого жанра или поджанров. Пятый возвращается к началу, без знака, что означает отсутствие поиска.

10.1.3 Ограничения

Если вы ищете жанр, а затем создаете сохранённый поиск, вы можете использовать поле «Ограничить», чтобы создать виртуальную библиотеку книг с этим жанром. Это полезно, если вы хотите выполнять другие поиски в жанре или управлять/обновлять метаданные для книг в жанре. Продолжая наш пример, вы можете создать сохраненный поиск с именем «History.Japanese», сначала щелкнув по жанру Japanese в браузере тегов, чтобы выполнить поиск в поле поиска, введя History.Japanese в сохраненное поле поиска, а затем нажав кнопку «Сохранить поиск» (зелёное поле с белым плюсом справа).



После создания сохраненного поиска вы можете использовать его как ограничение.



10.1.4 Полезные шаблонные функции

Возможно, вы захотите использовать информацию о жанре в шаблоне, например, сохранить на диск или отправить на устройство. Тогда может возникнуть вопрос: «Как получить самое внешнее имя или имена жанра?». Функция шаблонов calibre - подпункты предоставляется как раз для того чтобы сделать это проще.

Например, предположим, что вы хотите добавить внешний уровень жанра к шаблону save-to-disk, чтобы создавать папки жанра, как в «History/The Gathering Storm - Churchill, Winston» . Для этого вы должны извлечь первый уровень иерархии и добавить его в начале вместе с слэшем (косой чертой) для указания что нужно создать папку. Ниже приведён шаблон:

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

См. *The template language* (страница 164) для получения дополнительной информации о шаблонах и функции `subitems()`.

10.2 Учебник по XPath

Этот учебник — введение в XPath[57], язык запросов, который можно использовать для выбора произвольных частей HTML[58] в calibre. XPath — широко используемый стандарт, и поиск Google выдаст тонну информации. Однако этот учебник сосредотачивается на решении с помощью XPath задач, связанных с электронными книгами, например поиске заголовков разделов в неструктурированном HTML-документе.

Содержание

- *Выбор по имени тега* (страница 163)
- *Выбор по атрибутам* (страница 163)
- *Выбор по содержимому тега* (страница 163)
- *Образец электронной книги* (страница 164)
- *Встроенные функции XPath* (страница 164)

[57] https://en.wikipedia.org/wiki/XPath
[58] https://en.wikipedia.org/wiki/HTML

10.2.1 Выбор по имени тега

Простейшая форма выбора — выбор тегов по имени. Предположим, что вы хотите выбрать все теги `<h2>` в документе. Запрос XPath для этого прост:

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

Префикс // означает *поиск на любом уровне документа*. Теперь предположим, что вы хотите искать теги `<span>`, которые находятся внутри тегов `<a>`. Этого можно достичь с помощью:

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

Если вы хотите искать теги на определённом уровне в документе, измените префикс:

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

Это будет соответствовать только `<p>A very short e-book to demonstrate the use of XPath.</p>` в разделе «*Образец электронной книги* (страница 164)», а не каким-либо другим тегам `<p>`. Префикс `h:` в вышеприведённых примерах необходим для соответствия тегам XHTML. Потому что внутри calibre всё содержимое представлено как XHTML. В тегах XHTML есть *пространство имён*, а `h:` — префикс пространства имён для HTML-тегов.

Теперь предположим, что вы хотите выбрать теги `<h1>` и `<h2>`. Для этого нам нужна конструкция XPath, называемая *предикат*. А *предикат* — это просто тест, который используется для выбора тегов. Тесты могут быть сколь угодно мощными, и по мере продвижения по этому учебнику вы увидите более мощные примеры. Предикат создаётся путём заключения тестового выражения в квадратные скобки:

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

В этом выражении XPath имеется несколько новых функций. Во-первых, это использование подстановочного символа `*`. Это означает *соответствие любому тегу*. Теперь посмотрим на тестовое выражение `name()='h1' or name()='h2'`. *name()* — пример *встроенной функции*. Она просто вычисляет имя тега. Таким образом, с её помощью мы можем выбрать теги, имена которых либо *h1*, либо *h2*. Обратите внимание, что функция name () игнорирует пространства имён, поэтому нет необходимости в префиксе `h:`. XPath имеет несколько полезных встроенных функций. Некоторые из них будут показаны в этом учебнике.

10.2.2 Выбор по атрибутам

Чтобы выбрать теги на основе их атрибутов, необходимо использовать предикаты:

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

Здесь оператор `@` ссылается на атрибуты тега. Вы можете использовать некоторые из *XPath built-in functions* (страница 164) (встроенных функций XPath) для выполнения более сложных сопоставлений значений атрибутов.

10.2.3 Выбор по содержимому тега

Используя XPath, вы можете выбирать теги даже на основе содержащегося в них текста. Лучший способ сделать это — призвать силу *регулярных выражений* с помощью встроенной функции re:test ():

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

Здесь оператор `.` ссылается на содержимое тега, так же как оператор `@` ссылается на его атрибуты.

10.2.4 Образец электронной книги

```
<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 Встроенные функции XPath

name()
: Имя текущего тега.

contains()
: `contains(s1, s2)` возвращает *true*, если s1 содержит s2.

re:test()
: `re:test(src, pattern, flags)` возвращает *true*, если строка *src* соответствует регулярному выражению *pattern*. Особенно полезен флаг `i`: он отвечает за нечувствительность к регистру. Хороший учебник синтаксиса регулярных выражений для начинающих можно найти по ссылке regexp syntax[59].

10.3 Язык шаблонов calibre

Язык шаблонов calibre — это специфичный для calibre язык, используемый в calibre для таких задач, как указание путей к файлам, форматирование значений и вычисление значений для столбцов, указанных пользователем. Примеры:

- Укажите структуру папок и имена файлов при сохранении файлов из библиотеки calibre на диск или в читалку электронных книг
- Определите правила добавления значков и цветов в список книг calibre.
- Определите виртуальные столбцы, содержащие данные из других столбцов.
- Расширенный поиск в библиотеке.
- Расширенный поиск и замена метаданных.

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

Язык построен на понятии «шаблон», который определяет, какие метаданные книги использовать, вычисления над этими метаданными и способ их форматирования.

10.3.1 Базовые шаблоны

Базовый шаблон состоит из одного или нескольких `шаблонных выражений`. `Шаблонное выражение` состоит из текста и имен в фигурных скобках (`{}`), которые заменяются соответствующими метаданными из обрабатываемой книги. Например, шаблон по умолчанию в calibre, используемый для сохранения книг на устройство, имеет 4 `шаблонных выражения`:

```
{author_sort}/{title}/{title} - {authors}
```

Для книги Айзека Азимова «Основание» «The Foundation» by «Isaac Asimov» им станет:

```
Asimov, Isaac/The Foundation/The Foundation - Isaac Asimov
```

Косые черты не являются `шаблонными выражениями`, потому что они находятся между ними в `{}`. Такой текст остается там, где он появляется. Например, если шаблон:

```
{author_sort} Some Important Text {title}/{title} - {authors}
```

затем для книги «Основание» шаблон создает:

```
Asimov, Isaac Some Important Text The Foundation/The Foundation - Isaac Asimov
```

`Шаблонное выражение` может получить доступ ко всем метаданным, доступным в calibre, включая настраиваемые столбцы (столбцы, которые вы создаете самостоятельно), используя `имя поиска` столбца. Чтобы найти поисковое имя для столбца (иногда называемых полями), наведите указатель мыши на заголовок столбца в списке книг calibre. Имена подстановки для настраиваемых столбцов всегда начинаются с символа `#`. Для столбцов типа серии есть дополнительное поле с именем `#lookup name_index`, которое является индексом серии для этой книги в серии. Например, если у вас есть столбец настраиваемой серии с именем `#myseries`, также будет столбец с именем `#myseries_index`. Индекс столбца стандартной серии называется `series_index`.

В дополнение к стандартным полям на основе столбцов вы также можете использовать:

- `{formats}`- Список форматов, доступных в библиотеке calibre для книги
- `{identifiers:select(isbn)}` - ISBN книги

Если метаданные для поля для данной книги не определены, тогда поле в шаблоне заменяется пустой строкой (`''`). Например, рассмотрим следующий шаблон:

```
{author_sort}/{series}/{title} {series_index}
```

Если книга Азимова «Второе основание» входит в серию «Основание», то шаблон дает:

```
Asimov, Isaac/Foundation/Second Foundation 3
```

Если для книги не введена серия, то шаблон выдает:

```
Asimov, Isaac/Second Foundation
```

Обработчик шаблонов автоматически удаляет несколько косых черт и начальные или конечные пробелы.

10.3.2 Расширенное форматирование

Помимо подстановки метаданных, шаблоны могут условно включать дополнительный текст и управлять форматированием заменяемых данных.

Условно включая текст

Иногда вы хотите, чтобы текст отображался в выводе, только если поле не пустое. Обычный случай - это series и series_index, когда вы не хотите ничего или два значения, разделенные дефисом. calibre обрабатывает этот случай, используя специальный синтаксис `шаблонного выражения`.

Например, используя приведенный выше пример Foundation, предположим, что вы хотите, чтобы шаблон создавал *Foundation - 3 - Second Foundation*. Этот шаблон производит такой вывод:

```
{series} - {series_index} - {title}
```

Однако, если в книге нет серии, шаблон создаст - - *title*, что, вероятно, не то, что вам нужно. Обычно люди хотят, чтобы в результате получился заголовок без лишних дефисов. Вы можете сделать это, используя следующий синтаксис шаблона:

```
{field:|prefix_text|suffix_text}
```

Это `шаблонное выражение` говорит, что если `field` имеет значение `XXXX`, то результатом будет `prefix_textXXXXXsuffix_text`. Если поле пусто (не имеет значения), результатом будет пустая строка (ничего), потому что префикс и суффикс игнорируются. Префикс и суффикс могут содержать пробелы.

Не используйте подшаблоны (`{ … }`) или функции (см. Ниже) в префиксе или суффиксе.

Используя этот синтаксис, мы можем решить указанную выше проблему без серии с шаблоном:

```
{series}{series_index:| - | - }{title}
```

Дефисы будут добавлены только в том случае, если у книги есть индекс серии, который есть только в том случае, если в ней есть серия. Продолжая пример с Foundation снова, шаблон создаст *Foundation - 1 - Second Foundation*.

Примечания:

- Вы должны включить двоеточие после `имени поиска`, если используете префикс или суффикс.
- Вы должны использовать либо никакой, либо оба символа `|`. Использовать один, как в `{field:| -}`, не допускается.
- Допускается отсутствие текста для префикса или суффикса, например, в `{series:|| -}`. Шаблон `{title:||}` такой же, как `{title}`.

Форматирование

Предположим, вы хотите, чтобы `series_index` был отформатирован как три цифры с ведущими нулями. Это помогает:

`{series_index: 0>3s}` - три цифры с ведущими нулями

Для завершающих нулей используйте:

`{series_index: 0<3s}`- три цифры с завершающими нулями

Если вы используете индексы серий с дробными значениями, например 1.1, и нужно чтобы десятичные точки совпадали. Например, чтобы индексы 1 и 2.5 отображались как 01.00 и 02.50 и правильно сортировались на устройстве, которое выполняет лексическую сортировку. Для этого используйте:

`{series_index: 05.2f}` - пять символов, состоящих из двух цифр с ведущими нулями, десятичной точкой и двумя цифрами после десятичной точки.

Если вам нужны только первые две буквы данных, используйте:

`{author_sort:.2}` - Только первые две буквы сортировочного имени автора

Большая часть форматирования языка шаблонов calibre из Python. Дополнительные сведения о синтаксисе расширенных операций форматирования см. в Python documentation[60].

10.3.3 Использование шаблонов для определения пользовательских столбцов

Шаблоны могут использоваться для отображения информации, которая не входит в метаданные calibre, или для отображения метаданных, отличных от обычного формата calibre. Например, чтобы отобразить `ISBN`, поле, которое штатно не отображается в calibre. Создайте настраиваемый столбец с типом *Столбец, построенный из других столбцов* (в дальнейшем называемый *составные столбцы*) и предоставьте шаблон для генерации отображаемого текста. В столбце отобразится результат оценки шаблона. Чтобы отобразить ISBN, создайте составной столбец и введите `{identifiers:select (isbn)}` в поле шаблона. Чтобы отобразить столбец, содержащий значения двух пользовательских столбцов серий, разделённых запятой, используйте `{#series1:||,}{#series2}`.

В составных столбцах можно использовать любой параметр шаблона, включая форматирование.

Примечание. Вы не можете редактировать данные, отображаемые в составном столбце. Вместо этого вы редактируете исходные столбцы. Если вы редактируете составной столбец, например, дважды щелкнув по нему, calibre откроет для редактирования шаблон, а не базовые данные.

10.3.4 Шаблоны и коммутации

Коммутационные панели используются для изменения метаданных, записанных в книги, во время операций отправки на устройство и сохранения на диск. Коммутационная панель позволяет указать шаблон предоставления данных для записи в метаданные книги. Вы можете использовать коммутационные панели для изменения следующих полей: authors, author_sort, language, publisher, tags, title, title_sort. Эта функция помогает в использовании разных метаданных в книгах на устройствах, решении проблем сортировки или отображения.

При создании коммутационной панели вы указываете формат и устройство, для которых она будет использоваться. Предоставляется специальное устройство `save_to_disk`, которое используется при сохранении форматов (в отличие от их отправки на устройство). После того, как вы выбрали формат и устройство, вы выбираете поля метаданных, которые нужно изменить, предоставляя шаблоны для предоставления новых значений. Эти шаблоны *присоединены* к своим полям назначения, отсюда и название *plugboards*. Конечно, в этих шаблонах можно использовать составные столбцы.

Плагборды довольно гибкие и могут быть написаны в режиме Single Function Mode, Template Program Mode, General Program Mode, или Python Template mode.

Когда может применяться коммутационная панель (контент-сервер, сохранение на диск или отправка на устройство), calibre выполняет поиск определенных подключаемых модулей, чтобы выбрать правильный вариант для данного формата и устройства. Например, чтобы найти подходящую коммутационную панель для книги EPUB, отправляемой на устройство ANDROID, calibre выполняет поиск в коммутационных панелях, используя следующий порядок поиска:

- коммутационная панель с точным соответствием по формату и устройству, например, `EPUB` и `ANDROID`
- коммутационная панель с точным соответствием по формату и специальным выбором `any device`, например, `EPUB` и `any device`
- коммутационная панель со специальным выбором `any format` и точным соответствием на устройстве, например, `any format` и `ANDROID`
- коммутационная панель с `any format` и `any device`

Поля тегов и авторов обрабатываются по-особому, потому что оба эти поля могут содержать более одного элемента. У книги может быть много тегов и много авторов. Когда вы указываете, что одно из этих двух полей должно быть изменено, результат шаблона проверяется, чтобы увидеть, есть ли там более одного элемента. Для

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

тегов результат будет разделен на части, если calibre найдет запятую. Например, если в шаблоне выдается значение `Thriller, Horror`, результатом будут два тега: `Thriller` и `Horror`. Невозможно поставить запятую в середине тега.

То же самое происходит с авторами, но с использованием другого символа для сокращения, *&* (амперсанд) вместо запятой. Например, если в шаблоне указано значение `Blogs, Joe&Posts, Susan`, тогда в книге будет два автора: `Blogs, Joe` и `Posts, Susan`. Если шаблон выдает значение `Blogs, Joe;Posts, Susan`, то у книги будет один автор с довольно странным именем.

Плагины влияют на метаданные, записанные в книгу, когда они сохраняются на диск или записываются на устройство. Плагины не влияют на метаданные, используемые командами «сохранить на диск» и «отправить на устройство» для создания имен файлов. Вместо этого имена файлов создаются с использованием шаблонов, введенных в соответствующем окне настроек.

10.3.5 Использование функций в шаблонах - режим одной функции

Предположим, вы хотите отобразить значение поля в верхнем регистре, когда это поле обычно находится в регистре заголовков. Вы можете сделать это с помощью шаблонных функций. Например, чтобы отобразить заголовок в верхнем регистре, используйте функцию `uppercase`, как в `{title:uppercase()}`. Чтобы отобразить его в регистре заголовка, используйте `{title:titlecase()}`.

Функции входят в форматную часть шаблона после `:` и перед первым `|` или закрывающим `}`, если не используется префикс/суффикс. Если у вас есть и формат, и ссылка на функцию, функция появляется после второго `:`. Функции возвращают соответствующее измененное значение столбца, указанного в шаблоне.

Синтаксис использования функций - один из:

```
{lookup_name:function(arguments)}
{lookup_name:format:function(arguments)}
{lookup_name:function(arguments)|prefix|suffix}
{lookup_name:format:function(arguments)|prefix|suffix}
```

За именами функций всегда должны стоять открывающие и закрывающие круглые скобки. Некоторым функциям требуются дополнительные значения (аргументы), и они заключаются в круглые скобки. Аргументы разделяются запятыми. Буквальным запятым (запятым как тексту, а не разделителям аргументов) должна предшествовать обратная косая черта (`\\`). Последний (или единственный) аргумент не может содержать закрывающую текстовую скобку.

Функции оцениваются до спецификации формата и префикса/суффикса. См. Ниже пример использования формата и функции.

Важно: если у вас есть опыт программирования, обратите внимание, что синтаксис в *Single Function Mode* не соответствует вашим ожиданиям. Строки не заключаются в кавычки, а пробелы имеют значение. Все аргументы считаются константами; нет выражений.

Не используйте подшаблоны (`{ … }`) в качестве аргументов функции. Вместо этого используйте *Template Program Mode* (страница 177) и *General Program Mode* (страница 172).

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 *Справочник по шаблонным функциям* (страница 187). The documentation tells you what arguments the functions require and what the functions do. For example, here is the documentation of the *ifempty* (страница 194) function.

- `ifempty(value, text_if_empty)` – если `value` не пустое, то вернуть это `value`, иначе вернуть `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* (страница 208)`(value)` – возвращает значение, в котором первая буква в верхнем регистре, а остальные в нижнем регистре.
- *ceiling* (страница 191)`(value)` – возвращает наименьшее целое число, большее или равное `value`. Выдаёт исключение, если `value` не является числом.
- *cmp* (страница 200)`(value, y, lt, eq, gt)` – сравнивает `value` и `y` после преобразования обоих в числа.
- *contains* (страница 194)`(value, pattern, text_if_match, text_if_not_match)` – проверяет, соответствует ли значение регулярному выражению `pattern`. Возвращает `text_if_match` при совпадении, иначе возвращает `text_if_not_match`.
- ff_““date_arithmetic`(value, calc_spec, fmt)'' -- вычислить новую дату от ``value` с помощью `calc_spec`. Вернуть новую дату, отформатированную в соответствии с необязательным `fmt`: если не указано, то результат будет в формате ISO. Строка `calc_spec` - конкатенация пар „vW“ (valueWhat), где `v` - возможно отрицательное число, а W - одна из следующих букв:
  - `s`: добавить `v` добавить к `date`
  - `m`: добавить `v` минут к `date`
  - `h`: добавить `v` часов к `date`
  - `d`: добавить `v` дней к `date`
  - `w`: добавить `v` недель к `date`
  - `y`: добавить `v` лет к `date`, где год это 365 дней.

  Пример: `'1s3d-1m'` добавит 1 секунду, прибавит 3 дня и вычтет 1 минуту из `date`.
- *encode_for_url* (страница 210)`(value, use_plus)` – возвращает значение `value` закодированное для использования в URL, как указанно с `use_plus`. Значение URL-кодированное. Далее, если `use_plus` равно `0`, то пробелы заменяются знаками `'+'` (плюс). Если это `1`, то пробелы заменяются `%20`.
- *floor* (страница 191)`(value)` – возвращает наибольшее целое число, меньшее или равное `value`. Выдаёт исключение, если `value` не является числом.
- *format_date* (страница 209)`(value, format_string)` – форматирует `value`, которое должно быть строкой даты, используя `format_string`, возвращающую строку.
- *format_number* (страница 210)`(value, template)` – интерпретирует `value` как число и форматирует это число, используя шаблон форматирования Python, такой как `{0:5.2f}` или `{0:,d}` или ``${0:5,.2f}`. Шаблон форматирования должен начинаться с `{0:` и заканчиваться `}`, как в приведенных выше примерах. Исключение: вы можете опустить начальные `{0:` и завершающие `}`, если шаблон формата содержит только формат. См. [URL href=»https://manual.calibre-ebook.com/template_lang.html»] Template Language[/URL] и

[URL href=»https://docs.python.org/3/library/string.html#formatstrings»] Python[/URL]. Возвращает пустую строку, если сбой форматирования.

- *fractional_part* (страница 191) `(value)` – возвращает значение после десятичной точки.
- *human_readable* (страница 210) `(value)` – ожидает, что значение будет числом, и возвращает строку, представляющую это число в КБ, МБ, ГБ и т. д.
- *ifempty* (страница 194) `(value, text_if_empty)` – если `value` не пустое, то вернуть это `value`, иначе вернуть `text_if_empty`.
- *language_strings* (страница 206) `(value, localize)` – вернуть значение названий языков для языковых кодов (см. здесь имена и коды[61]) , переданных в качестве значения `value`. Пример: `{languages:language_strings()}`. Если `localize` равно нулю, вернуть строки на английском языке. Если `localize` не равно нулю, вернуть строки на языке текущей локали. `lang_codes` - это список, разделённый запятыми.
- *list_contains* (страница 202) `(value, separator, [ pattern, found_val, ]* not_found_val)` – интерпретирует `value` как список элементов, разделенных разделителем `separator`, проверяя `pattern` для каждого элемента в списке.
- *list_count* (страница 196) `(value, separator)` – интерпретирует значение как список элементов, разделенных символом `separator` и возвращает количество элементов в списке.
- *list_count_matching* (страница 197) `(value, pattern, separator)` – интерпретирует `value` как список элементов, разделённых `separator`, возвращая количество элементов в списке, которые соответствуют регулярному выражению `pattern`. Псевдонимы: `list_count_matching()`, `count_matching()`
- *list_item* (страница 202) `(value, index, separator)` – обрабатывает значение как список элементов, разделённых `separator`, возвращая элемент с номером index. Первый элемент имеет номер ноль. Последний элемент с номером индекса `-1` как *list_item(-1,separator)*. Если элемент не в списке, то возвращается пустое значение. Разделитель (separator) работает так же, как и в функции *count*, обычно запятая, но амперсанд для списков, вроде author.
- *list_sort* (страница 199) `(list, direction, separator)` – вернуть список, отсортированный с использованием лексической сортировки без учета регистра. Если `direction` равно нулю, список сортируется по возрастанию, иначе по убыванию. Элементы списка разделены разделителем `separator`, как и элементы в возвращаемом списке.
- *lookup* (страница 201) `(value, [ pattern, key, ]* else_key)` – Шаблоны будут проверяться по значению `value` по порядку. Если шаблон совпадает, возвращается значение поля, названного `key`. Если ни один шаблон не совпадает, возвращается значение поля с именем `else_key`. Смотрите функцию *switch()* (страница 201).
- *lowercase* (страница 208) `(value)` – возвращает `value` в нижнем регистре.
- *mod* (страница 192) `(value, y)` – возвращает `floor` остаток от `value / y`. Выдаёт исключение, если `value` или `y` не является числом.
- *rating_to_stars* (страница 210) `(value, use_half_stars)` – Возвращает рейтинг как `value` в виде строки, состоящей из звездочек (★"). Значение должно быть числом от `0` до `5`. Задайте для `use_half_stars` значение `1`, если вы хотите, чтобы символы половинной звезды для дробных чисел были доступны в настраиваемых столбцах рейтингов.
- *re* (страница 195) `(value, pattern, replacement)` – вернуть значение `value` после применения регулярного выражения. Все вхождения `pattern` в значении `value` заменяются на `replacement`. Язык шаблонов не регистрозависим Python regular expressions[62].

[61] https://www.loc.gov/standards/iso639-2/php/code_list.php
[62] https://docs.python.org/3/library/re.html

- *re_group* (страница 195) `(value, pattern [, template_for_group]*)` – вернуть строку, полученную путем применения шаблона регулярного выражения `pattern` к `value` и замены каждого сопоставленного экземпляра строкой, вычисленной путем замены каждой сопоставленной группы значением, возвращаемым соответствующим шаблоном. В [URL href=»https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode»] программном режиме шаблона, как и в случае с `template` и `eval`, вы используете `[[` для `{` и `]]` для `}`. В следующем примере выполняется поиск ряда, состоящего более чем из одного слова, и первое слово переводится в верхний регистр [CODE] program: re_group(field(„series“), «(S*)(.*)», «{$:uppercase()}», «{$}»)“} [/CODE]
- *round* (страница 192) `(value)` – возвращает ближайшее к `value` целое число. Выдаёт исключение, если `value` не является числом.
- *select* (страница 203) `(value, key)` – обрабатывает `value` как разделённый запятыми список элементов вида `id:id_value` (формат `identifier` calibre). Функция находит первую пару с совпадающим с `key` идентификатором, и возвращает соответствующее `id_value`. При отсутствии совпадений возвращается пустая строка.
- *shorten* (страница 195) `(value, left chars, middle text, right chars)` – Возвращает сокращённую версию значения``value``, состоящую из символов `left chars` с начала значения, за которыми следует `middle text`, за которыми следуют символы `right chars` с конца значения. `left chars` и `right chars` должны быть неотрицательными целыми числами.

  Пример: предположим, что вы хотите отобразить заголовок длиной не более 15 символов. Один из шаблонов, который делает это, - `{title:short(9,-,5)}`. Для книги с названием «Ancient English Laws in the Times of Ivanhoe» результатом будет `Ancient E-nhoe`: первые 9 символов названия, `-`, затем последние 5 символов. Если длина значения меньше, чем `left chars` + `right chars` + длина `middle text`, то значение будет возвращено без изменений. Например, название *The Dome* не изменится.
- *str_in_list* (страница 203) `(separator, [ string, found_val, ]+ not_found_val)` – интерпретирует значение как список элементов, разделенных `separator`, затем сравнивает `string` с каждым значением в списке. `string` не является регулярным выражением. Если `string` равна любому элементу (без учета регистра), вернуть соответствующее значение `found_val`. Если `string` содержит разделители `separator`, она также рассматривается как список, и каждое подзначение проверяется. Пары `string` и `found_value` могут повторяться сколько угодно раз, что позволяет возвращать разные значения в зависимости от значения строки. Если ни одна из строк не совпадает, возвращается `not_found_value`. Строки проверяются по порядку. Возвращается первое совпадение.
- *subitems* (страница 200) `(value, start_index, end_index)` – Эта функция разделяет списки на иерархические элементы, подобные тегам, такие как жанры.
- *sublist* (страница 200) `(value, start_index, end_index, separator)` – интерпретирует `value` как список значений, разделённых `separator`, возвращая новый список с началом в `start_index` с концом в `end_index`. Первый пункт - номер 0 (ноль). Если индекс - отрицательный, то его начало - от конца списка. В особых случаях end_index нуля считается длиной списка.

  Примеры, использующие режим базового шаблона и предополагающие что столбец тегов (разделённый запятыми) содержит «A, B, C»:

  - `{tags:sublist(0,1,\,)}` returns «A»
  - `{tags:sublist(-1,0,\,)}` returns «C»
  - `{tags:sublist(0,-1,\,)}` returns «A, B»
- *substr* (страница 196) `(value, start, end)` – возвращает подстроку - символы строки `value` от `start` до `end`. Первый символ строки `value` имеет номер ноль. Если `end` отрицательный - это сколько символов будет отсчитываться с конца строки (справа). Если `end` равен 0 (ноль), то это - последний символ строки. Например, `substr('12345', 1, 0)` вернёт `'2345'`, и `substr('12345', 1, -1)` вернёт `'234'`.

- *swap_around_articles* (страница 196) `(value, separator)` – возвращает `value` значение со статьями, перемещёнными в конец, разделёнными точкой с запятой. Значение `value` может быть списком, тогда обрабатывается каждый элемент в списке. Если `value` - список, укажите `separator`. Если `separator` не указан (или его значение - пустая строка), то значение `value` рассматривается как одно значение, а не как список. `articles` используются calibre для создания ``title_sort.
- *swap_around_comma* (страница 196) `(value)` – если значение `value` имеет вид `B, A`, возвращает `A B`. Это наиболее полезно для преобразования имён вида LN, FN (Ф, И О) к виду FN LN (И О Ф). Если значение не содержит запятой в `value`, то оно возвращается без изменений.
- *switch* (страница 201) `(value, [patternN, valueN,]+ else_value)` – для каждой пары `patternN, valueN`, проверяет, соответствует ли значение регулярному выражению *patternN*`, и если да, возвращает связанное `` valueN``. Если нет совпадений с шаблоном, возвращается `else_value`. У вас может быть столько пар `patternN, valueN`, сколько хотите. Возвращается первое совпадение.
- *test* (страница 196) `(value, text_if_not_empty, text_if_empty)` – возвращает `text_if_not_empty` если значение не пусто, иначе возвращает `text_if_empty`.
- *titlecase* (страница 208) `(value)` – возвращает `value` в регистре ЗАГОЛОВКОВ.
- *transliterate* (страница 196) `(value)` – возвращает строку в латинском алфавите, которая по звучанию похожа на слова исходной строки. Например, если `value` было `Фёдор Михáйлович Достоевский`, то функция вернёт `Fiodor Mikhailovich Dostoievskii`.
- *uppercase* (страница 208) `(value)` – возвращает `value` в верхнем регистре

Использование функций и форматирования в одном шаблоне

Предположим, у вас есть целочисленный настраиваемый столбец `#myint`, который вы хотите отображать с ведущими нулями, как в `003`. Один из способов сделать это - использовать формат 03. Однако по умолчанию, если число (целое или с плавающей запятой) равно нулю, тогда значение отображается как пустая строка, поэтому нулевые значения будут давать пустую строку, а не `000`. Если вы хотите увидеть значения `000`, используйте как строку формата, так и функцию ifempty, чтобы изменить пустое значение обратно в ноль. Шаблон будет:

```
{#myint:0>3s:ifempty(0)}
```

Обратите внимание, что вы также можете использовать префикс и суффикс. Если вы хотите, чтобы номер отображался как `[003]` или `[000]`, то используйте шаблон:

```
{#myint:0>3s:ifempty(0)|[|]}
```

10.3.6 Общий программный режим

`Общий программный режим` (`GPM`) заменяет `шаблонные выражения` программой, написанной на `языке шаблонов`. Синтаксис языка определяется следующей грамматикой:

```
program         ::= 'program:' expression_list
expression_list ::= top_expression [ ';' top_expression ]*
top_expression  ::= or_expression
or_expression   ::= and_expression [ '||' and_expression ]*
and_expression  ::= not_expression [ '&&' not_expression ]*
not_expression  ::= [ '!' not_expression ]* | concatenate_expr
concatenate_expr::= compare_expr [ '&' compare_expr ]*
compare_expr    ::= add_sub_expr [ compare_op add_sub_expr ]
compare_op      ::= '==' | '!=' | '>=' | '>' | '<=' | '<' |
                    'in' | 'inlist' | 'inlist_field' |
                    '==#' | '!=#' | '>=#' | '>#' | '<=#' | '<#'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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
```

Примечания:

- `top_expression` всегда имеет значение. Значение `expression_list` - это значение последнего `top_expression` в списке. Например, значение списка выражений `1;2;'foobar';3` равно `3`.
- В логическом контексте любое непустое значение - `True`
- В логическом контексте пустым значением является `False`
- Строки и числа могут использоваться как взаимозаменяемые. Например, `10` и `'10'` - это одно и то же.
- Комментарии - это строки, начинающиеся с символа „#“. Комментарии, начинающиеся позже в строке, не поддерживаются.

Приоритет оператора

Приоритет оператора (порядок оценки) от наивысшего (оценивается первым) до самого низкого (оценивается последним):

- Вызовы функций, константы, выражения в скобках, выражения операторов, выражения присваивания, ссылки на поля.
- Унарный плюс (+) и минус (-). Эти операторы вычисляют справа налево.

  Эти и все другие арифметические операторы возвращают целые числа, если результат выражения дает дробную часть, равную нулю. Например, если выражение возвращает `3.0`, оно изменяется на `3`.
- Умножение (`*`) и деление (`/`). Эти операторы ассоциативны и выполняются слева направо. Используйте круглые скобки, если хотите изменить порядок оценки.
- Сложение (+) и вычитание (-). Эти операторы ассоциативны и выполняются слева направо.
- Числовые и строковые сравнения. Эти операторы возвращают `1`, если сравнение прошло успешно, в противном случае - пустую строку (`''`). Сравнения не ассоциативны: `a < b < c` - синтаксическая ошибка.
- Конкатенация строк (`&`). Оператор `&` возвращает строку сформированную путем объединения левого и правого выражений. Пример: `'aaa' & 'bbb'` возвращает `'aaabbb'`. Оператор является ассоциативным и оценивает слева направо.
- Унарное логическое НЕ (`!`). Этот оператор возвращает `'1'`, если выражение ложно (возвращает пустую строку), в противном случае `''`.
- Логическое и (`&&`). Этот оператор возвращает „1“, если и левое, и правое выражения имеют значение True, или пустую строку `''`, если любое из них имеет значение False. Он ассоциативен, вычисляется слева направо и выполняет `short-circuiting `_.
- Логическое ИЛИ (`||`). Этот оператор возвращает `1`, если левое или правое выражение истинно, или `''`, если оба ложны. Он ассоциативен, вычисляется слева направо и выполняет *short-circuiting `_. Это `включающее ИЛИ*, возвращающее``“1“``, если и левое, и правое выражения истинны.

Ссылки на поля

A `field_reference` evaluates to the value of the metadata field named by lookup name that follows the `$` or `$$`. Using `$` is equivalent to using the *field* (страница 205) function. Using `$$` is equivalent to using the *raw_field* (страница 206) function. Examples:

```
* $authors ==> field('authors')
* $#genre ==> field('#genre')
* $$pubdate ==> raw_field('pubdate')
* $$#my_int ==> raw_field('#my_int')
```

If выражения

Выражения `if` сначала оценивают условие. Если `condition` равно True (непустое значение), тогда `expression_list` в предложении `then` оценивается. Если оно равно False, то при его наличии вычисляется список выражений в предложении `elif` или `else`. Части `elif` и `else` не являются обязательными. Слова `if`, `then`, `elif`, `else` и `fi` зарезервированы; вы не можете использовать их в качестве имен идентификаторов. Вы можете помещать символы новой строки и пробелы везде, где они имеют смысл. `condition` - это `top_expression`, а не `expression_list`; точки с запятой не допускаются. `expression_list` - это разделенные точкой с запятой последовательности «top_expressions». Выражение `if` возвращает результат последнего выражения `top_expression` в вычисленном списке выражений или пустую строку, если список выражений не оценивался.

Примеры:

```
* program: if field('series') then 'yes' else 'no' fi
* program:
      if field('series') then
          a = 'yes';
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

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

Пример вложенного `if`:

```
program:
  if field('series') then
    if check_yes_no(field('#mybool'), '', '', '1') then
      'yes'
    else
      'no'
    fi
  else
    'no series'
  fi
```

Как сказано выше, `if` выдаёт значение. Это означает, что всё следующее эквивалентно:

```
* 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 выражения

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* (страница 199) 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.

Пример: этот шаблон удаляет первое иерархическое имя для каждого значения в жанре (`#genre`), создавая список с новыми именами:

```
program:
  new_tags = '';
  for i in '#genre':
    j = re(i, '^.*?\.(.*)$', '\1');
    new_tags = list_union(new_tags, j, ',')
  rof;
  new_tags
```

Если исходный жанр - *History.Military, Science Fiction.Alternate History, ReadMe*, то шаблон возвращает *Military, Alternate History, ReadMe*. Вы можете использовать этот шаблон в calibre *Массовое редактирование метаданных*

- *Поиск и замена* с помощью: guilabel:` Search for`, установленного на `template`, чтобы удалить первый уровень иерархии и присвоить полученное значение Жанру.

Примечание: последняя строка в шаблоне, `new_tags`, в этом случае не является строго необходимой, потому что `for` возвращает значение последнего `top_expression` в списке выражений. Значение присваивания - это значение его выражения, поэтому значение оператора `for` - это то, что было присвоено `new_tags`.

Определение функции

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.

Аргументы позиционные. При вызове функции предоставленные аргументы сопоставляются слева направо с определенными параметрами, при этом значение аргумента присваивается параметру. Будет ошибкой указывать больше аргументов, чем определено параметров. Параметры могут иметь значения по умолчанию, например `a = 25`. Если для этого параметра не указан аргумент, используется значение по умолчанию, в противном случае параметру присваивается пустая строка.

Оператор `return` можно использовать в локальной функции.

Функция должна быть определена до того, как её можно будет использовать.

Пример. Этот шаблон вычисляет приблизительную продолжительность в годах, месяцах и днях из числа дней. Функция `to_plural()` форматирует вычисляемые значения. Обратите внимание, что в примере также используется оператор `&`:

```
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')
```

Операторы отношения

Relational operators return `'1'` if the comparison is true, otherwise the empty string (`''`).

Есть две формы операторов отношения: сравнение строк и сравнение чисел.

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.

Операторы сравнения чисел: `==#`, `!=#`, `<#`, `<=#`, `>#`, `>=#`. Левое и правое выражения должны оцениваться как числовые значения с двумя исключениями: как строковое значение «None» (неопределенное поле), так и пустая строка оцениваются как нулевое значение.

Примеры:

- `program: field('series') == 'foo'` returns `'1'` if the book's series is *foo*, otherwise `''`.
- `program: 'f.o' in field('series')` вернёт `'1'` если серия книги соответствует регулярному выражению `f.o` (например, *foo*, *Off Onyx* и т. д.), иначе `''`.
- `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 `>` оператор выполняет лексическое сравнение.
- `program: if 11 ># 2 then 'yes' else 'no' fi` вернёт `'yes'` потому что оператор `>#` выполняет числовое сравнение.

Functions in General Program Mode

See *Справочник по шаблонным функциям* (страница 187) for the list of functions built into the template language.

Примечания:

- As opposed to *Single Function Mode* (страница 168), in General Program Mode you must specify the first parameter `value`.
- All parameters are expression_lists (see the grammar above).

10.3.7 Более сложные программы в шаблонных выражениях - Template Program Mode

Template Program Mode (*TPM*) - смесь General Program Mode и :ref:` Single Function Mode`. TPM отличается от режима одиночной функции тем, что он позволяет писать шаблонные выражения, которые ссылаются на другие поля метаданных, используют вложенные функции, изменяют переменные и выполняют арифметические операции. Он отличается от *General Program Mode* тем, что шаблон заключен между символами``{`` и``}``и не начинается со слова `program:`. Программная часть шаблона - это список выражений General Program Mode.

Пример: предположим, что вы хотите, чтобы шаблон отображал серию книги, если она есть, в противном случае показывала бы значение настраиваемого поля #genre. Вы не можете сделать это в *Single Function Mode* (страница 168), потому что вы не можете ссылаться на другое поле метаданных в выражении шаблона. В *TPM* это можно сделать, как демонстрирует следующее выражение:

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

Пример показывает несколько вещей:

- *TPM* используется, если выражение начинается с `:'` и заканчивается `'}`. Предполагается, что все остальное находится в *Single Function Mode* (страница 168).

  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.
- пробелы игнорируются и могут использоваться в любом месте выражения.
- константные строки заключаются в соответствующие кавычки, либо `'`, либо `''`.

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 Режим шаблона Python

Python Template Mode (PTM) lets you write templates using native Python and the calibre API[63]. 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.

Шаблон PTM начинается с:

```
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'
```

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

Вы можете добавить приведенный выше текст в свой шаблон с помощью контекстного меню, доступ к которому обычно осуществляется с помощью щелчка правой кнопкой мыши. Комментарии несущественны и могут быть удалены. Вы должны использовать отступы Python.

Объект контекста поддерживает `str(context)`, который возвращает строку содержимого контекста, и `context.attributes`, который возвращает список имён атрибутов в контексте.

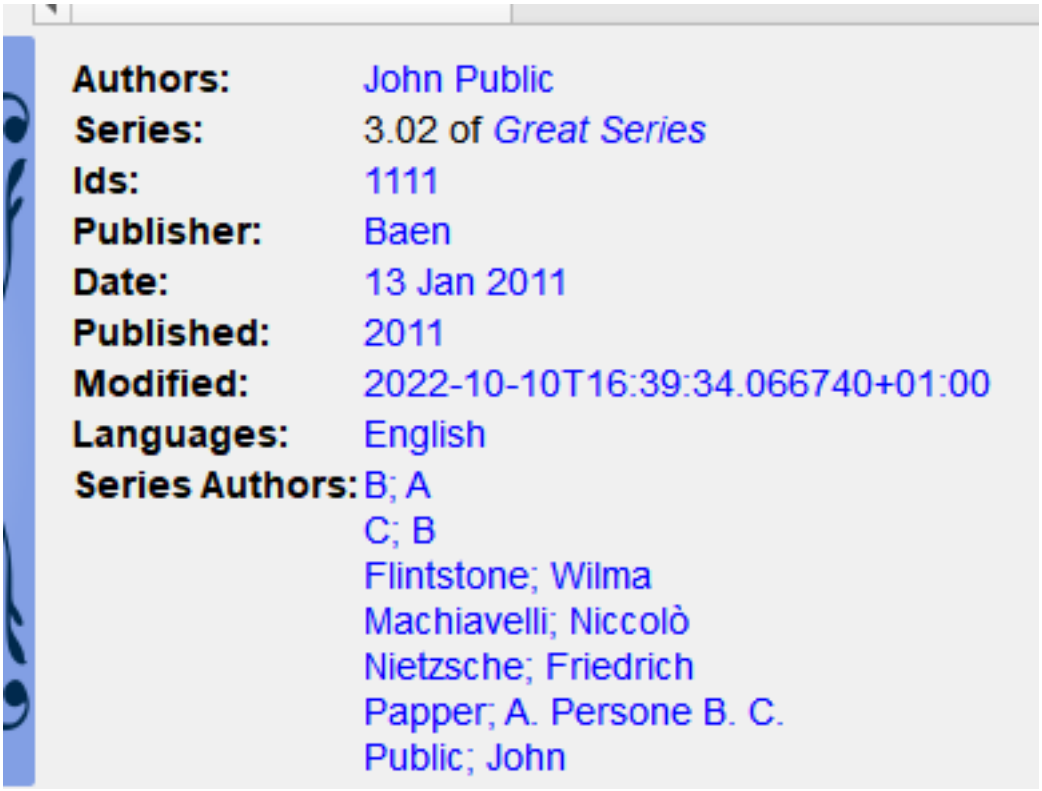
Атрибут `context.funcs` позволяет вызывать встроенные и пользовательские функции шаблонов, а также Сохранённые шаблоны GPM/Python, чтобы вы могли выполнять их непосредственно в своем коде. Функции извлекаются по их именам. Если имя конфликтует с ключевым словом Python, добавьте в конце имени символ подчеркивания. Примеры:

```
context.funcs.list_re_group()
context.funcs.assert_()
```

Вот пример шаблона PTM, который создает список всех авторов серии. Список хранится как *Столбец, состоящий из других столбцов, ведущих себя как теги*. Он отображается в *Описание книги* и отмечен как *Категории в отдельных строках* (в *Настройки → Оформление → Описание книги*). Этот параметр требует, чтобы список был разделён запятыми. Чтобы удовлетворить это требование, шаблон преобразует запятые в именах авторов в точки с запятой, а затем создаёт список авторов, разделённых запятыми. Затем авторы сортируются, поэтому в шаблоне используется 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))
```

Вывод в *Описание книги* выглядит так:



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* (страница 211)`(path, [query_name, query_value]+)` – эта функция - простейший способ создания URL-адреса запроса. Она использует `path`, веб-сайт и страницу, которые вы хотите запросить и пары `query_name`, `query_value`, из которых строится запрос. Как правило, `query_value` должно быть закодировано в URL. С помощью этой функции оно всегда закодировано, пробелы всегда заменяются знаками „+“.

- *make_url_extended* (страница 211)`(...)` – эта функция аналогична *make_url()* (страница 211) но дает вам больше контроля над компонентами URL. Компонентами URL являются

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

  См. Uniform Resource Locator[64] в Wikipedia.

  У функции два варианта:

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

  и

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

- *query_string* (страница 212)`([query_name, query_value, how_to_encode]+)` – возвращает строку запроса URL сконструированную из триад `query_name, query_value, how_to_encode`. Строка запроса это серия элементов выглядящих как `query_name=query_value`, где `query_value` это URL-кодирование. Элементы запроса разделяются символами `'&'` (ampersand).

- *encode_for_url* (страница 210)`(value, use_plus)` – возвращает значение `value` закодированное для использования в URL, как указанно с `use_plus`. Значение URL-кодированное. Далее, если `use_plus` равно `0`, то пробелы заменяются знаками `'+'` (плюс). Если это `1`, то пробелы заменяются `%20`.

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* (страница 211) 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* (страница 211), *query_string* (страница 212), and *encode_for_url* (страница 210) 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 *Схема URL-адресов calibre://* (страница 259) 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

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

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* (страница 213) 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* (страница 195):

```
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 Сохранённые шаблоны

И *Общий программный режим* (страница 172), и *Режим шаблона Python* (страница 178) поддерживают сохранение шаблонов и вызов этих шаблонов из другого шаблона, во многом аналогично вызову хранимых функций. Вы сохраняете шаблоны, используя *Настройки → Расширенные → Шаблонные функции*. В этом диалоговом окне - дополнительная информация. Вы вызываете шаблон так же, как вызываете функцию, передавая позиционные аргументы, если необходимо. Аргументом может быть любое выражение. Примеры вызова шаблона, предполагая, что сохраненный шаблон называется `foo`:

- `foo()` – вызов шаблона без аргументов.
- `foo(a, b)` вызывает шаблон, передавая значения переменных `a` и `b`.
- `foo(if field('series') then field('series_index') else 0 fi)` – если в книге есть `series`, то передать `series_index`, в противном случае - значение `0`.

В GPM вы извлекаете аргументы, переданные при вызове сохраненного шаблона, с помощью функции `arguments`. Она и объявляет, и инициализирует локальные переменные, фактически параметры. Переменные являются позиционными; они получают значение параметра, заданного при вызове, в той же позиции. Если соответствующий параметр не указан в вызове, тогда `arguments` присваивает этой переменной предоставленное значение по умолчанию. Если значение по умолчанию отсутствует, переменной присваивается пустая строка. Например, следующая функция `arguments` объявляет 2 переменные, `key`, `alternate`:

```
arguments(key, alternate='series')
```

Примеры, снова предполагаем, что сохраненный шаблон назван `foo`:

- `foo('#myseries')` – аргументу `key` присваивается значение `'myseries'`, а аргументу `alternate` присваивается значение по умолчанию `'series'`.

- `foo('series', '#genre') переменной ``key` присваивается значение `'series'`, а переменной `alternate` присваивается значение `'#genre'`.
- `foo()` - переменной `key` присваивается пустая строка, а переменной `alternate` присваивается значение `'series'`.

В PTM аргументы передаются в параметре `arguments`, который представляет собой список строк. Невозможно указать значения по умолчанию. Вы должны проверить длину списка `аргументов`, чтобы убедиться, что количество аргументов соответствует ожидаемому.

Простой способ проверить сохраненные шаблоны - использовать диалоговое окно `Тестер шаблонов`. Для облегчения доступа используйте сочетание клавиш в *Preferences-Advanced-Keyboard shortcuts-Template tester*. Ярлык для диалогового окна `Сохраненные шаблоны`" поможет быстрее переключаться между тестером и редактированием исходного кода сохраненного шаблона.

10.3.11 Передача шаблонам дополнительной информации

Разработчик может выбрать передачу дополнительной информации процессору шаблонов, например метаданные книги для конкретного приложения или информацию о том, что процессору предлагается сделать. Шаблон может получить доступ к этой информации и использовать ее во время оценки.

** Разработчик: как передать дополнительную информацию **

Дополнительная информация - это словарь `dict` Python, содержащий пары `variable_name: variable_value`, где значения должны быть строками. Шаблон может получить доступ к `dict`, создавая локальные переменные шаблона с именем `variable_name`, содержащие значение `variable_value`. Пользователь не может изменить имя, поэтому лучше использовать имена, которые не будут конфликтовать с другими локальными переменными шаблона, например, добавив к имени префикс подчеркивания _.

Этот dict передается процессору шаблонов (`formatter`) с использованием именованного параметра `global_vars=your_dict`. Полная сигнатура метода:

```
def safe_format(self, fmt, kwargs, error_value, book,
                column_name=None, template_cache=None,
                strip_results=True, template_functions=None,
                global_vars={})
```

** Автор шаблона: как получить доступ к дополнительной информации **

Вы получаете доступ к дополнительной информации (словарю `globals`) в шаблоне с помощью шаблонной функции:

```
globals(id[=expression] [, id[=expression]]*)
```

где `id` - любое допустимое имя переменной. Эта функция проверяет, содержит ли дополнительная информация, предоставленная разработчиком, имя. Если это так, функция присваивает предоставленное значение локальной переменной шаблона с этим именем. Если имя отсутствует в дополнительной информации и если предоставлено `expression`, то `expression` оценивается, и результат присваивается локальной переменной. Если ни значение, ни выражение не предоставлены, функция присваивает пустую строку (`''`) локальной переменной.

Шаблон может установить значение в словаре `globals` с помощью шаблонной функции:

```
set_globals(id[=expression] [, id[=expression]]*)
```

Эта функция устанавливает `globals` dict key:value пару `id:value`, где `value` - значение шаблонной локальной переменной `id`. Если эта локальная переменная не существует, тогда `value` устанавливается равным результату вычисления выражения `expression`.

10.3.12 Замечания о разнице между режимами

Три программных режима *Single Function Mode* (страница 168) (SFM), *Template Program Mode* (страница 177) (*TPM*) и *General Program Mode* (страница 172) (*GPM*) работают по-разному. SFM задуман как „простой“, поэтому он скрывает много тонкостей в языке программирования.

Различия:

- В SFM значение столбца всегда передается как „invisible“ «невидимый» первый аргумент функции, включенной в шаблон.
- SFM не поддерживает разницу между переменными и строками; все значения являются строками.
- Следующий шаблон SFM возвращает либо название серии, либо строку «no series»:

```
{series:ifempty(no series)}
```

  Эквивалентный шаблон в *TPM*

```
{series:'ifempty($, 'no series')'}
```

  Эквивалентный шаблон в *GPM*

```
program: ifempty(field('series'), 'no series')
```

  Первый аргумент `ifempty` - это значение поля `series`. Второй аргумент - это строка `no series`. В SFM автоматически передается первый аргумент, значение поля (невидимый аргумент).

- Некоторые функции-шаблоны, например, `booksize()` и `current_library_name()`, не принимают аргументов. Из-за „невидимого аргумента“ вы не можете использовать эти функции в SFM.
- Вложенные функции, когда функция вызывает другую функцию для вычисления аргумента, не могут использоваться в SFM. Например, этот шаблон, предназначенный для возврата первых 5 символов значения серии в верхнем регистре, не будет работать в SFM:

```
{series:uppercase(substr(0,5))}
```

- *TPM* и *GPM* поддерживают вложенные функции. Вышеупомянутый шаблон в *TPM* будет:

```
{series:'uppercase(substr($, 0,5))'}
```

  В GPM это было бы:

```
program: uppercase(substr(field('series'), 0,5))
```

- Как отмечалось выше в разделе Template Program Mode, использование символов `{` и``}`` в строковых литералах *TPM* может привести к ошибкам или неожиданным результатам, поскольку они сбивают с толку обработчик шаблонов. Он пытается рассматривать их как границы шаблона, а не как символы. В некоторых, но не во всех случаях, вы можете заменить `{` на `[[` и `}` на *]]*. Как правило, если ваша программа содержит символы `{` и``}``, вам следует использовать *General Program Mode*.

10.3.13 Пользовательские шаблонные функции Python

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* (страница 210), *list_count_field* (страница 197), and *check_yes_no* (страница 194) 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* (страница 204)
  - *connected_device_uuid* (страница 204)
  - *current_virtual_library_name* (страница 204)
  - *is_marked* (страница 206)
  - *virtual_libraries* (страница 207)

10.3.15 Особые примечания для шаблонов сохранения/отправки

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.

Например, предположим, что нам нужна структура папок *series/series_index - title*, с оговоркой, что если серия не существует, то заголовок должен находиться в верхней папке. Шаблон для этого:

```
{series:||/}{series_index:|| - }{title}
```

Косая черта и дефис появляются только в том случае, если серия не пуста.

Функция поиска позволяет нам выполнять еще более интересную обработку. Например, предположим, что если в книге есть серия, то нам нужна структура папок *series/series index - title.fmt*. Если в книге нет серии, нам нужна структура папок *genre/author_sort/title.fmt*. Если у книги нет жанра, мы хотим использовать „Unknown“. Нам нужны два совершенно разных пути, в зависимости от значения серии.

Чтобы этого добиться, мы:

1. Создать составное поле (дать ему имя для поиска #aa), содержащее `{series}/{series_index} - {title}`. Если серия не пуста, тогда этот шаблон создаст *series/series_index - title*.
2. Создать составное поле (дать ему имя для поиска #bb), содержащее `{#genre: ifempty (Unknown)}/{author_sort}/{title}`. Этот шаблон создаст *genre /author_sort/title*, где пустой жанр заменяется на *Unknown*.
3. Установить для шаблона сохранения значение `{series:lookup(.,#aa,#bb}`. Этот шаблон выбирает составное поле `#aa`, если серия не пуста, и составное поле `#bb`, если серия пуста, поэтому у нас есть два совершенно разных пути сохранения, в зависимости от того, пуста ли *series*.

10.3.16 Советы

- Используйте тестер шаблонов для тестирования шаблонов. Добавьте тестер в контекстное меню для книг в библиотеке и/или назначьте ему сочетание клавиш.
- Шаблоны могут использовать другие шаблоны, ссылаясь на составные столбцы, построенные с помощью желаемого шаблона. В качестве альтернативы вы можете использовать сохраненные шаблоны.
- В коммутации полю можно назначить соответствующее его типу пустое значение при помощи специального шаблона `{}`. Он всегда соответствует пустой строке.
- Описанный выше метод отображения чисел, даже если они имеют нулевое значение, работает со стандартным полем series_index.

10.3.17 Справочник по шаблонным функциям

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.

- *Арифметический* (страница 191)
  - *add* (страница 191)
  - *ceiling* (страница 191)
  - *divide* (страница 191)
  - *floor* (страница 191)
  - *fractional_part* (страница 191)
  - *mod* (страница 192)
  - *multiply* (страница 192)
  - *round* (страница 192)
  - *subtract* (страница 192)
- *Булево* (страница 192)
  - *and* (страница 192)
  - *not* (страница 192)
  - *or* (страница 192)
- *Другое* (страница 193)
  - *arguments* (страница 193)
  - *assign* (страница 193)
  - *globals* (страница 193)
  - *is_dark_mode* (страница 193)
  - *print* (страница 193)
  - *set_globals* (страница 193)

- *Манипулирование строками* (страница 194)
  - *character* (страница 194)
  - *check_yes_no* (страница 194)
  - *contains* (страница 194)
  - *field_exists* (страница 194)
  - *ifempty* (страница 194)
  - *re* (страница 195)
  - *re_group* (страница 195)
  - *shorten* (страница 195)
  - *strcat* (страница 195)
  - *strcat_max* (страница 195)
  - *strlen* (страница 196)
  - *substr* (страница 196)
  - *swap_around_articles* (страница 196)
  - *swap_around_comma* (страница 196)
  - *test* (страница 196)
  - *transliterate* (страница 196)
- *Манипуляция списком* (страница 196)
  - *list_count* (страница 196)
  - *list_count_field* (страница 197)
  - *list_count_matching* (страница 197)
  - *list_difference* (страница 197)
  - *list_equals* (страница 197)
  - *list_intersection* (страница 197)
  - *list_join* (страница 198)
  - *list_re* (страница 198)
  - *list_re_group* (страница 198)
  - *list_remove_duplicates* (страница 198)
  - *list_sort* (страница 199)
  - *list_split* (страница 199)
  - *list_union* (страница 199)
  - *range* (страница 199)
  - *subitems* (страница 200)
  - *sublist* (страница 200)
- *Относящийся* (страница 200)

  - *cmp* (страница 200)
  - *first_matching_cmp* (страница 201)
  - *strcmp* (страница 201)
  - *strcmpcase* (страница 201)
- *Перебор значений* (страница 201)
  - *first_non_empty* (страница 201)
  - *lookup* (страница 201)
  - *switch* (страница 201)
  - *switch_if* (страница 202)
- *Поиск по списку* (страница 202)
  - *identifier_in_list* (страница 202)
  - *list_contains* (страница 202)
  - *list_item* (страница 202)
  - *select* (страница 203)
  - *str_in_list* (страница 203)
- *Получить значения из метаданных* (страница 203)
  - *annotation_count* (страница 203)
  - *approximate_formats* (страница 203)
  - *author_links* (страница 203)
  - *author_sorts* (страница 204)
  - *booksize* (страница 204)
  - *connected_device_name* (страница 204)
  - *connected_device_uuid* (страница 204)
  - *current_library_name* (страница 204)
  - *current_library_path* (страница 204)
  - *current_virtual_library_name* (страница 204)
  - *field* (страница 205)
  - *formats_modtimes* (страница 205)
  - *formats_paths* (страница 205)
  - *formats_sizes* (страница 205)
  - *has_cover* (страница 205)
  - *is_marked* (страница 206)
  - *language_codes* (страница 206)
  - *language_strings* (страница 206)
  - *ondevice* (страница 206)

  - *raw_field* (страница 206)
  - *raw_list* (страница 206)
  - *series_sort* (страница 207)
  - *user_categories* (страница 207)
  - *virtual_libraries* (страница 207)
- *Рекурсия* (страница 207)
  - *eval* (страница 207)
  - *template* (страница 207)
- *Смена регистра* (страница 208)
  - *capitalize* (страница 208)
  - *lowercase* (страница 208)
  - *titlecase* (страница 208)
  - *uppercase* (страница 208)
- *Форматирование значений* (страница 208)
  - *finish_formatting* (страница 208)
  - *format_date* (страница 209)
  - *format_date_field* (страница 210)
  - *format_number* (страница 210)
  - *human_readable* (страница 210)
  - *rating_to_stars* (страница 210)
- *Функции URL* (страница 210)
  - *encode_for_url* (страница 210)
  - *make_url* (страница 211)
  - *make_url_extended* (страница 211)
  - *query_string* (страница 212)
  - *to_hex* (страница 213)
  - *urls_from_identifiers* (страница 213)
- *Функции базы данных* (страница 213)
  - *book_count* (страница 213)
  - *book_values* (страница 214)
  - *extra_file_modtime* (страница 214)
  - *extra_file_names* (страница 214)
  - *extra_file_size* (страница 214)
  - *get_link* (страница 215)
  - *get_note* (страница 215)

  - *has_extra_files* (страница 215)
  - *has_note* (страница 215)
- *Функции даты* (страница 216)
  - *date_arithmetic* (страница 216)
  - *days_between* (страница 216)
  - *today* (страница 216)
- *API of the Metadata objects* (страница 216)

Арифметический

add

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

`add(x [, y]*)` – возвращает сумму своих аргументов. Выдаёт исключение, если аргумент не является числом. В большинстве случаев эту функцию может заменить оператор `+`.

ceiling

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

`ceiling(value)` – возвращает наименьшее целое число, большее или равное `value`. Выдаёт исключение, если `value` не является числом.

divide

class calibre.utils.formatter_functions.**BuiltinDivide**

`divide(x, y)` – возвращает `x / y`. Выдает исключение, если `x` или `y` не являются числами. Эту функцию часто можно заменить оператором `/`.

floor

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

`floor(value)` – возвращает наибольшее целое число, меньшее или равное `value`. Выдаёт исключение, если `value` не является числом.

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.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «``» for CODE_TEXT on line 4 in «fractional_part»

mod

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

`mod(value, y)` – возвращает `floor` остаток от `value / y`. Выдаёт исключение, если `value` или `y` не является числом.

multiply

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

`multiply(x [, y]*)` – возвращает произведение своих аргументов. Выдаёт исключение, если какой-либо аргумент не является числом. Эту функцию часто можно заменить оператором `*`.

round

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

`round(value)` – возвращает ближайшее к `value` целое число. Выдаёт исключение, если `value` не является числом.

subtract

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

`subtract(x, y)` – возвращает `x - y`. Выдаёт исключение, если `x` или `y` не являются числами. Эту функцию часто можно заменить оператором `-`.

Булево

and

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

`and(value [, value]*)` – возвращает строку «1», если все значения не пустые, в противном случае возвращает пустую строку. Значения - сколько захотите. Обычно можно заменить на оператор `&&`. Одна из причин не заменять `and` на `&&` заключается в том, что короткое замыкание может изменить результаты из-за побочных эффектов. Например, `and(a='',b=5)` всегда будет выполнять оба присваивания, а оператор `&&` не будет выполнять второе.

not

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

`not(value)` – возвращает строку `1`, если значение пустое, иначе возвращает пустую строку. Эту функцию обычно можно заменить унарным оператором not (`!`).

or

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

`or(value [, value]*)` – возвращает строку «1», если все значения не пустые, в противном случае возвращает пустую строку. Значения - сколько захотите. Во многих случаях оператор `||` может заменить эту функцию. Причина, по которой нельзя заменить - короткое замыкание изменит результаты из-за побочных эффектов.

Другое

arguments

class calibre.utils.formatter_functions.**BuiltinArguments**

`arguments(id[=expression] [, id [=expression]]*)` – используется в сохранённом шаблоне для извлечения аргументов, переданных в вызове. Как объявляет так и инициализирует локальные переменные, фактически параметры. Переменные позиционные; они получают значение, указанное в вызове, в той же позиции. Если соответствующий параметр не указан в вызове, то аргументы присваивают этой переменной заданное значение по умолчанию. Если значение по умолчанию отсутствует, переменная устанавливается в пустую строку.

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.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «``» for CODE_TEXT on line 4 in «assign»

globals

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

`globals(id[=expression] [, id[=expression]]*)` – извлекает «глобальные переменные», которые могут быть переданы в форматтер. Имя *id* - это имя глобальной переменной. Оно одновременно объявляет и инициализирует локальные переменные передаваемыми именами (в параметрах `id` глобальных переменных. Если соответствующая переменная не представлена в глобальных, то этой переменной присваивается указанное по умолчанию значение. Если нет значения по умолчанию, то эта переменная принимает значение пустой строки.)

is_dark_mode

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

`is_dark_mode()` – вернуть `'1'` если calibre запущен в тёмном режиме, иначе `''` (пустая строка). Эту функцию можно использовать в расширенных настройках цвета и правилах выбора значков для выбора различных цветов / значков в зависимости от режима. Пример:

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

print

class calibre.utils.formatter_functions.**BuiltinPrint**

`print(a [, b]*)` – выводит аргументы на стандартный вывод. Пока вы не запустите calibre из командной строки (`calibre-debug -g`), вывод идёт в чёрную дыру (black hole). Функция `print` всегда возвращает свой первый аргумент.

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.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «``» for CODE_TEXT on line 5 in «set_globals»

Манипулирование строками

character

class calibre.utils.formatter_functions.**BuiltinCharacter**

`character(character_name)` - возвращает символ, названный character_name. Например, `character('newline)` возвращает символ новой строки (`'\n'`). Поддерживаемые имена символов: `newline`, `return`, `tab`, и `backslash`. Эта функция используется для ввода этих символов в выходные данные шаблонов.

check_yes_no

class calibre.utils.formatter_functions.**BuiltinCheckYesNo**

`check_yes_no(field_name, is_undefined, is_false, is_true)` – проверяет значение поля yes/no, названного ключом поиска `field_name`, на значение, указанное в параметрах, возвращая `'Yes'`, если совпадение найдено, иначе возвращает пустую строку. Установите параметр `is_undefined`, `is_false` или `is_true` в 1 (число), чтобы проверить это условие, иначе установите его равным 0 (ноль). Пример: `check_yes_no("#bool", 1, 0, 1)` возвращает `'Yes'`, если yes/no поле `#bool` не определено (ни True, ни False), или True.

Более одного из `is_undefined`, `is_false` или `is_true` может быть установлено в 1. Эта функция обычно используется в GUI и контент-сервере.

contains

class calibre.utils.formatter_functions.**BuiltinContains**

`contains(value, pattern, text_if_match, text_if_not_match)` – проверяет, соответствует ли значение регулярному выражению `pattern`. Возвращает `text_if_match` при совпадении, иначе возвращает `text_if_not_match`.

field_exists

class calibre.utils.formatter_functions.**BuiltinFieldExists**

`field_exists(field_name)` – проверяет, существует ли поле (столбец) с именем `field_name`, возвращая `'1'`, если это так, и пустую строку, если нет.

ifempty

class calibre.utils.formatter_functions.**BuiltinIfempty**

`ifempty(value, text_if_empty)` – если `value` не пустое, то вернуть это `value`, иначе вернуть `text_if_empty`.

re

class calibre.utils.formatter_functions.**BuiltinRe**

`re(value, pattern, replacement)` – вернуть значение `value` после применения регулярного выражения. Все вхождения `pattern` в значении `value` заменяются на `replacement`. Язык шаблонов не регистрозависим Python regular expressions[65].

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[66], 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* )(.*)", "{$:uppercase()}", "{$}")'}
```

Отображение документации на английском языке из-за ошибки FFML: Missing closing [/URL] near text программн on line 3 in «re_group»

shorten

class calibre.utils.formatter_functions.**BuiltinShorten**

`shorten(value, left chars, middle text, right chars)` – Возвращает сокращённую версию значения``value``, состоящую из символов `left chars` с начала значения, за которыми следует `middle text`, за которыми следуют символы `right chars` с конца значения. `left chars` и `right chars` должны быть неотрицательными целыми числами.

Пример: предположим, что вы хотите отобразить заголовок длиной не более 15 символов. Один из шаблонов, который делает это, - `{title:short(9,-,5)}`. Для книги с названием «Ancient English Laws in the Times of Ivanhoe» результатом будет `Ancient E-nhoe`: первые 9 символов названия, -, затем последние 5 символов. Если длина значения меньше, чем `left chars` + `right chars` + длина `middle text`, то значение будет возвращено без изменений. Например, название *The Dome* не изменится.

strcat

class calibre.utils.formatter_functions.**BuiltinStrcat**

`strcat(a [, b]*)` – возвращает строку объединяющую все аргументов. Может принимать любое количество аргументов. В большинстве случаев эту функцию может заменить оператор `&` .

strcat_max

class calibre.utils.formatter_functions.**BuiltinStrcatMax**

`strcat_max(max, string1 [, prefix2, string2]*)` – Возвращает строку, образованную объединением аргументов. Возвращаемое значение инициализируется `string1`. Пары `prefix, string` добавляются в конец значения, если результирующая длина строки меньше, чем `max`. `string1` возвращается, даже если `string1` длиннее `max`. Вы можете передать сколько угодно пар `prefix, string`.

[65] https://docs.python.org/3/library/re.html
[66] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

strlen

class calibre.utils.formatter_functions.**BuiltinStrlen**

`strlen(value)` – Возвращает длину строки `value`.

substr

class calibre.utils.formatter_functions.**BuiltinSubstr**

`substr(value, start, end)` – возвращает подстроку - символы строки `value` от `start` до `end`. Первый символ строки `value` имеет номер ноль. Если `end` отрицательный - это сколько символов будет отсчитываться с конца строки (справа). Если `end` равен 0 (ноль), то это - последний символ строки. Например, `substr('12345', 1, 0)` вернёт `'2345'`, и `substr('12345', 1, -1)` вернёт `'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`.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «`» for CODE_TEXT on line 7 in «swap_around_articles»

swap_around_comma

class calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

`swap_around_comma(value)` – если значение `value` имеет вид `B, A`, возвращает `A B`. Это наиболее полезно для преобразования имён вида LN, FN (Ф, И О) к виду FN LN (И О Ф). Если значение не содержит запятой в `value`, то оно возвращается без изменений.

test

class calibre.utils.formatter_functions.**BuiltinTest**

`test(value, text_if_not_empty, text_if_empty)` – возвращает `text_if_not_empty` если значение не пусто, иначе возвращает `text_if_empty`.

transliterate

class calibre.utils.formatter_functions.**BuiltinTransliterate**

`transliterate(value)` – возвращает строку в латинском алфавите, которая по звучанию похожа на слова исходной строки. Например, если `value` было `Фёдор Михáйлович Достоевский`, то функция вернёт `Fiodor Mikhailovich Dostoievskii`.

Манипуляция списком

list_count

class calibre.utils.formatter_functions.**BuiltinCount**

`list_count(value, separator)` – интерпретирует значение как список элементов, разделенных символом `separator` и возвращает количество элементов в списке. В большинстве списков используется запятая в качестве разделителя, но `authors` используют амперсанд (&).

Примеры: `{tags:list_count(,)}`, `{authors:list_count(&)}`.

Псевдонимы: `count()`, `list_count()`

list_count_field

class calibre.utils.formatter_functions.**BuiltinFieldListCount**

`list_count_field(lookup_name)` – вернуть количество элементов в поле с именем для поиска `lookup_name`. Поле должно быть многозначным, например `authors` или `tags`, в противном случае функция выдаст ошибку. Эта функция намного быстрее, чем `list_count()`, потому что она работает непосредственно с данными calibre без предварительного преобразования их в строку. Пример: `list_count_field('tags')`.

list_count_matching

class calibre.utils.formatter_functions.**BuiltinListCountMatching**

`list_count_matching(value, pattern, separator)` – интерпретирует `value` как список элементов, разделённых `separator`, возвращая количество элементов в списке, которые соответствуют регулярному выражению `pattern`. Псевдонимы: `list_count_matching()`, `count_matching()`

list_difference

class calibre.utils.formatter_functions.**BuiltinListDifference**

`list_difference(list1, list2, separator)` – возвращает список, созданный удалением из списка `list1` любого элемента, найденного в списке `list2`, используя регистронезависимое сравнение. Элементы в списках `list1` и `list2` разделяются разделителем `separator`, так же как и элементы в возвращаемом списке.

list_equals

class calibre.utils.formatter_functions.**BuiltinListEquals**

`list_equals(list1, sep1, list2, sep2, yes_val, no_val)` – возвращает `yes_val` если `list1` и `list2` содержат одинаковые элементы, иначе возвращает `no_val`. Элементы определяются разделением каждого списка с использованием соответствующего символа разделителя (`sep1` или `sep2`). Порядок элементов в списке не важен. Сравнение регистронезависимое.

list_intersection

class calibre.utils.formatter_functions.**BuiltinListIntersection**

`list_intersection(list1, list2, separator)` – вернуть список, созданный удалением из `list1` любого элемента, не найденного в `list2`, с использованием сравнения без учета регистра. Элементы в `list1` и `list2` разделены разделителем `separator`, как и элементы в возвращаемом списке.

list_join

class calibre.utils.formatter_functions.**BuiltinListJoin**

list_join(c_separator, list1, separator1 [, list2, separator2]*) – возвращает список, созданный путем объединения элементов в исходных списках (list1 и т.д.) используя with_separator между элементами в результирующем списке. Элементы в каждом из исходных списков list[123...] разделён соответствующим разделителем separator[123 ...]. Список может содержать нулевые значения. Это может быть поле типа publisher (издатель), которое является однозначным, фактически списком из одного элемента. Дубликаты удаляются с помощью сравнения без учета регистра. Детали возвращены в том порядке, как они появляются в списке источников. Если элементы в списках, отличаются только регистром букв, то используется последнее. Все разделители может быть больше, чем один символ.

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', ',')
```

Вы можете использовать выражения для создания списка. Например, предположим, что вам нужны элементы для authors и #genre, но с изменением жанра на слово «Genre: « далее следует первая буква жанра, т.е. «Fiction» станет «Genre: F». Получится следующее:

```
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) – создать список, вначале разделив src_list на элементы с помощью символа разделителя separator. Для каждого элемента в списке проверить, соответствует ли он include_re. Если это так, добавить его в список для возврата. Если opt_replace не является пустой строкой, применить замену перед добавлением элемента в возвращаемый список.

list_re_group

class calibre.utils.formatter_functions.**BuiltinListReGroup**

list_re_group(src_list, separator, include_re, search_re [, template_for_group]*) – Как и list_re, за исключением того, что замены не являются обязательными. При замене используется re_group(item, search_re, template ...).

list_remove_duplicates

class calibre.utils.formatter_functions.**BuiltinListRemoveDuplicates**

list_remove_duplicates(list, separator) – вернуть список, созданный удалением повторяющихся элементов в list. Если элементы отличаются только регистром, возвращается последний. Элементы в списке list разделены разделителем separator, как и элементы в возвращаемом списке``list``.

list_sort

class calibre.utils.formatter_functions.**BuiltinListSort**

`list_sort(list, direction, separator)` – вернуть список, отсортированный с использованием лексической сортировки без учета регистра. Если `direction` равно нулю, список сортируется по возрастанию, иначе по убыванию. Элементы списка разделены разделителем `separator`, как и элементы в возвращаемом списке.

list_split

class calibre.utils.formatter_functions.**BuiltinListSplit**

`list_split(list_val, sep, id_prefix)` – разделяет `list_val` на отдельные значения с помощью `sep`, затем присваивает значения переменным с именем `id_prefix_N`, где N — позиция значения в списке. Первый элемент имеет позицию 0 (ноль). Функция возвращает последний элемент в списке.

Пример:

```
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)` – возвращает список, составленный путем слияния элементов в `list1` и `list2`, удаляя повторяющиеся элементы с помощью сравнения без учета регистра. Если элементы отличаются регистром, используется значение из списка `list1`. Элементы в списках `list1` и `list2` разделены разделителем``separator``, как и элементы в возвращаемом списке. Псевдонимы: `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)
```

Отображение документации на английском языке из-за ошибки FFML: Missing closing «`» for italics on line 9 in «range»

subitems

class calibre.utils.formatter_functions.**BuiltinSubitems**

`subitems(value, start_index, end_index)` – Эта функция разделяет списки на иерархические элементы, подобные тегам, такие как жанры. Она интерпретирует *значение* как разделенный запятыми список элементов, похожих на теги, где каждый элемент представляет собой список, разделенный точкой. Возвращает новый список, созданный путем извлечения из каждого элемента компонентов от `start_index` до `end_index`, затем результаты объединяются обратно. Дубликаты удаляются. Первый подпункт в списке, разделенном точкой, имеет нулевой индекс. Если индекс отрицательный, то он отсчитывается от конца списка. В качестве особого случая предполагается, что `end_index` , равный нулю, равен длине списка.

Примеры:

- 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)` – интерпретирует `value` как список значений, разделённых `separator`, возвращая новый список с началом в `start_index` с концом в `end_index`. Первый пункт - номер 0 (ноль). Если индекс - отрицательный, то его начало - от конца списка. В особых случаях end_index нуля считается длиной списка.

Примеры, использующие режим базового шаблона и предополагающие что столбец тегов (разделённый запятыми) содержит «A, B, C»:

- `{tags:sublist(0,1,\,)}` returns «A»
- `{tags:sublist(-1,0,\,)}` returns «C»
- `{tags:sublist(0,-1,\,)}` returns «A, B»

Относящийся

cmp

class calibre.utils.formatter_functions.**BuiltinCmp**

`cmp(value, y, lt, eq, gt)` – сравнивает `value` и `y` после преобразования обоих в числа. Возвращает `lt` если `value <# y`, `eq` если `value ==# y`, иначе `gt`. Во многих случаях операторы числового сравнения (`==#`, `<#`, `>#` и т. п.) могут заменить эту функцию.

first_matching_cmp

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

`first_matching_cmp(val, [cmp1, result1,]+, else_result)` – последовательно сравнивает `val < cmp`, возвращая ассоциированный `result` для первого успешного сравнения. Если сравнение не удалось, возвращает `else_result`. Пример:

```
i = 10;
first_matching_cmp(i,5,"small",10,"middle",15,"large","giant")
```

вернёт `"large"`. В том же примере с первым значением 16 возвращается `"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.)

Отображение документации на английском языке из-за ошибки FFML: Missing closing «`» for italics on line 3 in «strcmp»

strcmpcase

class calibre.utils.formatter_functions.**BuiltinStrcmpcase**

`strcmpcase(x, y, lt, eq, gt)` – выполняет сравнение строк `x` и `y` с учетом регистра. Возвращает `lt`, если `x < y`, `eq` если `x == y`, иначе `gt`.

Примечание. Это НЕ поведение по умолчанию, используемое calibre, например, в операторах лексического сравнения (==, >, < и т. д.). Эта функция может привести к неожиданным результатам, желательно использовать `strcmp()`, когда это возможно.

Перебор значений

first_non_empty

class calibre.utils.formatter_functions.**BuiltinFirstNonEmpty**

`first_non_empty(value [, value]*)` – возвращает первое непустое значение. Если все значения пусты, возвращается пустая строка. Вы можете иметь сколько угодно значений.

lookup

class calibre.utils.formatter_functions.**BuiltinLookup**

`lookup(value, [ pattern, key, ]* else_key)` – Шаблоны будут проверяться по значению `value` по порядку. Если шаблон совпадает, возвращается значение поля, названного `key`. Если ни один шаблон не совпадает, возвращается значение поля с именем `else_key`. Смотрите функцию *switch()* (страница 201).

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.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «`» for italics on line 5 in «switch»

switch_if

class calibre.utils.formatter_functions.**BuiltinSwitchIf**

`switch_if([test_expression, value_expression,]+ else_expression)` – для каждой пары `test_expression, value_expression` проверяет, является ли `test_expression` True (непустым), и если да, то возвращает результат `value_expression`. Если ни одно из `test_expression` не равно True, возвращается результат `else_expression`. У вас может быть столько пар `test_expression, value_expression`, сколько вы хотите.

Поиск по списку

identifier_in_list

class calibre.utils.formatter_functions.**BuiltinIdentifierInList**

`identifier_in_list(val, id_name [, found_val, not_found_val])` – рассматривает `val` как список идентификаторов, разделённый запятыми. Идентификатор имеет формат `id_name:value`. `id_name` параметр - id_name текст для поиска, или `id_name` или `id_name:regexp`. Первый случай соответствует, если есть какой-либо идентификатор, соответствующий этому id_name. Второй случай соответствует, если id_name соответствует идентификатору, а регулярное выражение соответствует значению идентификатора. Если указаны `found_val` и `not_found_val`, то, если есть совпадение, возвращается `found_val`, в противном случае возвращается `not_found_val`. Если `found_val` и `not_found_val` не указаны, то, если есть совпадение, возвращается пара `identfier:value`, иначе пустая строка (`''`).

list_contains

class calibre.utils.formatter_functions.**BuiltinInList**

`list_contains(value, separator, [ pattern, found_val, ]* not_found_val)` – интерпретирует `value` как список элементов, разделенных разделителем `separator`, проверяя `pattern` для каждого элемента в списке. Если `pattern` соответствует элементу, возвращает `found_val`, иначе `not_found_val`. Пара `pattern` и `found_value` могут повторяться столько раз, сколько требуется, что позволяет возвращать разные значения в зависимости от значения элемента. Шаблоны проверяются по порядку и возвращается первое совпадение.

Псевдонимы: in_list(), list_contains()

list_item

class calibre.utils.formatter_functions.**BuiltinListitem**

`list_item(value, index, separator)` – обрабатывает значение как список элементов, разделённых `separator`, возвращая элемент с номером index. Первый элемент имеет номер ноль. Последний элемент с номером индекса `-1` как *list_item(-1,separator)*. Если элемент не в списке, то возвращается пустое значение. Разделитель (separator) работает так же, как и в функции *count*, обычно запятая, но амперсанд для списков, вроде author.

select

class calibre.utils.formatter_functions.**BuiltinSelect**

`select(value, key)` – обрабатывает `value` как разделённый запятыми список элементов вида `id:id_value` (формат `identifier` calibre). Функция находит первую пару с совпадающим с `key` идентификатором, и возвращает соответствующее `id_value`. При отсутствии совпадений возвращается пустая строка.

str_in_list

class calibre.utils.formatter_functions.**BuiltinStrInList**

`str_in_list(separator, [ string, found_val, ]+ not_found_val)` – интерпретирует значение как список элементов, разделенных `separator`, затем сравнивает `string` с каждым значением в списке. `string` не является регулярным выражением. Если `string` равна любому элементу (без учета регистра), вернуть соответствующее значение `found_val`. Если `string` содержит разделители `separator`, она также рассматривается как список, и каждое подзначение проверяется. Пары `string` и `found_value` могут повторяться сколько угодно раз, что позволяет возвращать разные значения в зависимости от значения строки. Если ни одна из строк не совпадает, возвращается `not_found_value`. Строки проверяются по порядку. Возвращается первое совпадение.

Получить значения из метаданных

annotation_count

class calibre.utils.formatter_functions.**BuiltinAnnotationCount**

`annotation_count()` – вернуть общее количество аннотаций всех типов, прикрепленных к текущей книге. Эта функция работает только в графическом интерфейсе и контент-сервере.

approximate_formats

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

`approximate_formats()` – возвращает разделенный запятыми список форматов, связанных с книгой. Поскольку список берется из базы данных calibre, а не из файловой системы, нет гарантии, что список правильный, но обычно это верно. Обратите внимание, что результирующие названия форматов всегда написаны заглавными буквами, как в EPUB.

Эта функция работает только в графическом интерфейсе. Если нужно использовать эти значения в шаблоне для сохранения на диск или отправки на устройство, то создайте пользовательский столбец типа «Столбец состоящий из других столбцов» и используйте функцию в нём, а затем используйте значение этого столбца в ваших шаблонах сохранения/отправки.

author_links

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

`author_links(val_separator, pair_separator)` – возвращает строку, содержащую список авторов и значения ссылок на этих авторов в форме: `author1 val_separator author1_link pair_separator author2 val_separator author2_link` и т.д.

Автор отделяется от значения ссылки строкой `val_separator` без добавленных пробелов. Предполагая, что `val_separator` является двоеточием, пары `author:link value` разделяются строковым аргументом `pair_separator` без добавленных пробелов. Выберите разделители, которые не встречаются в именах авторов или ссылках. Автор будет указан, даже если ссылка на автора пуста.

author_sorts

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

`author_sorts(val_separator)` – возвращает строку, содержащую список значений сортировок автора для авторов книги. Сортировка - метаданные автора (отличающиеся от author_sort в книгах). Возвращаемый список имет формат `author_sort1 val_separator author_sort2` и т.д. без добавляемых пробелов. Значения author_sort в этом списке в том же порядке, как и авторы в книге. Если вам нужны пробелы вокруг разделителя `val_separator`, включите их в строку `val_separator`.

booksize

class calibre.utils.formatter_functions.**BuiltinBooksize**

`booksize()` — возвращает значение поля `size`. Возвращает „“, если книга не имеет форматов.

Эта функция работает только в графическом интерфейсе. Если вы желаете использовать её значение в шаблоне для сохранения на диск или отправки на устройство, то создайте пользовательский столбец типа *Столбец, состоящий из других столбцов* и используйте функцию в нём, а затем используйте значение этого столбца в ваших шаблонах сохранения/отправки.

connected_device_name

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceName**

`connected_device_name(storage_location_key)` – если устройство подключено, вернуть имя устройства, в противном случае вернуть пустую строку. Каждое место хранения на устройстве может иметь другое имя. Названия `storage_location_key` - `'main'`, `'carda'` и `'cardb'`. Эта функция работает только в графическом интерфейсе.

connected_device_uuid

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceUUID**

`connected_device_uuid(storage_location_key)` – если устройство подключено, вернуть uuid устройства (уникальный идентификатор), иначе вернуть пустую строку. У каждого места хранения на устройстве свой uuid. Названия локаций - `'main'`, `'carda'` и `'cardb'`. Эта функция работает только в графическом интерфейсе.

current_library_name

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**

`current_library_name()` – возвращает последнее имя в пути к текущей библиотеке calibre.

current_library_path

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryPath**

`current_library_path()` – возвращает полный путь к текущей библиотеке calibre.

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.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «`» for italics on line 4 in «current_virtual_library_name»

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`.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «``» for CODE_TEXT on line 3 in «field»

formats_modtimes

class calibre.utils.formatter_functions.**BuiltinFormatsModtimes**

`formats_modtimes(date_format_string)` – возвращает разделённый запятыми список разделённых двоеточиями элементов `FMT:DATE`, представляющих собой время изменения форматов книги. Параметр `date_format_string` задаёт формат даты. Смотрите функцию *format_date()* (страница 209) для подробностей. Вы можете использовать функцию *select()* (страница 203) для получения времени модификации в заданном формате. Названия форматов всегда в верхнем регистре, как в EPUB.

formats_paths

class calibre.utils.formatter_functions.**BuiltinFormatsPaths**

`formats_paths()` – возвращает разделённый запятыми список элементов, разделённых двоеточиями``FMT:PATH``, представляющих собой полный путь к форматам книги. Вы можете использовать функцию `select()` для получения пути для заданного формата. Названия форматов всегда в верхнем регистре, как в EPUB.

formats_sizes

class calibre.utils.formatter_functions.**BuiltinFormatsSizes**

`formats_sizes()` – возвращает разделённый запятыми список элементов, разделённых двоеточиями``FMT:SIZE``, представляющих собой размеры в байтах форматов книги. Вы можете использовать функцию `select()` для получения размера для заданного формата. Названия форматов всегда в верхнем регистре, как в EPUB.

has_cover

class calibre.utils.formatter_functions.**BuiltinHasCover**

`has_cover()` – возвращает `'Yes'`, если у книги есть обложка, иначе возвращает пустую строку.

is_marked

class calibre.utils.formatter_functions.**BuiltinIsMarked**

`is_marked()` – проверить, помечена ли (*marked*) книга в calibre. Если это так, то вернуть значение метки, либо `'true'``(lower case), либо список именованных меток, разделенных запятыми. Возврат ``''`, если книга не отмечена. Эта функция работает только в графическом интерфейсе.

language_codes

class calibre.utils.formatter_functions.**BuiltinLanguageCodes**

`language_codes(lang_strings)` – возвращает коды языка language codes[67] для строк, взятых из *lang_strings*. Строки должны быть на языке текущей локализации. `lang_strings` - это список, разделённый запятыми.

language_strings

class calibre.utils.formatter_functions.**BuiltinLanguageStrings**

`language_strings(value, localize)` – вернуть значение названий языков для языковых кодов (см. здесь имена и коды[68]) , переданных в качестве значения `value`. Пример: `{languages:language_strings()}`. Если `localize` равно нулю, вернуть строки на английском языке. Если `localize` не равно нулю, вернуть строки на языке текущей локали. `lang_codes` - это список, разделённый запятыми.

ondevice

class calibre.utils.formatter_functions.**BuiltinOndevice**

`ondevice()` – возвращает строку `'Yes'` если параметр `ondevice` установлен, иначе возвращает пустую строку. Эта функция работает только в графическом интерфейсе. Если вы желаете использовать её значение в шаблоне для сохранения на диск или отправки на устройство, то создайте пользовательский столбец типа *Столбец, состоящий из других столбцов* и используйте функцию в нём, а затем используйте значение этого столбца в ваших шаблонах сохранения/отправки.

raw_field

class calibre.utils.formatter_functions.**BuiltinRawField**

`raw_field(lookup_name [, optional_default])` – возвращает поле метаданных с именем `lookup_name` без применения форматирования. Вычисляет и возвращает необязательный второй аргумент `optional_default`, если значение поля не определено (`None`). Префикс `$$` может использоваться вместо функции, как в `$$pubdate`.

raw_list

class calibre.utils.formatter_functions.**BuiltinRawList**

`raw_list(lookup_name, separator)` – возвращает список метаданных с именем `lookup_name` без применения какого-либо форматирования или сортировки и с элементами, разделёнными разделителем `separator`.

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

series_sort

class calibre.utils.formatter_functions.**BuiltinSeriesSort**

`series_sort()` – возвращает сортировочное значение серии.

user_categories

class calibre.utils.formatter_functions.**BuiltinUserCategories**

`user_categories()` – возвращает разделённый запятыми список категорий пользователя, которые содержат эту книгу. Эта функция работает только в GUI. Если вы хотите использовать эти значения в шаблонах save-to-disk или send-to-device, вы должны создать пользовательский «Столбец, состоящий из других столбцов», использовать функцию в шаблоне этого столбца и использовать значение этого столбца в ваших шаблонах сохранения/отправки

virtual_libraries

class calibre.utils.formatter_functions.**BuiltinVirtualLibraries**

virtual_libraries() – возвращает разделённый запятыми список виртуальных библиотек, которые содержат эту книгу. Эта функция работает только в графическом интерфейсе. Если вы желаете использовать её значение в шаблоне для сохранения на диск или отправки на устройство, то нужно создать пользовательский столбец типа «Столбец, состоящий из других столбцов» и использовать функцию в нём, а затем использовать значение этого столбца в своих шаблонах сохранения/отправки. Эта функция работает только в графическом интерфейсе.

Рекурсия

eval

class calibre.utils.formatter_functions.**BuiltinEval**

`eval(string)` – оценивает строку как программу, передавая локальные переменные. Это позволяет использовать шаблонный процессор для построения сложных результатов из локальных переменных. В Template Program Mode[69], поскольку символы `{` и `}` интерпретируются до вычисления шаблона, вы должны использовать``[[`` для символа `{` и `]]` для `}` . Они конвертируются автоматически. Также обратите внимание, что префиксы и суффиксы (синтаксис `|prefix|suffix`) не могут использоваться в аргументе этой функции при использовании Template Program Mode.

template

class calibre.utils.formatter_functions.**BuiltinTemplate**

`template(x)` – обрабатывает `x` как шаблон. Сравнение выполняется в контексте, то есть переменные не передаются между вызовом и обработкой шаблона. Если не используется General Program Mode, поскольку символы `{` и `}` являются специальными, вы должны использовать `[[` для символа `{` и `]]` для символа `}`; они преобразуются автоматически. Например,``template('[[title_sort]]')`` вычислит шаблон `{title_sort}` и вернёт его значение. Обратите также внимание, что префиксы и суффиксы (синтаксис `|prefix |suffix`) не могут использоваться в аргументе этой функции при использовании template program mode.

[69] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

Смена регистра

capitalize

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

`capitalize(value)` – возвращает значение, в котором первая буква в верхнем регистре, а остальные в нижнем регистре.

lowercase

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

`lowercase(value)` – возвращает `value` в нижнем регистре.

titlecase

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

`titlecase(value)` – возвращает `value` в регистре ЗАГОЛОВКОВ.

uppercase

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

`uppercase(value)` – возвращает `value` в верхнем регистре

Форматирование значений

finish_formatting

class calibre.utils.formatter_functions.**BuiltinFinishFormatting**

`finish_formatting(value, format, prefix, suffix)` – применяет `format`, `prefix` и `suffix` к `value` так же как в шаблонах `finish_formatting(value, format, prefix, suffix) {series_index:05.2f|- |- }`. Эта функция предусмотрена для упрощения преобразования сложных шаблонов с одной функцией или шаблонного программного режима в шаблоны *GPM*. Например, следующая программа выдает тот же результат, что и вышеупомянутый шаблон:

```
program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")
```

Другой пример: для шаблона:

```
{series:re(([^\s])[^\s]+(\s|$),\1)}{series_index:0>2s| - | - }{title}
```

используйте:

```
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)` – форматирует `value`, которое должно быть строкой даты, используя `format_string`, возвращающую строку. Лучше всего, если дата указана в формате ISO, поскольку использование других форматов даты часто приводит к ошибкам, поскольку фактическое значение даты не может быть определено однозначно. Обратите внимание, что функция `format_date_field()` работает быстрее и надёжнее.

Коды форматирования:

- `d` : день как число без начального нуля (1 до 31)
- `dd` : день как число с начальным нулём (01 до 31)
- `ddd` : локализованная аббревиатура названия дня (Напр. «Пн» до «Вскр»)
- `dddd` : полное локализованное название дня недели (напр. «Понедельник» до «Воскресенье»)
- `M` : номер месяца в году без начального нуля (1 to 12)
- `MM` : номер месяца в году с начальным нулём (01 to 12)
- `MMM` : сокращённое локализованное название месяца (напр. «Янв» до «Дек»)
- `MMMM` : полное локализованное название месяца (от «Январь» до «Декабрь»)
- `yy` : две последние цифры года (от 00 до 99
- `yyyy` : полный номер года из четырёх цифр
- `h` : часы без начального 0 (от 0 до 11 или от 0 до 23, в зависимости от am/pm (12/24) формата времени
- `hh` : часы с начальным 0 (от 00 до 11 или от 00 до 23, в зависимости от am/pm)
- `m` : минуты без начального 0 (от 0 до 59)
- `mm` : минуты с начальным 0 (00 to 59)
- `s` : секунды без начального 0 (0 to 59)
- `ss` : секунды с начальным 0 (00 to 59)
- `ap` : использовать 12-часовой формат вместо 24-часового, с «ap» заменённым на локализованную строку для am или pm
- `AP` : использовать 12-часовой формат вместо 24-часового, с «AP» заменённым на локализованную строку для AM или PM
- `aP` : использовать 12-часовой формат вместо 24-часового, с «aP» заменённым на локализованную строку для AM или PM
- `Ap` : использовать 12-часовой формат вместо 24-часового, с «Ap» заменённым на локализованную строку для AM или PM
- `iso` : дата, время и временная зона (должен быть единственным форматом)
- `to_number` : преобразовать дату & время в число с плаввающей запятой (*timestamp*)
- `from_number` : преобразуйте число с плавающей запятой (*timestamp*) в дату в формате ISO. Если вам нужен другой формат даты, добавьте желаемую строку форматирования после `from_number` и двоеточия (`:`). Пример:

```
format_date(val, 'from_number:MMM dd yyyy')
```

Вы можете получить неожиданные результаты, если форматируемая дата содержит локализованные названия месяцев, что может произойти, если вы изменили формат даты на содержащий``MMMM``. Используйте `format_date_field()` для предотвращения этой проблемы.

format_date_field

class calibre.utils.formatter_functions.**BuiltinFormatDateField**

`format_date_field(field_name, format_string)` – форматирует значение в поле `field_name`, которое должно быть искомым именем поля даты, стандартным или настраиваемым. См. *format_date()* (страница 209) для кодов форматирования. Эта функция намного быстрее, чем format_date(), и её следует использовать при форматировании значения в поле (столбце). Её нельзя использовать для вычисляемых дат или дат в строковых переменных. Примеры:

```
format_date_field('pubdate', 'yyyy.MM.dd')
format_date_field('#date_read', 'MMM dd, yyyy')
```

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[70] and the Python[71] documentation for more examples. Returns the empty string if formatting fails.

Отображение документации на английском языке из-за ошибки FFML: Missing closing «``» for CODE_TEXT on line 7 in «format_number»

human_readable

class calibre.utils.formatter_functions.**BuiltinHumanReadable**

`human_readable(value)` – ожидает, что значение будет числом, и возвращает строку, представляющую это число в КБ, МБ, ГБ и т. д.

rating_to_stars

class calibre.utils.formatter_functions.**BuiltinRatingToStars**

`rating_to_stars(value, use_half_stars)` – Возвращает рейтинг как `value` в виде строки, состоящей из звездочек (★“). Значение должно быть числом от `0` до `5`. Задайте для `use_half_stars` значение `1`, если вы хотите, чтобы символы половинной звезды для дробных чисел были доступны в настраиваемых столбцах рейтингов.

Функции URL

encode_for_url

[70] https://manual.calibre-ebook.com/template_lang.html
[71] https://docs.python.org/3/library/string.html#formatstrings

class calibre.utils.formatter_functions.**BuiltinEncodeForURL**

`encode_for_url(value, use_plus)` – возвращает значение `value` закодированное для использования в URL, как указанно с `use_plus`. Значение URL-кодированное. Далее, если `use_plus` равно `0`, то пробелы заменяются знаками `'+'` (плюс). Если это `1`, то пробелы заменяются `%20`.

Если вы не хотите, чтобы значение было кодированным, но чтобы пробелы были заменены, используйте функцию *re()* (страница 195) , как в `re ($ series, ' ', '%20')`

Смотрите также функции *make_url()* (страница 211), *make_url_extended()* (страница 211) и *query_string()* (страница 212).

make_url

class calibre.utils.formatter_functions.**BuiltinMakeUrl**

`make_url(path, [query_name, query_value]+)` – эта функция - простейший способ создания URL-адреса запроса. Она использует `path`, веб-сайт и страницу, которые вы хотите запросить и пары `query_name`, `query_value`, из которых строится запрос. Как правило, `query_value` должно быть закодировано в URL. С помощью этой функции оно всегда закодировано, пробелы всегда заменяются знаками „+“.

Должна быть предоставлена хотя бы одна пара `query_name`, `query_value`.

Пример: конструирование поискового URL-запроса в Wikipedia для автора *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
```

Если вы пишете пользовательский URL-шаблон сведений о книге столбцов, используйте `$item_name` или `field('item_name')` чтобы получить значение поля, на которое был нажат клик. Пример: если *Niccolò Machiavelli* было нажато затем можно сконструировать запрос URL:

```
make_url('https://en.wikipedia.org/w/index.php', 'search', $item_name)
```

См. функции *make_url_extended()* (страница 211), *query_string()* (страница 212) и *encode_for_url()* (страница 210).

make_url_extended

class calibre.utils.formatter_functions.**BuiltinMakeUrlExtended**

`make_url_extended(...)` – эта функция аналогична *make_url()* (страница 211) но дает вам больше контроля над компонентами URL. Компонентами URL являются

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

См. Uniform Resource Locator[72] в Wikipedia.

У функции два варианта:

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

и

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

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

Эта функция возвращает URL-адрес, созданный на основе `scheme`, `authority`, `path`, и либо `query_string` либо строки запроса, построенной из пар аргументов запроса. `authority` можен быть пустым, в случае `calibre` scheme URL. Вы должны указать либо `query_string`, либо хотя бы одну пару `query_name, query_value`. Если вы укажете `query_string` и оно будет пустым, то результирующий URL-адрес не будет содержать раздела строки запроса.

Пример 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
```

См. функцию *query_string()* (страница 212) для примера ипользующего `make_url_extended()` с `query_string`.

Если вы создаёте пользовательский шаблон URL-адреса сведений о столбцах книги, используйте `$item_name` или `field ('item_name')`, чтобы получить значение нажатого поля. Пример: если был нажат *Niccolò Machiavelli*, то вы можете создать URL-адрес, используя :

```
make_url_extended('https', 'en.wikipedia.org', '/w/index.php', 'search', $item_name')
```

См. функции *make_url()* (страница 211), *query_string()* (страница 212) and *encode_for_url()* (страница 210).

query_string

class calibre.utils.formatter_functions.**BuiltinQueryString**

`query_string([query_name, query_value, how_to_encode]+)` – возвращает строку запроса URL сконструированную из триад `query_name, query_value, how_to_encode`. Строка запроса это серия элементов выглядящих как `query_name=query_value`, где `query_value` это URL-кодирование. Элементы запроса разделяются символами `'&'` (ampersand).

Если `how_to_encode` это `0` затем `query_value` это кодированное и пробелы заменяются символом `'+'` (plus). Если `how_to_encode` это `1` затем `query_value` это кодированное с пробелами заменёнными на `%20`. Если `how_to_encode` это `2` затем `query_value` возвращается неизменным; нет кодирования и пробелы не заменяются. Если вы хотите чтобы `query_value` не кодировалась но пробелы заменялись используйте *re()* (страница 195) function, как в `re($series, ' ', '%20')`

Вы используете эту функцию, если вам нужен конкретный контроль над тем, как создаются части строки запроса . Затем вы могли бы использовать результирующую строку запроса в *make_url_extended()* (страница 211), 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
```

У вас должен быть хотя бы одна триада `query_name, query_value, how_to_encode`, но можете использовать их столько, сколько пожелаете.

Возвращаемое значение представляет собой строку запроса URL со всеми указанными элементами, например: `name1=val1[&nameN=valN]*`. Заметьте что `'?'` *path / query string* разделитель не включён в возвращаемый результат.

Если вы пишете пользовательский URL-шаблон сведений о столбцах книги, используйте `$item_name` or `field('item_name')` чтобы получить некодированное значение поля, по которому был произведен щелчок. У вас также есть `item_value_quoted`, где значение уже закодировано со знаками плюс замена пробелов и `item_value_no_plus`, где значение уже закодировано с заменой пробелов `%20`.

См. функции *make_url()* (страница 211), *make_url_extended()* (страница 211) и *encode_for_url()* (страница 210).

to_hex

class calibre.utils.formatter_functions.**BuiltinToHex**

`to_hex(val)` – возвращает строку `val`, закодированную в шестнадцатеричном формате. Полезно при создании URL-ов calibre.

urls_from_identifiers

class calibre.utils.formatter_functions.**BuiltinUrlsFromIdentifiers**

`urls_from_identifiers(identifiers, sort_results)` – задаёт список идентификаторов, разделённых запятыми, где `identifier` (идентификато)р представляет собой пару значений, разделённых двоеточиями (`id_name:id_value`), возвращает список URL-адресов HTML, разделённых запятыми, сгенерированных из идентификаторов. Список не отсортирован, если `sort_results` равен `0` (символ или число), в противном случае он сортируется в алфавитном порядке по имени идентификатора. URL-адреса генерируются так же, как столбец встроенных идентификаторов, когда они отображаются в Подробности книги.

Функции базы данных

book_count

class calibre.utils.formatter_functions.**BuiltinBookCount**

`book_count(query, use_vl)` – возвращает количество книг, найденных с помощью поиска для `query`. Если `use_vl` равно `0` (нулю), то виртуальные библиотеки игнорируются. Эта функция и её компаньон `book_values()` особенно полезны в поиске по шаблону, поддерживающем поиск, объединяющий информацию из многих книг, например, поиск серий только по одной книге. Его нельзя использовать в составных столбцах если для настройки `allow_template_database_functions_in_composites` не задано значение True. Можно использовать только в GUI. Например, этот поиск по шаблону использует эту функцию и её дополнение для поиска всех серий, содержащих только одну книгу:

- Определить хранимый шаблон (используя *Preferences → Advanced → Template functions*) названный `series_only_one_book` (имя - произвольное). Шаблон:

```
program:
vals = globals(vals='');
if !vals then
all_series = book_values('series', 'series:true', ',', 0);
for series in all_series:
if book_count('series:="' & series & '"', 0) == 1 then
vals = list_join(',', vals, ',', series, ',')
fi
rof;
set_globals(vals)
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
fi;
str_in_list(vals, ',', $series, 1, '')
```

При первом запуске шаблона (первая проверенная книга) он сохраняет результаты поиска в базе данных в `global` *глобальной* переменной шаблона с именем `vals`. Эти результаты используются для проверки последующих книг без повторного поиска.

- Использовать сохранённый шаблон при поиске по шаблону:

```
template:"program: series_only_one_book()#@#:n:1"
```

Использование сохранённого шаблона вместо помещения шаблона в поиск устраняет проблемы, вызванные требованием экранировать кавычки в поисковом выражении.

Функция используется только в GUI и контент-сервере.

book_values

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

`book_values(column, query, sep, use_vl)` – возвращает список уникальных значений, содержащихся в столбце `column` (имя поиска), разделенных `sep`, в найденных книгах путем поиска `query`. Если `use_vl` равно `0` (ноль), тогда виртуальные библиотеки игнорируются. Эта функция и её компаньон `book_count()` особенно полезны при поиске по шаблону, поддерживая поиск, объединяющий информацию из многих книг, например, поиск серий только с одной книгой. Нельзя использовать в составных столбцах, если для настройки `allow_template_database_functions_in_composites` установлено значение True. Можно использовать только в GUI и контент-сервере.

extra_file_modtime

class calibre.utils.formatter_functions.**BuiltinExtraFileModtime**

`extra_file_modtime(file_name, format_spec)` – возвращает время модификации дополнительного файла `file_name` в папке `data/` книги, если он существует, иначе `-1`. Modtime форматируется в соответствии с `format_spec` (см. *format_date()* (страница 209)). Если `format_string` пусто, возвращает modtime как количество секунд с плавающей запятой с начала эпохи. См. также функции *has_extra_files()* (страница 215), *extra_file_names()* (страница 214) и *extra_file_size()* (страница 214). Эпоха зависит от ОС. Эту функцию можно использовать только в GUI и контент-сервере.

extra_file_names

class calibre.utils.formatter_functions.**BuiltinExtraFileNames**

`extra_file_names(sep [, pattern])` – возвращает разделенный сепараторами (`sep`) список дополнительных файлов в папке книги `data/`. Если указан необязательный параметр `pattern`, регулярное выражение, то список фильтруется для файлов, соответствующих `pattern`. Сопоставление с `pattern` нечувствительно к регистру. См. также функции *has_extra_files()* (страница 215), *extra_file_modtime()* (страница 214) и *extra_file_size()* (страница 214). Эту функцию можно использовать только в GUI и контент-сервере.

extra_file_size

class calibre.utils.formatter_functions.**BuiltinExtraFileSize**

`extra_file_size(file_name)` – возвращает размер в байтах дополнительного файла `file_name` в папке `data/` книги, если он существует, иначе `-1`. См. также функции *has_extra_files()* (страница 215),

extra_file_names() (страница 214) и *extra_file_modtime()* (страница 214). Эту функцию можно использовать только в GUI и контент-сервере.

get_link

class calibre.utils.formatter_functions.**BuiltinGetLink**

`get_link(field_name, field_value)` – извлечь ссылку для поля `field_name` со значением `field_value`. Если прикрепленной ссылки нет, вернуть пустую строку. Примеры:

- Следующее возвращает ссылку, прикрепленную к тегу `Fiction`:

```
get_link('tags', 'Fiction')
```

- Этот шаблон создаёт список ссылок для всех тегов, связанных с книгой в форме `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
```

Эта функция работает только в графическом интерфейсе и на сервере контента.

get_note

class calibre.utils.formatter_functions.**BuiltinGetNote**

`get_note(field_name, field_value, plain_text)` – получить примечание для поля `field_name` со значением `field_value`. Если `plain_text` пустой, вернуть HTML примечания. Если „plain_text“ `1` (или `'1'`), вернуть простой текст примечания. Если примечание не существует, вернуть „“ (пустая строка) в обоих случаях. Пример:

has_extra_files

class calibre.utils.formatter_functions.**BuiltinHasExtraFiles**

`has_extra_files([pattern])` – возвращает количество дополнительных файлов, иначе „“ (пустая строка). Если указан необязательный параметр `pattern` (регулярное выражение), то список фильтруется по файлам, которые соответствуют `pattern`, прежде чем файлы будут подсчитаны. Сопоставление с `pattern` нечувствительно к регистру. См. также функции *extra_file_names()* (страница 214), *extra_file_size()* (страница 214) и *extra_file_modtime()* (страница 214). Можно использовать только в GUI и контент-сервере.

has_note

class calibre.utils.formatter_functions.**BuiltinHasNote**

`has_note(field_name, field_value)`. Проверьте, есть ли в поле примечание. Эта функция имеет два варианта:

- если `field_value` не `''` (пустая строка) вернуть `'1'` если значение `field_value` в поле `field_name` имеет примечание, иначе `''`. Если *field_value* равно `''`, то вернуть список значений в *field_name* у которых

есть примечание. Если ни у одного элемента в поле нет примечания, вернуть `''`. Этот вариант полезен для отображения значков столбцов, если какое-либо значение в поле имеет примечание, а не конкретное значение. Пример: `has_note('authors', '')` возвращает список авторов имеющих примечания или `''` если нет авторов с примечаниями.

Вы можете проверить, все ли значения в `field_name` имеют примечание, сравнив длину списка возвращаемого значения этой функции с длиной списка значений в `field_name`. Пример:

```
list_count(has_note('authors', ''), '&') ==# list_count_field('authors')
```

Можно использовать только в GUI и контент-сервере.

Функции даты

date_arithmetic

class calibre.utils.formatter_functions.**BuiltinDateArithmetic**

„“date_arithmetic(value, calc_spec, fmt)““ – вычислить новую дату от `value` с помощью `calc_spec`. Вернуть новую дату, отформатированную в соответствии с необязательным `fmt`: если не указано, то результат будет в формате ISO. Строка `calc_spec` - конкатенация пар „vW“ (valueWhat), где `v` - возможно отрицательное число, а W - одна из следующих букв:

- `s`: добавить `v` добавить к `date`
- `m`: добавить `v` минут к `date`
- `h`: добавить `v` часов к `date`
- `d`: добавить `v` дней к `date`
- `w`: добавить `v` недель к `date`
- `y`: добавить `v` лет к `date`, где год это 365 дней.

Пример: `'1s3d-1m'` добавит 1 секунду, прибавит 3 дня и вычтет 1 минуту из `date`.

days_between

class calibre.utils.formatter_functions.**BuiltinDaysBetween**

`days_between(date1, date2)` – возвращает количество дней между `date1` и `date2`. Число положительно, если `date1` больше `date2`, в противном случае отрицательно. Если `date1` или `date2` не являются датой, функция возвращает пустую строку.

today

class calibre.utils.formatter_functions.**BuiltinToday**

`today()` – возвращает текущий день в виде строки. Это значение предназначено для использования в `format_date` или `days_between`, но можно использовать как и любую другую строку. Дата в формате ISO[73]

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.

[73] https://en.wikipedia.org/wiki/ISO_8601

class calibre.ebooks.metadata.book.base.**Metadata**(*title*, *authors=('Неизвестный',)*, *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()* (страница 217) 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* (страница 218).

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((
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    '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
    # 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
))
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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
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 Всё об использования регулярных выражений в calibre

Регулярные выражения - это функции, используемые во многих местах в calibre для выполнения сложных манипуляций с содержимым электронной книги и метаданными. Этот учебник представляет собой краткое введение в использование регулярных выражений calibre.

Содержание

- *Во-первых, слово предупреждения и слово мужества* (страница 221)
- *Где в calibre вы можете использовать регулярные выражения?* (страница 222)
- *Что на земле является регулярным выражением?* (страница 222)
- *Хотите объяснений?* (страница 222)
- *Это звучит не так уж плохо. Что дальше?* (страница 222)
- *Эй, аккуратно! Это начинает иметь смысл!* (страница 223)
- *Ну, эти специальные символы очень аккуратные и все такое, но что, если я хочу сопоставить точку или знак вопроса?* (страница 223)
- *Итак, какие наборы наиболее полезны?* (страница 224)
- *Но если бы у меня было несколько переменных строк, которые я хотел бы сопоставить, все усложнялось?* (страница 224)
- *Вы пропустили…* (страница 225)
- *В начале вы сказали, что есть способ сделать регулярное выражение нечувствительным к регистру?* (страница 225)
- *Я думаю, что начинаю понимать эти регулярные выражения сейчас … как я могу использовать их в calibre?* (страница 225)
  - *Конвертации* (страница 225)
  - *Добавление книг* (страница 226)
  - *Массовое редактирование метаданных* (страница 226)
- *Краткая справка* (страница 227)
- *Признательность* (страница 232)

10.4.1 Во-первых, слово предупреждения и слово мужества

Это неизбежно будет чем-то техническим, ведь регулярные выражения - это технический инструмент для выполнения технических задач. Мне придется использовать жаргон и понятия, которые могут показаться сложными или запутанными. Я попытаюсь объяснить эти концепции настолько ясно, насколько смогу, но на самом деле не могу обойтись без их использования. Это, как говорится, не обескураживайся никаким жаргоном, так как я пытался объяснить все новое. И хотя регулярные выражения сами по себе могут показаться тайной, черной магией (или, если быть более прозаичным, случайной цепочкой букв и знаков мумбо-юмбо), я обещаю, что они не так уж и сложны. Даже те, кто действительно хорошо понимает регулярные выражения, испытывают затруднения при чтении более сложных, но написание их не так сложно - вы строите выражение шаг за шагом. Итак, сделайте шаг и следуйте за мной в кроличью нору.

10.4.2 Где в calibre вы можете использовать регулярные выражения?

Есть несколько мест, где calibre использует регулярные выражения. В опциях конвертации есть :guilabel:«Поиск и замена», обнаружение метаданных по именам файлов в настройках импорта и «Поиск и замена» при массовом редактировании метаданных книг. Редактор книг calibre также может использовать регулярные выражения при поиске и замене. Наконец, вы можете использовать регулярные выражения при поиске в списке книг и при поиске в просмотрщике электронных книг.

10.4.3 Что на земле *является* регулярным выражением?

Регулярное выражение - это способ описания наборов строк. Одно регулярное выражение может *соответствовать* ряду различных строк. Это то, что делает регулярные выражения такими мощными - они представляют собой краткий способ описания потенциально большого числа вариаций.

Примечание

Я использую здесь строку в том смысле, в каком она используется в языках программирования: строка из одного или нескольких символов, символов, включая действительные символы, цифры, знаки препинания и так называемые пробелы (переносы строк, табуляторы и т. д.). Обратите внимание, что обычно прописные и строчные буквы не считаются одинаковыми, поэтому «a» - это символ, отличный от «A» и т. д. В calibre регулярные выражения не чувствительны к регистру в строке поиска, но не в параметрах преобразования. Есть способ сделать каждое регулярное выражение нечувствительным к регистру, но мы обсудим это позже. Это усложняется тем, что регулярные выражения допускают изменения в соответствующих им строках, поэтому одно выражение может соответствовать нескольким строкам, поэтому люди вообще не хотят их использовать. Подробнее об этом чуть позже.

10.4.4 Хотите объяснений?

Ну, вот почему мы здесь. Во-первых, это самая важная концепция в регулярных выражениях: *Сама строка является регулярным выражением, которое соответствует самому себе*. То есть, если бы я хотел сопоставить строку `"Hello, World!"` С помощью регулярного выражения, используемое регулярное выражение было бы `Hello, World!`. И да, это действительно так просто. Однако заметьте, что это соответствует *только* точной строке `"Hello, World!"`, а не, например. `"Hello, wOrld!"` или ``»hello, world!» `` или любому другому подобному варианту.

10.4.5 Это звучит не так уж плохо. Что дальше?

Следующее - начало действительно хороших вещей. Помните, где я говорил, что регулярные выражения могут соответствовать нескольким строкам? Здесь все становится немного сложнее. Скажем, в качестве более практичного упражнения электронная книга, которую вы хотели преобразовать, имела неприятный нижний колонтитул, подсчитывающий страницы, например «Страница 5 из 423». Очевидно, что номер страницы увеличится с 1 до 423, поэтому вам нужно будет сопоставить 423 различных строки, верно? Неправильно, на самом деле: регулярные выражения позволяют вам определять наборы символов, которые соответствуют: Чтобы определить набор, вы помещаете все символы, в набор в квадратных скобках. Так, например, набор `[abc]` будет соответствовать символу «a», «b» или «c». * Наборы всегда будут соответствовать только одному из символов в наборе *****. Они «понимают» диапазоны символов, то есть, если вы хотите сопоставить все символы нижнего регистра, вы должны использовать набор `[az]` для символов нижнего и верхнего регистра, которые вы используете `[a-zA -Z]` и так далее. Есть идея? Таким образом, очевидно, что с помощью выражения `Page [0-9] of 423` вы сможете сопоставить первые 9 страниц, сократив количество выражений до трех: Второе выражение `Page [0-9] [0-9] of 423` будет соответствовать всем двузначным номерам страниц, и я уверен, что вы можете догадаться, как будет выглядеть третье выражение. Да, начинайте. Запишите это

10.4.6 Эй, аккуратно! Это начинает иметь смысл!

Я надеялся, что вы скажете это. Но приготовьтесь, теперь становится ещё лучше! Мы только что увидели, что используя наборы, мы можем сопоставить один из нескольких символов одновременно. Но вы даже можете повторить символ или набор, сократив число выражений, необходимых для обработки приведенного выше примера с номером страницы, до одного. Да, ОДИН! Вы в восторге? А должны быть! Это работает так: некоторые так называемые специальные символы «+», «?» и «*», *повторяют единственный элемент, предшествующий им*. (Элемент означает либо один символ, набор символов, escape-последовательность или группу (мы узнаем о последних двух позже), короче говоря, любой отдельный объект в регулярном выражении.) Эти символы называются подстановочными знаками или квантификаторами. Чтобы быть более точным, «?» соответствует *0 или 1* предыдущего элемента, «*» соответствует *0 или более* предыдущего элемента, а «+» соответствует *1 или более* предыдущего элемента. Несколько примеров: выражение `a?` будет соответствовать либо «» (это пустая строка, в данном случае не совсем полезная), либо «a», выражение `a*` будет соответствовать «», «a», «aa» или любому количеству a в строке, и, наконец, выражение `a+` будет соответствовать «a», «aa» или любому количеству a в строке (Примечание: это не будет соответствует пустой строке!). То же самое для множеств: выражение `[0-9]+` будет соответствовать *каждому целому числу*! Я знаю, о чем вы думаете, и вы правы: если вы используете это в приведенном выше случае совпадения номеров страниц, разве это не будет единственным выражением, которое будет соответствовать всем номерам страниц? Да, выражение `Page [0-9]+ of 423` будет соответствовать каждому номеру страницы в этой книге!

Примечание

Примечание к этим квантификаторам: они обычно пытаются найти как можно больше текста, поэтому будьте осторожны при их использовании. Это называется «жадным поведением» - я уверен, вы понимаете, почему. Это становится проблематичным, когда вы, скажем, пытаетесь сопоставить тег. Рассмотрим, например, строку `"<p class="calibre2">Title here</p>"` и, допустим, вы хотите сопоставить открывающий тег (часть между первой парой угловых скобок, чуть подробнее о тегах позже). Можно подумать, что выражение `<p.*>` Будет соответствовать этому тегу, но на самом деле оно соответствует всей строке! (Символ «.» является другим специальным символом. Он соответствует чему угодно *кроме* переносов строк, поэтому, по сути, выражение `.*` будет соответствовать любой отдельной строке, о которой вы можете подумать.) Вместо этого попробуйте использовать `<p.*?>`, что делает квантификатор `"*"` не жадным. Это выражение будет соответствовать только первому открывающему тегу, как и предполагалось. На самом деле есть еще один способ сделать это: выражение `<p[^>]*>` будет соответствовать тому же открывающему тегу - вы поймете почему после следующего раздела. Просто отметьте, что довольно часто существует несколько способов написания регулярного выражения.

10.4.7 Ну, эти специальные символы очень аккуратные и все такое, но что, если я хочу сопоставить точку или знак вопроса?

Конечно, вы можете сделать это: просто поставьте обратную косую черту перед любым специальным символом, и он будет интерпретирован как буквальный символ без какого-либо специального значения. Эта пара с обратной косой чертой, за которой следует один символ, называется escape-последовательностью, а действие по добавлению обратной косой черты перед специальным символом называется экранированием этого символа. Экранирующая последовательность интерпретируется как отдельный элемент. Конечно, существуют escape-последовательности, которые делают больше, чем просто экранирование специальных символов, например, `"\t"` означает табулятор. Мы вернемся к некоторым escape-последовательностям позже. Да, и, кстати, в отношении этих специальных символов: рассмотрите любой символ, который мы обсуждаем в этом введении, как имеющий некоторую функцию, которая должна быть особенной и, следовательно, должна быть экранирована, если вы хотите буквальный символ.

10.4.8 Итак, какие наборы наиболее полезны?

Знал, что ты спросишь. Некоторые полезные наборы: `[0-9]` соответствует одному числу, `[az]` соответствует одной строчной букве, `[AZ]` соответствует одной заглавной букве, `[a-zA-Z]` сопоставление одной буквы и `[a-zA-Z0-9]` сопоставление одной буквы или числа. Также можете использовать escape-последовательность в качестве сокращения:

\d
: эквивалентно `[0-9]`

\w
: эквивалентно `[a-zA-Z0-9_]`

\s
: эквивалентно любому пробелу

Примечание

«Пробел» (Whitespace) - это термин для всего, что не будет напечатано. Эти символы включают пробел (space), табулятор (tabulator), перевод строки (line feed), перевод формы (form feed), возврат каретки (carriage return), неразрывные пробелы (non-breaking spaces) и т.п.

Примечание

Наборы верхнего и нижнего регистра могут соответствовать как верхнему, так и нижнему регистру, если включена настройка, позволяющая сделать поиск нечувствительным к регистру. Такие настройки находятся, например, в Настройки-Поиск в самом calibre и на панели Поиска в calibre *Просмотрщик электронных книг*, а также в инструменте calibre *Редактировать книгу*.

В качестве последней заметки о наборах вы также можете определить набор как любой символ, *но* из набора. Вы делаете это путем включения символа `"^"` в качестве *самого первого символа в наборе*. Таким образом, `[^a]` будет соответствовать любому символу, кроме «a». Это называется дополнением набора. Те сокращенные последовательности escape-последовательностей, которые мы видели ранее, также могут быть дополнены: `"\D"` означает любой нечисловой символ, что эквивалентно `[^0-9]`. Другие сокращения можно дополнить, как вы уже догадались, используя соответствующую заглавную букву вместо строчной. Итак, возвращаясь к примеру `<p[^>]*>` из предыдущего раздела, теперь вы можете видеть, что используемый им набор символов пытается соответствовать любому символу, кроме закрывающей угловой скобки.

10.4.9 Но если бы у меня было несколько переменных строк, которые я хотел бы сопоставить, все усложнялось?

Не бойся, жизнь всё ещё хороша и легка. Рассмотрим пример: в книге, которую вы конвертируете, на каждой нечетной странице написано «Title», а на каждой четной странице - «Author». Отлично смотрится в печати, верно? Но в электронных книгах это раздражает. Вы можете сгруппировать целые выражения в нормальных скобках, и символ `"|"` позволит вам сопоставить *либо* выражение справа, *либо* слева. Объедините их, и всё готово. Слишком быстро для тебя? Хорошо, во-первых, мы группируем выражения для нечетных и четных страниц, получая, таким образом, `(Title)(Author)` как два наших необходимых выражения. Теперь мы упростили ситуацию с помощью вертикальной черты (`"|"` называется символом вертикальной черты): если вы используете выражение `(Title|Author)`, вы либо получите совпадение для «Title» (на нечетных страницах) или вы должны соответствовать «Author» (на четных страницах). Ну, разве не просто?

Конечно, вы можете использовать вертикальную черту, не используя групповые скобки. Помните, когда я говорил, что квантификаторы повторяют элемент, предшествующий им? Вертикальная черта работает немного подругому: выражение «Title|Author» также будет соответствовать либо строке «Title», либо строке «Author», как в

приведенном выше примере с использованием группировки. *Вертикальная черта выбирает во всём выражении, предшествующее и следующее за ней*. Итак, если вы хотите сопоставить строки «Calibre» и «calibre» и хотите выбирать только между прописными и строчными буквами «c», вам нужно использовать выражение `(c|C)alibre`, где группировка гарантирует, что будет выбрана только буква «c». Если бы вы использовали `c|Calibre`, вы бы получили совпадение в строке «c» или в строке «Calibre», что не то, что мы хотели. Вкратце: если есть сомнения, используйте группировку вместе с вертикальной чертой.

10.4.10 Вы пропустили...

… подождите минутку, есть еще одна действительно полезная вещь, которую вы можете сделать с группами. Если у вас есть группа, которую вы ранее сопоставляли, вы можете использовать ссылки на эту группу позже в выражении: Группы нумеруются, начиная с 1, и вы ссылаетесь на них, экранируя номер группы, на которую хотите сослаться, таким образом, пятая группа будет ссылаться как `\5`. Итак, если вы искали `([^]+)\1` в строке «Test Test», вы бы соответствовали всей строке!

10.4.11 В начале вы сказали, что есть способ сделать регулярное выражение нечувствительным к регистру?

Да, спасибо, что уделили мне внимание и напомнили. Вы можете сказать calibre, как вы хотите, чтобы определенные вещи обрабатывались с помощью чего-то, называемого флагами. Вы включаете флаги в свое выражение, используя специальную конструкцию `(?flags go here)`, где, очевидно, вы бы заменили «flags go here» на конкретные флаги, которые вы хотите. Для игнорирования регистра флаг `i`, таким образом, вы включаете `(?I)` в свое выражение. Таким образом, `(?I)test` будет соответствовать «Test», «tEst», «TEst» и любым вариантам, которые вы можете придумать.

Другой полезный флаг позволяет точке соответствовать любому символу, * включая* перевод строки, флаг `s`. Если вы хотите использовать несколько выражений в выражении, просто поместите их в один и тот же оператор: `(?is)` проигнорирует регистр и сделает точку подходящей для всех. Неважно, какой флаг вы указали первым, `(?si)` будет эквивалентно приведенному выше.

10.4.12 Я думаю, что начинаю понимать эти регулярные выражения сейчас ... как я могу использовать их в calibre?

Конвертации

Начнем с настроек конвертации, которые действительно аккуратны. В части *Search & replace* вы можете ввести регулярное выражение (сокращение от регулярного выражения), которое описывает строку, которая будет заменена во время преобразования. Аккуратная часть - это волшебник. Нажмите на посох волшебника, и вы получите предварительный просмотр того, что calibre «видит» в процессе конвертации. Прокрутите вниз до строки, которую хотите удалить, выберите и скопируйте её, вставьте в поле регулярных выражений в верхней части окна. Если есть переменные части, такие как номера страниц или вроде того, используйте наборы и квантификаторы, чтобы покрыть их, и пока вы это делаете, не забудьте экранировать специальные символы, если они есть. Нажмите кнопку с надписью *Test*, и calibre выделит части, которые он заменит, если вы будете использовать регулярное выражение. Как только вы удовлетворены, нажмите OK и конвертируйте. Будьте осторожны, если в вашем источнике конвертациииии есть теги, подобные следующему:

```
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
```

(бессовестно вырванный из этой темы <https://www.mobileread.com/forums/showthread.php?t=75594 «>`_). Вам также придется удалить некоторые теги. В этом примере , я бы рекомендовал начать с тэга `<b class="calibre2">`, теперь вы должны закончить соответствующим закрывающим тэгом (открывающие тэги - `<tag>`, закрывающие тэги - `</tag>`), который просто является следующим `` в этом случае. (Обратитесь к хорошему руководству по HTML или спросите на форуме, если вы не уверены в этом.) Открывающий тег можно описать с помощью``<b.*?>``, закрывающий тег с помощью `</b>`, таким образом, мы могли бы удалить все между этими тегами, используя `<b. *?>.*?</b>`. Но использование этого выражения было бы плохой идеей, потому что оно удаляет всё, что заключено в <b> -теги - (что, кстати, рендерит вложенный текст жирным шрифтом), и будет справедливо поспорить, что мы удалим части книги таким образом. Вместо этого включите также начало вложенной строки, создав регулярное выражение `<b.*?>\s*Generated\s+by\s+ABC\s+Amber\s+LIT.*?</b>`. `\s` с квантификаторами находятся в здесь вместо того, чтобы явно использовать пробелы, как видно в строке, чтобы поймать любые варианты строки, которые могут возникнуть. Не забудьте проверить, какой из них calibre удалит, чтобы убедиться, что вы не удаляете те части, которые хотите сохранить, если тестируете новое выражение. Если вы проверите только одно вхождение, вы можете пропустить несоответствие где-то ещё в тексте. Также обратите внимание, что, если вы случайно удалите больше или меньше тегов, чем вы на самом деле хотели, calibre попытается восстановить повреждённый код после удаления.

Добавление книг

Другая вещь, для которой вы можете использовать регулярные выражения, - это извлечение метаданных из имён файлов. Вы можете найти эту функцию в разделе «Добавление книг» в настройках. Здесь есть особенность: вы можете использовать имена полей для полей метаданных, например, `(?P<title>)` будет означать, что calibre использует эту часть строки в качестве названия книги. Разрешенные имена полей перечислены в окнах вместе с другим хорошим тестовым полем. Пример: скажем, вы хотите импортировать целую кучу файлов с такими именами, как `Classical Texts: The Divine Comedy by Dante Alighieri.mobi` (`Классические тексты: Божественная комедия Данте Алигьери.mobi`). (Очевидно, это уже есть в вашей библиотеке, так как мы все любим классическую итальянскую поэзию) или `Science Fiction epics: The Foundation Trilogy by Isaac Asimov.epub` (`Фантастические эпопеи: трилогия Фонда» Исаака Азимова.epub`). Это, очевидно, схема именования, из которой calibre не будет извлекать какие-либо значимые данные - стандартное выражение для извлечения метаданных - `(?P<title>.+)-(?P<author>[^_]+) ``. Регулярное выражение, которое работает здесь, будет ``[a-zA-Z]+:(?P<title>. +) by (?P<author>.+)`. Обратите внимание, что внутри группы для поля метаданных вам нужно использовать выражения для описания того, что поле действительно соответствует. Также обратите внимание, что при использовании тестового поля calibre вам необходимо добавить расширение файла к вашему тестовому имени файла, иначе вы не получите никаких совпадений, несмотря на использование рабочего выражения.

Массовое редактирование метаданных

Последняя часть - поиск и замена регулярных выражений в полях метаданных. Вы можете получить доступ к этому, выбрав несколько книг в библиотеке и используя массовое редактирование метаданных. Будьте очень осторожны при использовании этой последней функции, поскольку она может сделать **Очень плохие вещи** для вашей библиотеки! Дважды проверьте, что ваши выражения делают то, что вы хотите, используя тестовые поля, и отмечайте только те книги, которые вы действительно хотите изменить! В режиме поиска по регулярному выражению вы можете искать в одном поле, заменять текст чем-либо и даже записывать результат в другое поле. Практический пример: скажем, ваша библиотека содержала книги из серии Фрэнка Герберта «Дюны», названные после моды `Dune 1 - Dune`, `Dune 2 - Dune Messiah` и так далее. Теперь вы хотите получить `Dune` в поле серии. Вы можете сделать это, выполнив поиск `(.*?) \d+ - .*` в поле заголовка и заменив его на `\1` в поле серии. Видишь, что я там сделал? Это ссылка на первую группу, которой вы заменяете поле серии. Теперь, когда у вас есть все серии, вам нужно только выполнить другой поиск `.*?-` в поле заголовка и замените его на `""` (пустую строку), снова в поле заголовка, и ваши метаданные будут аккуратными и чистыми. Разве это не здорово? Кстати, вместо замены всего поля вы также можете добавить или добавить к нему поле, так что, если вы *хотите* добавить к названию книги информацию о серии, вы также можете это сделать. Как вы, несомненно, уже заметили, есть флажок с надписью *Чувствительный к регистру*, поэтому вам не придется использовать флаги для выбора поведения здесь.

Ну, вот и заканчивается очень краткое введение в регулярные выражения. Надеюсь, я покажу вам достаточно, чтобы, по крайней мере, начать работу и продолжить самостоятельное обучение - хорошей отправной точкой будет документация Python для регулярных выражений <https://docs.python.org/library /re.html>`_.

Последнее слово предостережения: регулярные выражения мощная вещь, но в них также легко ошибиться. calibre предоставляет отличные возможности тестирования, чтобы увидеть, ведут ли себя ваши выражения так, как вы ожидаете. Используйте их. Старайтесь не стрелять себе в ногу. (Боже, мне нравится это выражение …) Но если вы, несмотря на предупреждение, пораните ногу (или любые другие части тела), постарайтесь извлечь из этого уроки.

10.4.13 Краткая справка

Краткая справка о синтаксисе регулярных выражений

В этот список помещены наиболее используемые / трудные для запоминания части движка регулярных выражений, доступные в calibre почти везде.

Содержание

- *Классы символов* (страница 227)
- *Сокращенные классы символов* (страница 228)
- *Квантификаторы* (страница 228)
- *Жадность* (страница 228)
- *Перечисление* (страница 229)
- *Исключение* (страница 229)
- *Якоря* (страница 229)
- *Группы* (страница 229)
- *Взгляды по сторонам* (страница 230)
- *Рекурсия* (страница 230)
- *Специальные символы* (страница 231)
- *Мета-символы* (страница 231)
- *Режимы* (страница 231)

Классы символов

Классы символов полезны для краткого представления различных групп символов.

Примеры:

Обозначение	Класс
`[а-я]`	Строчные буквы. Не включает символы с ударением и лигатурами.
`[а-я0-9]`	Строчные буквы от а до я а также цифры от 0 до 9.
`[А-Яа-я-]`	Заглавные, строчные буквы, а также тире. Чтобы включить тире в классе, вы должны поместить его в начале или в конце, чтобы не путать его с дефисом, который обозначает диапазон символов.
`[^0-9]`	Любые символы, кроме цифр. Каретка (^), помещенная в начале класса, исключает символы класса (дополненный класс).
`[[a-z]--[aeiouy]]` `[[а-я]--[аеёиоуыэю]`	Строчные согласные. Класс может быть включен в класс. Символы `--` исключают то, что следует за ними
`[\w--[\d_]]`	Все буквы (включая иностранные акцентированные символы). Сокращенные классы могут использоваться внутри класса

Пример:

```
<[^<>]+> to select an HTML tag
```

Сокращенные классы символов

Обозначение	Класс
`\d`	Цифра (то же, что и `[0-9]`)
`\D`	Любой нецифровой символ (то же, что и `[^0-9]`)
`\w`	Буквенно-цифровой символ (`[a-zA-Z0-9]`), включая символы с меткой ударения и лигатуры
`\W`	Любой “non-word” («несловесный») символ
`\s`	Space (пробел), non-breaking space (неразрывный пробел), tab (табуляция), return line (обратная строка)
`\S`	Любой символ «без пробелов»
`.`	Любой символ кроме новой строки. Используйте флажок «dot all» или модификатор regexp `(?s)`, чтобы включить символ новой строки.

Квантификаторы

Квантификатор	Количество вхождений выражения, предшествующего квантификатору
`?`	0 или 1 повторение регулярного выражения. Эквивалентно `{0,1}`.
`+`	Одно или более повторение регулярного выражения. Эквивалентно `{1,}`.
`*`	Любое - 0, 1 или более - количество повторений регулярного выражения. Эквивалентно `{0,}`.
`{n}`	Ровно n повторений регулярного выражения.
`{min,max}`	Количество повторений регулярного выражения, не менее min раз и не более max раз.
`{min,}`	Не менее min повторений регулярного выражения.
`{,max}`	Количество вхождений между 0 и включенным максимальным значением

Жадность

По умолчанию в квантификаторах механизм регулярных выражений является жадным: он максимально расширяет выбор. Поначалу это часто вызывает сюрпризы. ``? `` следует за квантификатором, чтобы сделать его ленивым. Старайтесь не ставить два в одном выражении, результат может быть непредсказуемым.

Остерегайтесь вложенных квантификаторов, например, шаблона `(a*)*`, поскольку он экспоненциально увеличивает время обработки.

Перечисление

Символ `|` в регулярном выражении соответствует логическому `ИЛИ` (`OR`). Т. е. могут совпадать либо предыдущее, либо последующее выражения.

Исключение

Метод 1

```
шаблон_исключения(*SKIP)(*FAIL)|шаблон_для_поиска
```

Пример:

```
"Blabla"(*SKIP)(*FAIL)|Blabla
```

выбирает Blabla, в строках Blabla или «Blabla или Blabla», но не в «Blabla».

Метод 2

```
шаблон_для_исключения\K|(шаблон_для_отбора)
```

```
"Blabla"\K|(Blabla)
```

выбирает Blabla, в строках Blabla или «Blabla или Blabla», но не в «Blabla».

Якоря

Якорь - это способ сопоставления логического расположения в строке, а не символа. Наиболее полезными якорями для обработки текста являются:

\b
: Обозначает границу слова, то есть переход от пробела к непропробельному символу. Например, вы можете использовать `\bsurd` для сопоставления с `surd`, но не `absurd`.

^
: Соответствует началу строки (в многострочном режиме, который используется по умолчанию)

$
: Соответствует концу строки (в многострочном режиме, который используется по умолчанию)

\K
: Сбрасывает начальную позицию выделения в его позицию в шаблоне. Некоторые движки регулярных выражений (но не calibre) не позволяют просматривать данные переменной длины, особенно с квантификаторами. Когда вы можете использовать `\K` с этими движками, это также позволяет вам избавиться от этого предела, написав эквивалент положительного взгляда позади длины переменной.

Группы

(выражение)
: Группа захвата, которая хранит выбор и может быть вызвана позже в шаблонах *search* или *replace* на `\n`, где `` n`` - порядковый номер группы захвата (начиная с 1 в порядке чтения)

(?:выражение)
: Группа, которая не захватывает выбор

(?>expression)
: Атомная группа: как только выражение удовлетворено, обработчик регулярных выражений проходит, и если остальная часть шаблона завершится неудачно, он не будет возвращаться, чтобы попробовать другие комбинации с выражением. Атомные группы не захватывают.

(?|expression)
: Группа сброса ветвей: ветви чередований, включенные в выражение, имеют одинаковые номера групп

(?<name>expression)
: Группа под названием «имя». Выбор может быть вызван позже в шаблоне *search* с помощью `(?P=name)` и в *replace* на `\g<name>`. Две разные группы могут использовать одно и то же имя.

Взгляды по сторонам

Взгляд по сторонам	Смысл
`?=`	Позитивный lookahead (взгляд вперёд) (будет размещен после выделения)
`?!`	Негативный lookahead (взгляд вперёд) (ставится после выделения)
`?<=`	Позитивный lookbehind (взгляд назад) (ставится перед выделением)
`?<!`	Негативный lookbehind (взгляд назад) (ставится перед выделением)

Lookaheads (взгляды вперед) и lookbehinds (взгляды назад) не потребляют символы, они имеют нулевую длину и не захватывают. Они являются атомарными группами: как только утверждение выполнено, движок регулярных выражений проходит, и если не удастся остальная часть шаблона, он не будет возвращаться в обход, чтобы попробовать другие комбинации.

При поиске нескольких совпадений в строке, в начальной позиции каждой попытки сопоставления, взгляд назад может проверять символы перед текущей позицией. Поэтому в строке 123 шаблон `(?<=\D)\d` (цифра, перед которой стоит цифра) должен теоретически выбрать 2 и 3. С другой стороны, `\d\K\d` может выбрать только 2, потому что начальная позиция после первого выбора находится непосредственно перед 3, и для второго совпадения недостаточно цифр. Точно так же `\d(\d)` только захватывает 2. В практике движка регулярных выражений calibre положительный взгляд назад (positive lookbehind) ведёт себя таким же образом и выбирает только 2, вопреки теории.

Группы могут быть помещены в lookarounds (взгляды по сторонам), но захват редко полезен. Тем не менее, если это будет полезно, необходимо будет быть очень осторожным при использовании квантификатора во взгляде назад: жадность, связанная с отсутствием обратного отслеживания, может дать неожиданный захват. По этой причине, используйте `\K`, а не положительный взгляд назад (positive lookbehind), когда у вас есть квантификатор (или, что еще хуже, несколько) в группе захвата положительного взгляда назад (positive lookbehind).

Пример негативного взгляда вперёд:

```
(?![^<>{}]*[>}])
```

Помещенный в конец шаблона препятствует выделению внутри тега или стиля, встроенного в файл.

Когда бы ни было возможно, всегда лучше «закрепить» обходные пути, чтобы уменьшить количество шагов, необходимых для получения результата.

Рекурсия

Обозначение	Смысл
`(?R)`	Рекурсия всего шаблона
`(?1)`	Рекурсия единственного шаблона нумерованной группы захвата, здесь группа 1

Рекурсия вызывает себя. Это полезно для сбалансированных запросов, таких как строки в кавычках, которые могут содержать встроенные строки в кавычках. Таким образом, если во время обработки строки между двойными кавычками мы сталкиваемся с началом новой строки между двойными кавычками, мы хорошо знаем, как это сделать, и мы вызываем себя. Тогда у нас есть шаблон, вроде:

```
start-pattern(?>atomic sub-pattern|(?R))*end-pattern
```

Чтобы выбрать строку между двойными кавычками, не останавливаясь на встроенной строке:

```
“((?>[^“”]+|(?R))*[^“”]+)”
```

Этот шаблон также можно использовать для изменения пар тегов, которые могут быть встроены, таких как теги `<div>`.

Специальные символы

Обозначение	Символ
\t	табуляция
\n	Разрыв строки
\x20	(разрываемый) пробел
\xa0	неразрываемый пробел

Мета-символы

Мета-символы - это те, которые имеют особое значение для движка регулярных выражений. Из них двенадцати должен предшествовать escape-символ, обратная косая черта (\), чтобы потерять свое особое значение и снова стать обычным символом:

```
^ . [ ] $ ( ) * + ? | \
```

Семи другим метасимволам не обязательно должен предшествовать обратный слеш (но он может быть без каких-либо других последствий):

```
{ } ! < > = :
```

Специальные символы теряют свой статус, если они используются внутри класса (между скобками `[]`). Закрывающая скобка и тире имеют особый статус в классе. Вне класса тире - это простой литерал, закрывающая скобка остается метасимволом.

Косая черта (/) и знак числа (или хэш-символ) (#) не являются метасимволами, их не нужно экранировать.

В некоторых инструментах, таких как regex101.com с движком Python, двойные кавычки имеют специальный статус разделителя и должны быть экранированы, либо параметры изменены. Это не так в редакторе calibre.

Режимы

(?s)
: Заставляет точку (`.`) совпадать с символами новой строки

(?m)
: Заставляет якоря `^` и `$` соответствовать началу и концу строк вместо начала и конца всей строки.

10.4.14 Признательность

Спасибо за помощь с советами, исправлениями и всем таким:

- ldolse
- ovidgoyal
- chaley
- dwanthny
- kacir
- Starson17
- Orpheu

Для получения дополнительной информации о регулярных выражениях см. Руководство пользователя Python[74]. Актуальная библиотека регулярных выражений, используемая calibre: regex[75], поддерживает несколько полезных улучшений по сравнению со стандартной библиотекой Python.

10.5 Создание собственных плагинов для расширения функциональности calibre

calibre имеет очень модульную конструкцию. Практически все функциональные возможности calibre поставляются в виде плагинов. Плагины используются для преобразования, для загрузки новостей (там они называются рецептами), для различных компонентов пользовательского интерфейса, для подключения к различным устройствам, для обработки файлов при добавлении их в calibre и так далее. Вы можете получить полный список всех встроенных плагинов в calibre, перейдя к :guilabel: *Preferences-> Advanced-> Plugins*.

Здесь мы научим вас, как создавать свои собственные плагины чтобы добавить новые функции в calibre.

Содержание

- *Анатомия плагина calibre* (страница 233)
- *Плагин пользовательского интерфейса* (страница 234)
  - *__init__.py* (страница 235)
  - *ui.py* (страница 236)
  - *main.py* (страница 237)
  - *Получение ресурсов из плагина ZIP-файла* (страница 240)
  - *Включение пользовательской конфигурации вашего плагина* (страница 241)
- *Редактировать книжные плагины* (страница 243)
  - *main.py* (страница 243)
- *Добавление переводов в ваш плагин* (страница 246)
- *API плагина* (страница 247)
- *Отладка плагинов* (страница 247)

[74] https://docs.python.org/library/re.html
[75] https://bitbucket.org/mrabarnett/mrab-regex/src/hg/

- *Больше примеров плагинов* (страница 247)
- *Поделиться своими плагинами с другими* (страница 247)

Примечание

Это относится только к calibre releases >= 0.8.60

10.5.1 Анатомия плагина calibre

Плагин calibre очень прост, это просто ZIP-файл, который содержит некоторый код Python и любые другие ресурсы, такие как файлы изображений, необходимые плагину. Без дальнейших церемоний, давайте посмотрим на основной пример.

Предположим, у вас установлена calibre, которую вы используете для самостоятельной публикации различных электронных документов в форматах EPUB и MOBI. Вы хотите, чтобы для всех файлов, сгенерированных calibre, их издатель был установлен как «Hello world», вот как это сделать. Создайте файл с именем `__init__.py` (это специальное имя, которое всегда должно использоваться для основного файла вашего плагина) и введите в него следующий код Python:

```
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
```

Вот и всё. Чтобы добавить этот код в calibre как плагин, просто запустите в папке, в которой вы создали, следующее `__init __.py`:

```
calibre-customize -b .
```

Примечание

В macOS инструменты командной строки находятся внутри пакета calibre, например, если вы установили calibre в `/Applications`, то инструменты командной строки будут в `/Applications/calibre.app/Contents/MacOS/`.

Вы можете скачать плагин Hello World из helloworld_plugin.zip[76].

Каждый раз, когда вы используете calibre для конвертации книги, вызывается метод плагина `run()`, и для конвертированной книги издателю присваивается значение «Hello World». Это тривиальный плагин, давайте перейдем к более сложному примеру, который фактически добавляет компонент в пользовательский интерфейс.

10.5.2 Плагин пользовательского интерфейса

Этот плагин будет распространяться на несколько файлов (чтобы сохранить код в чистоте). Он покажет вам, как получить ресурсы (изображения или файлы данных) из ZIP-файла плагина, позволит пользователям настраивать ваш плагин, как создавать элементы в пользовательском интерфейсе calibre и как получить доступ и запросить базу данных книг в calibre.

Вы можете скачать этот плагин из interface_demo_plugin.zip[77]

Первое, на что нужно обратить внимание, это то, что в этом ZIP-файле содержится гораздо больше файлов. Как объясняется ниже, обратите особое внимание на `plugin-import-name-interface_demo.txt`.

plugin-import-name-interface_demo.txt
: Пустой текстовый файл, используемый для включения магии многофайлового плагина. Этот файл должен присутствовать во всех плагинах, которые используют более одного файла .py. Он должен быть пустым, а его имя файла должно иметь вид: `plugin-import-name-**some_name**.txt`. Наличие этого файла позволяет вам импортировать код из файлов .py, присутствующих в ZIP-файле, используя оператор вроде:

```
from calibre_plugins.some_name.some_module import some_object
```

Всегда должен присутствовать префикс calibre_plugins. `some_name` происходит от имени пустого текстового файла. `some_module` относится к файлу `some_module.py` внутри ZIP-файла. Обратите внимание, что этот импорт столь же эффективен, как и обычный импорт Python. Вы можете создавать пакеты и подпакеты модулей .py внутри ZIP-файла, как обычно (путем определения __init__.py в каждой подпапке), и все должно «просто работать».

Имя, которое вы используете для `some_name`, входит в глобальное пространство имен, общее для всех плагинов, **поэтому сделайте его как можно более уникальным**. Но помните, что это должен быть действительный идентификатор Python (только буквы, цифры и подчеркивание).

__init__.py
: Как и прежде, файл, который определяет класс плагина

main.py
: Этот файл содержит реальный код, который делает что-то полезное

ui.py
: Этот файл определяет интерфейсную часть плагина

images/icon.png
: Значок для этого плагина

[76] https://calibre-ebook.com/downloads/helloworld_plugin.zip
[77] https://calibre-ebook.com/downloads/interface_demo_plugin.zip

about.txt
: Текстовый файл с информацией о плагине

translations
: Папка, содержащая файлы .mo с переводами пользовательского интерфейса вашего плагина на разные языки. Смотрите ниже для деталей.

Теперь давайте посмотрим на код.

__init__.py

Во-первых, обязательный `__init__.py` для определения метаданных плагина:

```
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.
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        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()
```

Единственная заслуживающая внимания особенность - это поле `actual_plugin`. Так как calibre имеет и командную строку, и интерфейсы GUI, плагины GUI, подобные этому, не должны загружать библиотеки GUI в __init__.py. Поле actual_plugin сделает это за вас, сообщив calibre, что фактический плагин находится в другом файле внутри вашего ZIP-архива, который будет загружен только в контексте графического интерфейса пользователя.

Помните, что для этого вам понадобится файл plugin-import-name-some_name.txt в вашем ZIP-файле плагина, как обсуждалось выше.

Также есть несколько способов включения пользовательской конфигурации плагина. Они обсуждаются ниже.

ui.py

Теперь давайте посмотрим на ui.py, который определяет реальный плагин GUI. Исходный код тщательно прокомментирован и должен быть понятен:

```
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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    # 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

        # 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

Фактическая логика для реализации диалогового окна Interface Plugin Demo.

```
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
        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())
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    def about(self):
        # Get the about text from a file inside the plugin zip file
        # The get_resources function is a builtin function defined for all your
        # plugin code. It loads files from the plugin zip file. It returns
        # the bytes from the specified file.
        #
        # Note that if you are loading more than one file, for performance, you
        # should pass a list of names to get_resources. In this case,
        # get_resources will return a dictionary mapping names to bytes. Names that
        # are not found in the zip file will not be in the returned dictionary.
        text = get_resources('about.txt')
        QMessageBox.about(self, 'About the Interface Plugin Demo',
                text.decode('utf-8'))

    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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        # Get currently selected books
        rows = self.gui.library_view.selectionModel().selectedRows()
        if not rows or len(rows) == 0:
            return error_dialog(self.gui, 'Cannot update metadata',
                                'No books selected', show=True)
        # Map the rows to book ids
        ids = list(map(self.gui.library_view.model().id, rows))
        db = self.db.new_api
        for book_id in ids:
            # Get the current metadata for this book from the db
            mi = db.get_metadata(book_id, get_cover=True, cover_as_data=True)
            fmts = db.formats(book_id)
            if not fmts:
                continue
            for fmt in fmts:
                fmt = fmt.lower()
                # Get a python file object for the format. This will be either
                # an in memory file or a temporary on disk file
                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'])
```

Получение ресурсов из плагина ZIP-файла

Система загрузки плагинов calibre определяет несколько встроенных функций, которые позволяют вам удобно получать файлы из ZIP-файла плагина.

get_resources(name_or_list_of_names)
: Эта функция должна вызываться со списком путей к файлам внутри ZIP-файла. Например, чтобы получить доступ к файлу `icon.png` в папке images в ZIP-файле, вы должны использовать: `images/icon.png`. Всегда используйте косую черту в качестве разделителя пути, даже в Windows. Когда вы передаете одно имя, функция вернет необработанные байты этого файла или None, если имя не было найдено в ZIP-файле. Если вы передаете более одного имени, возвращается dict, сопоставляющий имена с байтами. Если имя не найдено, его не будет в возвращаемом dict.

get_icons(name_or_list_of_names, plugin_name="")
: Обёртка для get_resources(), которая создает объекты QIcon из необработанных байтов, возвра-

щенных get_resources. Если имя не найдено в ZIP-файле, соответствующий QIcon будет нулевым. Чтобы поддерживать тему значков, передайте человекопонятное имя вашего плагина как `plugin_name`. Если пользователь использует тему значков с иконками для вашего плагина, они будут загружены в первую очередь.

Включение пользовательской конфигурации вашего плагина

Чтобы позволить пользователям настраивать ваш плагин, вы должны определить три метода в вашем базовом классе плагинов, **is_customizable**, **config_widget** и **save_settings**, как показано ниже:

```
    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:
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            ac.apply_settings()
```

calibre имеет много разных способов хранения данных конфигурации (наследие его длинной истории). Рекомендуемый способ - использовать класс **JSONConfig**, который хранит информацию о вашей конфигурации в файле .json.

Код для управления данными конфигурации в демо-плагине находится в config.py:

```
from calibre.utils.config import JSONConfig

# 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()
```

Объект `prefs` теперь доступен во всем коде плагина с помощью простого:

```
from calibre_plugins.interface_demo.config import prefs
```

Вы можете увидеть объект `prefs`, используемый в main.py:

```
    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

10.5.3 Редактировать книжные плагины

Теперь давайте немного поменяем механизмы и посмотрим на создание плагина для добавления инструментов в редактор книг calibre. Плагин доступен здесь:download_file:*editor_demo_plugin.zip*.

Первым шагом, как и для всех плагинов, является создание пустого текстового файла с именем для импорта, как описано *above* (страница 234). Мы назовем файл `plugin-import-name-editor_plugin_demo.txt`.

Теперь мы создаем обязательный файл `__init__.py`, который содержит метаданные о плагине – его имя, автора, версию и т. д.

```
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)
```

Один плагин редактора может предоставлять несколько инструментов, каждый инструмент соответствует одной кнопке на панели инструментов и записи в меню *Plugins* в редакторе. Они могут иметь подменю, если инструмент имеет несколько связанных действий.

Все инструменты должны быть определены в файле `main.py` вашего плагина. Каждый инструмент - это класс, который наследуется от класса *`calibre.gui2.tweak_book.plugin.Tool`* (страница 383). Давайте посмотрим на `main.py` из демо-плагина, исходный код тщательно прокомментирован и не требует пояснений. Прочитайте документы API класса *`calibre.gui2.tweak_book.plugin.Tool`* (страница 383) для получения более подробной информации.

main.py

Здесь мы увидим определение единственного инструмента, который умножит все размеры шрифта в книге на число, предоставленное пользователем. Этот инструмент демонстрирует различные важные концепции, которые вам понадобятся при разработке ваших собственных плагинов, поэтому вы должны внимательно прочитать (сильно комментированный) исходный код.

```
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'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    #: 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
            # 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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        # we make.
        self.boss.add_savepoint('Before: Magnify fonts')

        container = self.current_container  # The book being edited as a container␣
↪object

        # Iterate over all style declarations in the book, this means css
        # stylesheets, <style> tags and style="" attributes
        for name, media_type in container.mime_map.items():
            if media_type in OEB_STYLES:
                # A stylesheet. Parsed stylesheets are css_parser CSSStylesheet
                # objects.
                self.magnify_stylesheet(container.parsed(name), factor)
                container.dirty(name)  # Tell the container that we have changed the␣
↪stylesheet
            elif media_type in OEB_DOCS:
                # A HTML file. Parsed HTML files are lxml elements

                for style_tag in container.parsed(name).xpath('//*[local-name="style"]
↪'):
                    if style_tag.text and style_tag.get('type', None) in {None, 'text/
↪css'}:
                        # We have an inline CSS <style> tag, parse it into a
                        # stylesheet object
                        sheet = container.parse_css(style_tag.text)
                        self.magnify_stylesheet(sheet, factor)
                        style_tag.text = serialize(sheet, 'text/css', pretty_
↪print=True)
                        container.dirty(name)  # Tell the container that we have␣
↪changed the stylesheet
                for elem in container.parsed(name).xpath('//*[@style]'):
                    # Process inline style attributes
                    block = container.parse_css(elem.get('style'), is_
↪declaration=True)
                    self.magnify_declaration(block, factor)
                    elem.set('style', force_unicode(block.getCssText(separator=' '),
↪'utf-8'))

    def magnify_stylesheet(self, sheet, factor):
        # Magnify all fonts in the specified stylesheet by the specified
        # factor.
        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)
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        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
```

Давайте разберемся с `main.py`. Мы видим, что он определяет один инструмент с именем *Magnify fonts*. Этот инструмент запросит у пользователя число и умножит все размеры шрифтов в книге на это число.

Первым важным моментом является имя инструмента, которое вы должны установить для некоторой относительно уникальной строки, поскольку она будет использоваться в качестве ключа для этого инструмента.

Следующая важная точка входа - это *calibre.gui2.tweak_book.plugin.Tool.create_action()* (страница 384). Этот метод создает объекты QAction, которые появляются на панели инструментов плагинов и меню плагинов. Также, по желанию, назначается сочетание клавиш, которое пользователь может настроить. Инициированный сигнал от QAction подключается к методу ask_user(), который запрашивает у пользователя множитель размера шрифта, а затем запускает код увеличения.

Код увеличения хорошо прокомментирован и довольно прост. Главное, на что следует обратить внимание, это то, что вы получаете ссылку на окно редактора как `self.gui`, а редактор *Boss* как `self.boss`. *Boss* - это объект, который управляет пользовательским интерфейсом редактора. У него есть много полезных методов, которые описаны в классе *calibre.gui2.tweak_book.boss.Boss* (страница 384).

Наконец, есть `self.current_container`, который является ссылкой на книгу, редактируемую как объект *calibre.ebooks.oeb.polish.container.Container* (страница 375). Это представляет книгу как коллекцию составляющих её HTML/CSS/графических файлов и имеет удобные методы для выполнения многих полезных задач. Контейнерный объект и различные полезные служебные функции, которые можно повторно использовать в коде вашего плагина, описаны в *Документация API инструмента редактирования электронных книг* (страница 375).

10.5.4 Добавление переводов в ваш плагин

Вы можете перевести и отобразить все строки пользовательского интерфейса в вашем плагине на любом языке, установленном для основного пользовательского интерфейса.

Первый шаг - просмотреть исходный код вашего плагина и пометить все видимые пользователю строки как переводимые, заключив их в _(). Например:

```
action_spec = (_('My plugin'), None, _('My plugin is cool'), None)
```

Затем используйте какую-нибудь программу для генерации .po файлов из исходного кода вашего плагина. Для каждого языка, на который вы хотите перевести, должен быть один .po-файл. Например: de.po для немецкого, fr.po для французского и так далее. Для этого вы можете использовать программу Poedit[78].

Отправьте эти .po файлы своим переводчикам. Как только вы получите их обратно, скомпилируйте их в файлы .mo. Вы можете снова использовать Poedit для этого или просто сделать:

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

Поместите файлы .mo в папку `translations` вашего плагина.

Последний шаг - просто вызвать функцию *load_translations()* вверху .py файлов вашего плагина. Из соображений производительности эту функцию следует вызывать только в тех файлах .py, которые на самом деле содержат

[78] https://poedit.net/

переводимые строки. Таким образом, в типичном плагине интерфейса пользователя вы бы вызвали его в верхней части `ui.py`, но не в `__init __.py`.

Вы можете протестировать переводы ваших плагинов, изменив язык пользовательского интерфейса в calibre в разделе *Preferences-Interface-Look & feel* или запустив calibre с установленной переменной окружения `CALIBRE_OVERRIDE_LANG`. Например:

```
CALIBRE_OVERRIDE_LANG=de
```

Замените `de` на код языка, который вы хотите проверить.

Для переводов с множественным числом используйте функцию `ngettext()` вместо `_()`. Например:

```
ngettext('Delete a book', 'Delete {} books', num_books).format(num_books)
```

10.5.5 API плагина

Как вы могли заметить выше, плагин calibre - это класс. Существуют разные классы для разных типов плагинов в calibre. Подробности о каждом классе, включая базовый класс всех плагинов, можно найти в *Документация по API для плагинов* (страница 264).

Ваш плагин почти наверняка будет использовать код из calibre. Чтобы узнать, как найти различные функциональные возможности в кодовой базе calibre, прочитайте раздел о calibre *Макет кода* (страница 356).

10.5.6 Отладка плагинов

Первый, самый важный шаг - запустить calibre в режиме отладки. Вы можете сделать это из командной строки с помощью:

```
calibre-debug -g
```

Или изнутри calibre, щелчком правой кнопки мыши по *Preferences* или через сочетание клавиш `Ctrl+Shift+R`.

При запуске из командной строки выходные данные отладки будут выводиться на консоль, а при запуске из calibre выходные данные будут передаваться в текстовый файл.

Вы можете вставить операторы печати в любом месте вашего кода плагина, они будут выводиться в режиме отладки. Помните, что это Python, вам действительно не нужно ничего, кроме операторов print для отладки;) Я разработал все версии calibre, используя только эту технику отладки.

Вы можете быстро протестировать изменения в вашем плагине, используя следующую командную строку:

```
calibre-debug -s; calibre-customize -b /path/to/your/plugin/folder; calibre
```

Это отключит работающий calibre, дождется завершения выключения, затем обновит ваш плагин в calibre и перезапустит calibre.

10.5.7 Больше примеров плагинов

Вы можете найти список множества сложных плагинов calibre здесь[79].

10.5.8 Поделиться своими плагинами с другими

Если вы хотите поделиться плагинами, которые вы создали, с другими пользователями calibre, опубликуйте свой плагин в новой теме на **`форуме плагинов calibre <https://www.mobileread.com/forums/forumdisplay.php?f=237. >`_**.

[79] https://www.mobileread.com/forums/showthread.php?t=118764

10.6 Типографика математических выражений в электронных книгах

Просмотрщик электронных книг calibre умеет отображать математические выражения, встроенные в электронные книги (файлы EPUB и HTML). Вы можете набирать их с помощью TeX, или MathML, или AsciiMath. Просмотрщик эл.книг calibre использует отличную библиотеку MathJax[80]. Это краткое руководство по созданию электронных книг со встроенными формулами, которые хорошо работают в просмотрщике эл. книг calibre.

10.6.1 Простой HTML-файл с формулами

Вы можете вписать математические выражения внутрь простого HTML-файла, а читалка calibre превратит их в правильно набранные формулы. В примере ниже для математических выражений применяется нотация TeX. Вы увидите, что можно использовать обычные команды TeX, пусть с небольшой оговоркой: амперсанды и знаки «меньше» и «больше» должны записываться как &, < и > соответственно.

Сперва необходимо сообщить calibre, что документ содержит формулы. Для этого в раздел <head> HTML-файла добавляется следующий фрагмент кода:

```
<script type="text/x-mathjax-config"></script>
```

И всё, теперь можно писать формулы так же, как в .tex файле. К примеру, вот уравнения Лоренца:

```
<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>
```

Этот фрагмент в читалке calibre выглядит как снимок экрана ниже.

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Рис. 1: *Уравнения Лоренца*

Ниже приведён полный HTML-файл с большим количеством уравнений и встроенной математики. Вы можете преобразовать этот HTML-файл в EPUB в calibre, чтобы в итоге получить электронную книгу, которую легко распространять среди других людей.

```
<!DOCTYPE html>
<html>
<!-- Copyright (c) 2012 Design Science, Inc. -->
```

(продолжается на следующей странице)

[80] https://www.mathjax.org

(продолжение с предыдущей страницы)

```
<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;
  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 \\
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    \frac{\partial X}{\partial v} &amp;amp; \frac{\partial Y}{\partial v} &amp;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>

<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;amp; = \frac{4\pi}{c}\vec{\mathbf{j}} \\
  \nabla \cdot \vec{\mathbf{E}} &amp;amp; = 4 \pi \rho \\
  \nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial\vec{\mathbf{B}}}{\
↪partial t} &amp;amp; = \vec{\mathbf{0}} \\
  \nabla \cdot \vec{\mathbf{B}} &amp;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>
```

Примечание

Просмотрщик calibre поддерживает MathML, а также TeX, но вы должны включить строку `<script type="text/x-mathjax-config"></script>` в свой HTML-файл, иначе MathML не будет отображаться.

10.6.2 Дополнительная информация

Поскольку промотрщик эл. книг calibre для отрисовки формул использует библиотеку MathJax, лучшее место, где можно узнать больше о математике в электронных книгах и получить помощь, — это веб-сайт MathJax[81].

10.7 Создание AZW3 • EPUB • MOBI каталогов

Функция создания каталога calibre позволяет вам создавать каталог вашей библиотеки в различных форматах. Этот файл справки описывает параметры каталогизации при создании каталога в форматах AZW3, EPUB и MOBI.

- *Выбор книг для каталогизации* (страница 251)
- *Включенные разделы* (страница 252)
- *Префиксы* (страница 253)
- *Исключены книги* (страница 253)
- *Исключенные жанры* (страница 254)
- *Другие настройки* (страница 254)
- *Настраиваемый каталог обложек* (страница 255)
- *Дополнительные справочные ресурсы* (страница 255)

10.7.1 Выбор книг для каталогизации

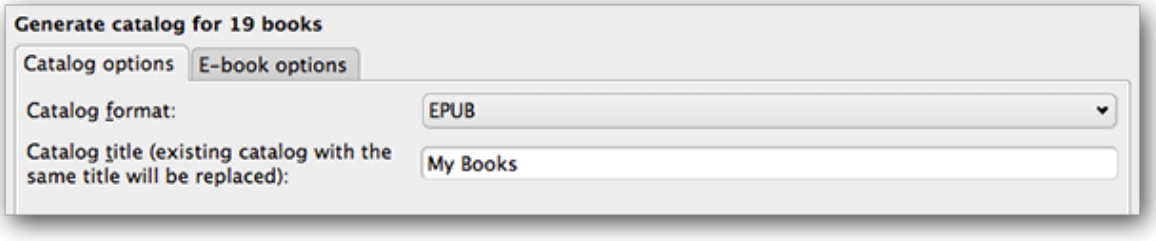
Если вы хотите каталогизировать *всю* библиотеку, удалите все критерии поиска или фильтры в главном окне. Если выбрана одна книга, все книги в вашей библиотеке будут кандидатами для включения в созданный каталог. Отдельные книги могут быть исключены по различным критериям; смотри раздел *Исключенные жанры* (страница 254) ниже для получения дополнительной информации.

Если Вы хотите каталогизировать только *часть* вашей библиотеки, у Вас есть два варианта:

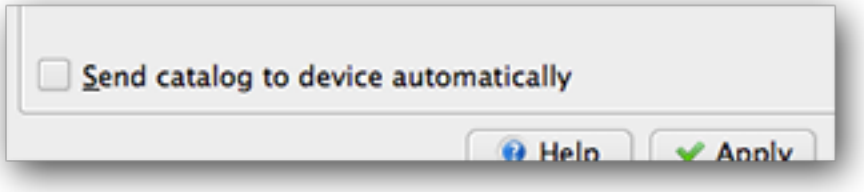
- Выберите книги, которые вы хотите каталогизировать. При выборе более чем одной книги в главном окне Calibre будут каталогизированы только выбранные книги.
- Используйте поле поиска или браузер тегов для фильтрации отображаемых книг. Каталогизируются только отображаемые книги.

Чтобы начать создание каталога, выберите пункт меню *Конвертировать книги* > *Создать каталог книг в библиотеке calibre*. Вы также можете добавить :guilabel: кнопку *Создать каталог* на панели инструментов в *Настройки* > *Интерфейс* > *Панели инструментов* для упрощения доступа к диалогу Создать каталог.

[81] https://www.mathjax.org

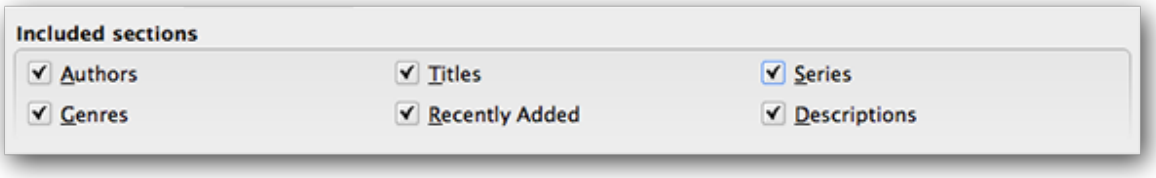


В *Опции каталога*, выберите **AZW3, EPUB или MOBI** в качестве формата Каталога. В поле *Catalog title* укажите имя, которое будет использоваться для сгенерированного каталога. Если каталог с таким же именем и форматом уже существует, он будет заменен вновь созданным каталогом.



Включение *Отправить каталог на устройство автоматически* будет загружать сгенерированный каталог на подключенное устройство по завершении.

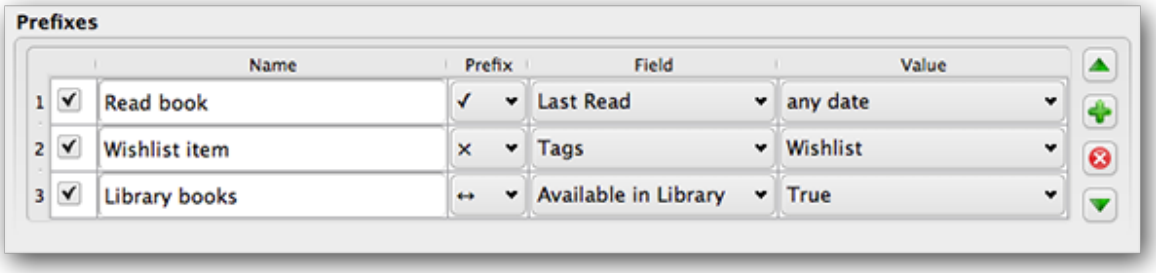
10.7.2 Включенные разделы



Разделы, включенные с помощью галочки, будут включены в сгенерированный каталог:

- *Authors* - все книги, отсортированные по автору, представлены в виде списка. Книги, не относящиеся к серии, перечислены перед книгами серии.
- *Titles* - все книги, отсортированные по названию, представлены в виде списка.
- *Series* - все книги, входящие в серию, отсортированные по сериям, представлены в виде списка.
- *Genres*- отдельные жанры, представленные в списке, отсортированные по автору и серии.
- *Recently Added*- все книги, отсортированные в обратном хронологическом порядке. Список включает книги, добавленные за последние 30 дней, а затем ежемесячные списки добавленных книг.
- *Descriptions*- подробная страница описания каждой книги, включая миниатюру обложки и комментарии. Сортировка по автору, с книгами, не относящимися к серии, перечисленными перед книгами серии.

10.7.3 Префиксы



Префиксные правила позволяют добавлять префикс в списки книг при выполнении определённых критериев. Например, вы можете пометить прочитанные книги с помощью галочки или книги в своем списке желаний с помощью X.

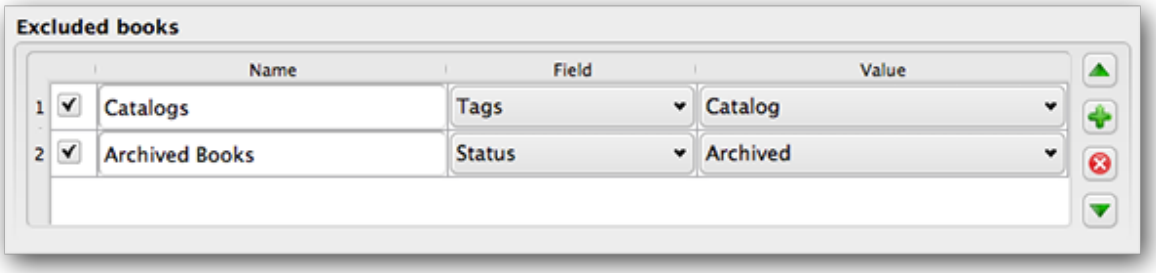
Флажок в первом столбце разрешает это правило. *Name* - это имя правила, которое вы предоставляете. *Field* или *Tags*, или пользовательский столбец из вашей библиотеки. *Value*- содержимое *Field* для совпадения. При выполнении префиксного правила книга помечается выбранным *Prefix*.

В приведенном выше примере приведены три правила префикса:

1. *Read book* указывает, что книга с любой датой в специальном столбце с именем *Last read* будет иметь префикс с символом галочки.
2. *Wishlist`элемент указывает, что любая книга с тегом guilabel:`Wishlist* будет иметь префикс с символом X.
3. *Library* указывает, что любая книга со значением True (или Yes) в настраиваемом столбце :guilabel:`Available in Library“ будет иметь префикс с символом двойной стрелки.

Первое совпадение с правилом содержит префикс. Отключенные или неполные правила игнорируются.

10.7.4 Исключены книги



Правила исключения позволяют вам указывать книги, которые не будут каталогизированы.

Флажок в первом столбце разрешает это правило. *Name* - это имя правила, которое вы предоставляете. *Field`или :guilabel:`Tags*, или пользовательский столбец в вашей библиотеке. *Value* - содержимое *Field* для соответствия. Когда правило исключения будет выполнено, книга будет исключена из сгенерированного каталога.

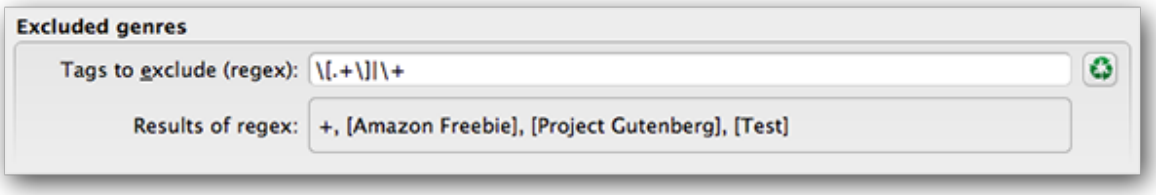
В приведенном выше примере были указаны два правила исключения:

1. Правило *Catalogs`указывает, что любая книга с тегом guilabel:`Catalog* будет исключена из сгенерированного каталога.

2. Правило *Archived* указывает, что любая книга со значением *Archived* в настраиваемом столбце *Status* будет исключена из сгенерированного каталога.

Все правила оцениваются для каждой книги. Отключенные или неполные правила игнорируются.

10.7.5 Исключенные жанры



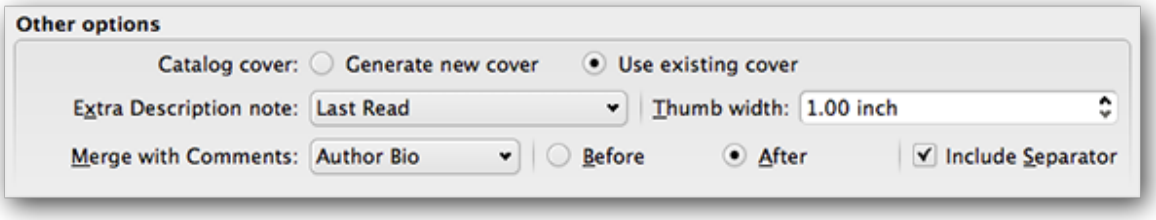
При создании каталога, теги в вашей базе данных - это жанры. Например теги `Fiction` и `Nonfiction` становятся жанрами в сгенерированном каталоге, с книгами, перечисленными в соответствующих списках жанров, на основе назначенных им тегам. Книга будет указана в каждом жанре, для которого у неё есть соответствующий тег.

Вы можете использовать определённые теги для других целей, возможно, +, чтобы указать читаемую книгу, или тег в скобках, вроде `[Amazon Freebie]`, чтобы указать источник книги. : guilabel:*Excluded genres „ regex позволяет указать теги, которые вы не хотите использовать в качестве жанров в сгенерированном каталоге. Шаблон регулярного исключения исключений* `[.+] +`` исключает любые теги формы `[tag]`, а также исключает знак `+` (умолчальный тег для чтения книг) от его использования в качестве жанров в сгенерированном каталоге.

Вы также можете использовать точное имя тега в регулярном выражении. Например, `[Amazon Freebie]` или `[Project Gutenberg]`. Если вы хотите перечислить несколько точных тегов для исключения, поместите между ними символ пайпа (вертикальной полосы): `[Amazon Freebie]|[Project Gutenberg]`.

Results of regex shows you which tags will be excluded when the catalog is built, based on the tags in your database and the regex pattern you enter. The results are updated as you modify the regex pattern.

10.7.6 Другие настройки



Catalog cover указывает, следует ли создавать новую обложку или использовать существующую. Можно создать пользовательскую обложку для ваших каталогов - см. custom catalog covers для получения дополнительной информации. Если вы создали пользовательскую обложку, и хотите её повторно использовать, выберите *Использовать существующую обложку*. В противном случае выберите :guilabel: *Сгенерировать новую обложку*.

Extra Description note`указывает содержимое настраиваемого столбца, которое должно быть вставлено на страницу «Описание», рядом с миниатюрой обложки. Например, вы можете отобразить дату, когда вы последний раз читали книгу, используя пользовательский столбец :guilabel:`Last Read. Для расширенного использования функции Description note см. этот пост на форуме calibre <https://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395>`_.

:guilabel:`Thumb width`задает предпочтение ширины для эскизов обложки, включенных в страницы описаний. Миниатюры кэшируются для повышения производительности. Чтобы поэкспериментировать с разной шириной,

попробуйте создать каталог всего для нескольких книг, пока вы не определите предпочитаемую ширину, а затем сгенерируйте полный каталог. В первый раз, когда каталог создается с новой шириной эскиза, производительность будет медленнее, но последующие сборки этого каталога будут использовать кэш эскизов.

Merge with Comments`определяет настраиваемый столбец, содержимое которого будет недеструктивно слито с метаданными комментариев во время генерации каталога. Например, у вас может быть собственный столбец :guilabel:`Author bio, который вы хотите добавить к метаданным комментариев. Вы можете вставлять содержимое пользовательского столбца *до или после* раздела «Комментарии» и, при необходимости, разделять добавленный контент с помощью горизонтального разделителя. Настраиваемые типы столбцов - «текст, комментарии и составные».

10.7.7 Настраиваемый каталог обложек

С установленным плагином Generate Cover[82] вы можете создавать собственные обложки для своего каталога. Чтобы установить плагин, перейдите по ссылке *Настройки* > *Дополнительно* > *Плагины* > *Получить новые плагины*.

10.7.8 Дополнительные справочные ресурсы

Для получения дополнительной информации о функции Каталога в calibre см. форум MobileRead Creating Catalogs - Start here[83], где вы можете найти информацию о том, как настроить шаблоны каталогов и как отправить отчет об ошибке.

Чтобы задать вопросы или обсудить функции каталогизации calibre с другими пользователями, посетите форум MobileRead Library Management[84].

10.8 Виртуальные библиотеки

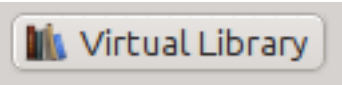
В calibre Виртуальная библиотека - это способ указать calibre открывать только подмножество обычной библиотеки. Например, вы можете работать только с книгами определённого автора или с книгами, имеющими только

[82] https://www.mobileread.com/forums/showthread.php?t=124219
[83] https://www.mobileread.com/forums/showthread.php?t=118556
[84] https://www.mobileread.com/forums/forumdisplay.php?f=236

определённый тег. Использование виртуальных библиотек является предпочтительным способом разделения вашей большой коллекции книг на небольшие подколлекции. Лучше разделить вашу библиотеку на несколько небольших библиотек, так как, когда вы хотите выполнить поиск по всей вашей коллекции, вы можете просто вернуться к полной библиотеке. В calibre нет возможности поиска одновременно по нескольким отдельным библиотекам.

Виртуальная библиотека отличается от простого поиска. Поиск будет ограничивать только список книг, отображаемых в списке книг. Виртуальная библиотека делает это, и кроме того, она также ограничивает записи, показанные в *Браузер тегов* слева. В браузере тегов будут отображаться только теги, авторы, серии, издатели и т. д., взятые из книг в виртуальной библиотеке. Таким образом, виртуальная библиотека ведет себя так, как будто действительная библиотека содержит только ограниченный набор книг.

10.8.1 Создание виртуальных библиотек



Чтобы использовать Виртуальную библиотеку, нажмите кнопку *Виртуальная библиотека*, расположенную слева от панели поиска, и выберите параметр *Создать виртуальную библиотеку*. В качестве первого примера давайте создадим виртуальную библиотеку, которая показывает нам только книги определенного автора. Нажмите на ссылку *Authors*, как показано на рисунке ниже, выберите автора, которого хотите использовать, и нажмите OK.

Диалоговое окно «Создать виртуальную библиотеку» заполнено для вас. Нажмите «OK», и вы увидите, что новая виртуальная библиотека была создана, автоматически выбрана и отображает только книги выбранного автора. Что касается calibre, это значит, что ваша виртуальная библиотека содержит только книги выбранного автора.

Вы можете в любой момент вернуться в полную библиотеку, снова щелкнув по *Virtual library* и выбрав запись с именем: guilabel:<*None*>.

Виртуальные библиотеки основаны на *поисках*. Вы можете использовать любой поиск в качестве основы виртуальной библиотеки. Виртуальная библиотека будет содержать только книги, соответствующие этому поиску. Сначала введите поиск, который вы хотите использовать, в строке поиска или создайте поиск, используя *Браузер тегов*. Когда вы будете довольны полученными результатами, нажмите кнопку *Виртуальная библиотека*, выберите *Создать библиотеку* и введите имя для новой Виртуальной библиотеки. Виртуальная библиотека будет создана на основе только что введенного вами поиска. Поиски очень мощная вещь, примеры того, что вы можете с ними сделать, см. в *Интерфейс поиска* (страница 12).

Примеры полезных виртуальных библиотек

- **Книги добавленные в calibre в последний день**
  date:>1daysago
- **Книги добавленные в calibre за последний месяц**
  date:>30daysago
- **Книги с рейтингом 5 звезд**
  rating:5
- **Книги с рейтингом не менее 4 звезд**
  rating:>=4
- **Книги без рейтинга**
  rating:false
- **Периодические издания, загруженные функцией Fetch News в calibre:**
  tags:=News and author:=calibre
- **Книги без тегов:**
  tags:false
- **Книги без обложек:**
  cover:false

10.8.2 Работа с виртуальными библиотеками

Вы можете отредактировать ранее созданную Виртуальную библиотеку или удалить ее, щелкнув по *Виртуальная библиотека* и выбрав соответствующее действие.

Вы можете сказать calibre, что вы всегда хотите применять определенную виртуальную библиотеку, когда открыта текущая библиотека, перейдя по адресу :guilabel: *Настройки->Интерфейс->Поведение*.

Вы можете быстро использовать текущий поиск в качестве временной Виртуальной библиотеки, нажав кнопку *Виртуальная библиотека* и выбрав запись ** current search*.

Вы можете отобразить все доступные виртуальные библиотеки в виде вкладок над списком книг. Это особенно удобно, если вам очень нравится переключаться между виртуальными библиотеками. Нажмите кнопку *Виртуальная библиотека* и выберите *Показать виртуальные библиотеки как вкладки*. Вы можете изменить порядок вкладок, перетаскивая и закрывая те, которые вы не хотите видеть. Закрытые вкладки можно восстановить, щелкнув правой кнопкой мыши на панели вкладок.

10.8.3 Использование виртуальных библиотек в поиске

Вы можете искать книги в виртуальной библиотеке, используя префикс `vl:`. Например, `vl: Read` найдет все книги в виртуальной библиотеке *Read*. Поиск `vl: Read и vl:"Science Fiction"` найдет все книги, которые находятся в виртуальных библиотеках *Read* и *Science Fiction*.

Значение, следующее за `vl:` должно быть именем виртуальной библиотеки. Если имя виртуальной библиотеки содержит пробелы, заключите его в кавычки.

Одно из применений поиска в виртуальной библиотеке - это сервер содержимого. В *Preferences-Sharing over the net-Require username and password* вы можете ограничить библиотеки calibre, видимые пользователю. Для каждой видимой библиотеки вы можете указать поисковое выражение, чтобы дополнительно ограничить количество просматриваемых книг. Используйте `vl:"Virtual library name"`, чтобы ограничить количество книг теми, что находятся в виртуальной библиотеке.

10.8.4 Использование дополнительных ограничений

Вы можете дополнительно ограничить книги, показанные в виртуальной библиотеке, используя *Дополнительные ограничения*. Дополнительным ограничением является сохраненный ранее созданный вами поиск, который можно применить к текущей виртуальной библиотеке, чтобы дополнительно ограничить книги, отображаемые в виртуальной библиотеке. Например, если у вас есть виртуальная библиотека для книг, помеченных как *Historical Fiction*, и сохраненный поиск, который показывает непрочитанные книги, вы можете нажать кнопку *Virtual Library* и выбрать *Дополнительные ограничения* - для показа только непрочитанных книг по исторической фантастике. Чтобы узнать о сохраненных поисках, смотрите ved_searches.

ГЛАВА 11

Схема URL-адресов calibre://

calibre регистрируется как программа-обработчик для URL-адресов calibre://. Таким образом, вы можете использовать их для выполнения таких действий, как открытие книг, поиск книг и т. д. из других программ / документов или через командную строку. Например, запустив в командной строке следующее:

```
calibre calibre://switch-library/Some_Library
```

Откроет calibre с библиотекой под названием Some Library. Имена библиотек - это имя папки библиотеки с пробелами, заменёнными подчеркиванием. Специальное значение _ означает текущую библиотеку. Ниже описаны различные типы URL-адресов.

Вы можете размещать такие ссылки даже внутри HTML-файлов, документов Word или чего-то подобного, и операционная система автоматически запустит calibre для выполнения указанного действия.

- *Переключение на определённую библиотеку* (страница 259)
- *Отображение определённой книги в calibre* (страница 260)
- *Открытие определённой книги в читалке электронных книг в указанном месте* (страница 260)
- *Поиск книг* (страница 261)
- *Открыть окно сведений о книге в какой-либо библиотеке* (страница 261)
- *Открыть примечания, связанные с автором/серией/и т.д.* (страница 261)
- *Шестнадцатеричное кодирование параметров URL-адреса* (страница 262)

11.1 Переключение на определённую библиотеку

Синтаксис URL-адресов:

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

Имена библиотеки - это имя папки библиотеки с пробелами, замененными подчеркиванием. Специальное значение`` _`` означает текущую библиотеку. Вы также можете использовать: *hex encoding* (страница 262) для имен библиотек, это полезно, если в именах библиотек есть специальные символы, которые потребовали бы URL-кодирования имён. Имена библиотек в шестнадцатеричном формате выглядят так:

```
_hex_-AD23F4BC
```

Где часть после `_hex_-` префикса - это имя библиотеки, закодированное в UTF-8, каждый байт представлен двумя шестнадцатеричными символами.

11.2 Отображение определённой книги в calibre

Синтаксис URL-адресов:

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

Показ книги с `` book_id „“ (число) в calibre. Идентификаторы книг можно увидеть в интерфейсе calibre, наведя курсор на ссылку *Нажмите, чтобы открыть* на панели *Book details*. Это число в скобках в конце пути к папке с книгами.

Вы можете скопировать ссылку на текущую книгу, показанную в calibre, щёлкнув правой в *Панели описания книги* и выбрав *Копировать ссылку на книгу*.

Если поиск активен и книга не соответствует поиску, поиск очищается.

Если выбрана виртуальная библиотека, calibre будет использовать её при показе книги. Если книга не найдена в этой виртуальной библиотеке, виртуальная библиотека очищается.

Если вы хотите переключиться на определённую виртуальную библиотеку, используйте:

```
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
```

замена пробелов в имени виртуальной библиотеки на `%20`. Если книга не найдена в этой виртуальной библиотеке, виртуальная библиотека игнорируется.

11.3 Открытие определённой книги в читалке электронных книг в указанном месте

Синтаксис URL-адресов:

```
calibre://view-book/Library_Name/book_id/book_format?open_at=location
```

Здесь `book_format` - это формат книги, например, `EPUB` или `MOBI`, а `location` - это опциональное местоположение внутри книги. Простейший способ получить эти ссылки - открыть книгу в просмотрщике, затем в элементах управления просмотрщиком выбрать *Go to → Location*, и там будет ссылка, которую вы можете скопировать/вставить где-либо.

11.4 Поиск книг

Синтаксис URL-адресов:

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

Здесь допустим любой запрос *search expression* (страница 12). Если поисковое выражение сложное, *encode it as a hex string* (страница 262) и используйте вместо этого `eq`. Если не указать запрос, текущий поиск будет очищен.

По умолчанию, если выбрана Виртуальная библиотека, calibre очистит её перед выполнением поиска, чтобы убедиться, что все книги найдены. Если вы хотите сохранить Виртуальную библиотеку, используйте:

```
calibre://search/Library_Name?q=query&virtual_library=_
```

Если вы хотите переключиться на определённую виртуальную библиотеку, используйте::

```
calibre://search/Library_Name?virtual_library=Library%20Name
or
calibre://search/Library_Name?encoded_virtual_library=hex_encoded_virtual_library_name
```

заменяя пробелы в названии виртуальной библиотеки на `%20`.

Если вы выполняете поиск в calibre и хотите сгенерировать для него ссылку, щёлкните правой кнопкой мыши по строке поиска и выберите *Копировать поиск как URL*.

11.5 Открыть окно сведений о книге в какой-либо библиотеке

Синтаксис URL-адресов:

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

Это открывает окно сведений о книге для указанной книги из указанной библиотеки без изменения текущей библиотеки или выбранной книги.

11.6 Открыть примечания, связанные с автором/серией/и т.д.

Синтаксис URL-адресов:

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

Это открывает окно, показывающее примечания указанного элемента. Самый простой способ создать такие URL-адреса - показать примечания, желаемые в calibre, и нажать кнопку *Copy URL*, чтобы скопировать URL в буфер обмена и вставить его туда, где вам нужно.

Здесь `Field_Name` - это имя столбцов, таких как `authors` или `tags`. Для созданных пользователем столбцов замените ведущий символ `#` в имени столбца на подчеркивание, так что `#mytags` станет `_mytags`.

Дополнительно к указанию элементов по идентификатору с помощью `Item_Id`, вы также можете указывать их по имени с использованием `val_Item_Name` или `hex_Hex_Encoded_Item_Name`. Например:

```
calibre://book-details/Library_Name/authors/val_John%20Doe
```

11.7 Шестнадцатеричное кодирование параметров URL-адреса

Шестнадцатеричное кодирование параметров URL-адреса выполняется сначала путем кодирования параметра как байтов UTF-8, а затем замены каждого байта двумя шестнадцатеричными символами, представляющими байт. Например, строка «abc» - это байты «0x61 0x62 и 0x63» в UTF-8, поэтому закодированная версия - это строка: «616263».

ГЛАВА 12

Настройка calibre

calibre имеет высокомодульную конструкцию. Все его части можно настроить. Здесь вы узнаете:

- как использовать переменные среды и *твики* для настройки поведения calibre,
- как указать свои собственные статические ресурсы, такие как значки и шаблоны, чтобы переопределить умолчальные значения
- как использовать *плагины* для добавления функциональности в calibre.
- как делиться темами значков и плагинами с другими пользователями calibre.
- Чтобы узнать, как создавать *рецепты* для добавления новых источников онлайн-контента в calibre, посетите Раздел *Добавление любимых новостных веб-сайтов* (страница 33).

Примечание

Обратите внимание, что, хотя темы значков и плагины индексируются и загружаются с помощью встроенного средства обновления calibre, они не являются частью calibre, и их исходный код и поддержка находятся на форумах Mobileread[85] в своих ветках поддержки.

- *Переменные окружения* (страница 298)
- *Твики* (страница 299)
- *Переопределение значков, шаблонов и т. д.* (страница 311)
- *Создание собственной темы значков для calibre* (страница 312)
- *Настройка calibre с помощью плагинов* (страница 312)

[85] https://www.mobileread.com/forums/forumdisplay.php?f=166

12.1 Документация по API для плагинов

Определяет различные абстрактные основные классы, которые можно разделить на подклассы, для создания эффективных плагинов. К полезным классам относятся:

- *Плагин* (страница 264)
- *FileTypePlugin* (страница 266)
- *Плагины метаданных* (страница 268)
- *Плагины каталогизации* (страница 269)
- *Плагины загрузки метаданных* (страница 270)
- *Плагины преобразования* (страница 274)
- *Драйверы устройств* (страница 277)
- *Действия в пользовательском интерфейсе* (страница 292)
- *Плагины параметров* (страница 295)

12.1.1 Плагин

class calibre.customize.**Plugin**(*plugin_path*)

Плагин calibre. Включая эти полезные плагины:

- `self.installation_type`: Хранит способ установки плагина.
- **`self.plugin_path`: Хранит путь к ZIP-файлу, содержащему** этот плагин или None, если это встроенный плагин
- **`self.site_customization`: Сохраняет введенную строку настройки** пользователем.

Методы, которые необходимо переопределить в подклассах:

- *`initialize()`* (страница 265)
- *`customization_help()`* (страница 266)

Полезные методы:

- *`temporary_file()`* (страница 266)
- `__enter__()`
- *`load_resources()`* (страница 265)

`supported_platforms = []`

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

`name = 'Trivial Plugin'`

Название этого плагина. Вы должны установить для него что-то другое, кроме Trivial Plugin, чтобы он работал.

`version = (1, 0, 0)`

Версия этого плагина в виде 3 выпусков (основной, дополнительный, ревизия)

description = 'Ничего не делает'

Короткая строка, описывающая, что делает этот плагин

author = 'Неизвестный'

Автор этого плагина

priority = 1

Если для одного типа файла существует несколько подключаемых модулей, они запускаются в порядке убывания приоритета. Плагины с более высоким приоритетом будут запущены первыми. Максимально возможный приоритет - `sys.maxsize`. Приоритет по умолчанию = 1.

minimum_calibre_version = (0, 4, 118)

Минимальная требуемая для этого плагина версия calibre

installation_type = None

Способ установки этого плагина

can_be_disabled = True

При False пользователь не сможет отключить этот плагин. Используйте с осторожностью.

type = 'Основной'

Тип плагина. Используется для категоризации в графической оболочке.

initialize()

Вызывается один раз при инициализации подключаемых модулей calibre. Плагины повторно инициализируются каждый раз, когда добавляется новый плагин. Также обратите внимание, что если плагин запускается в рабочем процессе, например, добавлении книг, то плагин будет инициализирован для каждого нового рабочего процесса.

Выполните здесь любую инициализацию конкретного плагина, например, извлечение ресурсов из ZIP-файла плагина. Путь к ZIP-файлу доступен как `self.plugin_path`.

Обратите внимание, что на данном этапе `self.site_customization` сейчас **недоступен**.

config_widget()

Реализуйте этот метод и *save_settings()* (страница 265) в своем плагине, чтобы использовать настраиваемое диалоговое окно конфигурации, а не полагаться на простую настройку по умолчанию на основе строк.

Этот метод, если он реализован, должен возвращать QWidget. Виджет может иметь необязательный метод validate(), который не принимает аргументов и вызывается сразу после того, как пользователь нажимает кнопку OK. Изменения применяются тогда и только тогда, когда метод возвращает True.

Если по какой-либо причине вы не можете выполнить настройку в это время, верните кортеж из двух строк (сообщение, подробности), они будут отображаться как диалоговое окно с предупреждением для пользователя, и процесс будет прерван.

save_settings(*config_widget*)

Сохраните настройки, указанные пользователем, с помощью config_widget.

Параметры

config_widget – Виджет, возвращаемый *config_widget()* (страница 265).

do_user_config(*parent=None*)

Этот метод показывает диалог конфигурации для этого плагина. Он возвращает True, если пользователь нажимает OK, в противном случае - False. Изменения применяются автоматически.

load_resources(*names*)

Если этот плагин поставляется в виде ZIP-файла (плагин, добавленный пользователем), этот метод позволит вам загружать ресурсы из ZIP-файла.

Например, чтобы загрузить изображение:

```
pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png'])['images/icon.png
↪'])
icon = QIcon(pixmap)
```

Параметры

names – Список путей к ресурсам в ZIP-файле с использованием разделителя /

Результат

Словарь вида `{name: file_contents}`. Любые имена, которые не были найдены в ZIP-файле, не будут представлены в словаре.

customization_help(*gui=False*)

Вернуть строку с информацией о том, как настроить этот плагин. По умолчанию вызывается `NotImplementedError`, что указывает на то, что плагин не требует настройки.

Если вы повторно реализуете этот метод в своем подклассе, пользователю будет предложено ввести строку в качестве настройки для этого плагина. Строка настройки будет доступна как `self.site_customization`.

Настройкой сайта может быть что угодно, например, путь к необходимому двоичному файлу на компьютере пользователя.

Параметры

gui – Если True, вернуть HTML-справку, иначе вернуть текстовую справку.

temporary_file(*suffix*)

Возвращает файловый объект, который является временным файлом в файловой системе. Этот файл останется доступным даже после закрытия и будет удален только при завершении работы интерпретатора. Используйте член `name` возвращаемого объекта, чтобы получить полный путь к созданному временному файлу

Параметры

suffix – Суффикс временного файла.

cli_main(*args*)

Этот метод является основной точкой входа для интерфейса командной строки ваших плагинов. Он вызывается, когда пользователь делает: calibre-debug -r «Plugin Name». Любые переданные аргументы присутствуют в переменной args.

12.1.2 FileTypePlugin

class calibre.customize.**FileTypePlugin**(*plugin_path*)

Базовые классы: *Plugin* (страница 264)

Плагин, связанный с определенным набором типов файлов.

file_types = {}

Набор типов файлов, для которых следует запускать этот плагин. Используйте „*“ для всех типов файлов. Например: `{'lit', 'mobi', 'prc'}`

on_import = False

Если True, этот плагин запускается при добавлении книг в базу данных

on_postimport = False

Если True, этот плагин запускается после добавления книг в базу данных. В этом случае вызываются методы плагина postimport и postadd.

on_postconvert = False

Если True, этот плагин запускается после конвертации книги. В этом случае вызывается postconvert метод плагина.

on_postdelete = False

Если True, этот плагин запускается после удаления файла книги из базы данных. В этом случае вызывается postdelete метод плагина.

on_preprocess = False

Если True, этот плагин запускается непосредственно перед конвертацией.

on_postprocess = False

Если True, этот плагин запускается после конвертации в финальный файл, созданный плагином вывода конвертации.

type = 'Тип файла'

Тип плагина. Используется для категоризации в графической оболочке.

run(*path_to_ebook*)

Запустить плагин. Должен быть реализован в подклассах. Он должен выполнить все необходимые изменения в электронной книге и вернуть абсолютный путь к измененной электронной книге. Если никаких изменений не требуется, он должен вернуть путь к исходной электронной книге. Если обнаружена ошибка, должно возникнуть исключение. Реализация по умолчанию просто возвращает путь к исходной электронной книге. Обратите внимание, что путь к исходному файлу (до запуска плагинов любого типа файла доступен как self.original_path_to_file).

Изменённый файл электронной книги должен быть создан с помощью метода `temporary_file()`

Параметры

path_to_ebook – Абсолютный путь к электронной книге.

Результат

Абсолютный путь к изменённой электронной книге.

postimport(*book_id*, *book_format*, *db*)

Вызывается пост-импортом, т.е. после того, как файл книги был добавлен в базу данных. Обратите внимание, что это отличается от *postadd()* (страница 268), который вызывается при первом создании записи книги. Этот метод вызывается всякий раз, когда новый файл добавляется к записи книги. Полезно для изменения записи книги на основе содержимого вновь добавленного файла.

Параметры

- **book_id** – Идентификатор базы данных добавленной книги.
- **book_format** – Тип файла добавленной книги.
- **db** – Библиотека базы данных.

postconvert(*book_id*, *book_format*, *db*)

Вызывается после конвертации, т. е. после того, как файл выходной книги конвертации был добавлен в базу данных. Обратите внимание, что он запускается только после преобразования, а не после добавления книги. Это полезно для изменения записи книги на основе содержимого вновь добавленного файла.

Параметры

- **book_id** – Идентификатор базы данных добавленной книги.
- **book_format** – Тип файла добавленной книги.
- **db** – Библиотека базы данных.

postdelete(*book_id*, *book_format*, *db*)

Вызывается после удаления, т. е. после того, как файл книги был удален из базы данных. Обратите внимание, что он не запускается при удалении записи книги, а только при удалении одного или нескольких форматов из книги. Это полезно для изменения записи книги на основе формата удаленного файла.

Параметры

- **book_id** – Идентификатор базы данных добавленной книги.
- **book_format** – Тип файла добавленной книги.
- **db** – Библиотека базы данных.

postadd(*book_id*, *fmt_map*, *db*)

Вызывается post add, т.е. после того, как книга была добавлена в БД. Обратите внимание, что это отличается от *postimport()* (страница 267), который вызывается после того, как в книгу был добавлен отдельный файл книги. postadd() вызывается только тогда, когда вся запись книги, возможно, с более чем одним файлом книги, была создана впервые. Это полезно, если вы хотите изменить запись книги в базе данных при первом добавлении книги в calibre

Параметры

- **book_id** – Идентификатор базы данных добавленной книги.
- **fmt_map** – Сопоставление формата файла с путем, из которого был добавлен формат файла. Обратите внимание, что это может указывать, а может и не указывать на существующий файл, поскольку иногда файлы добавляются в виде потоков. В этом случае это может быть фиктивное значение или несуществующий путь.
- **db** – База данных библиотеки

12.1.3 Плагины метаданных

class calibre.customize.**MetadataReaderPlugin**(**args*, ***kwargs*)

Базовые классы: *Plugin* (страница 264)

Плагин, реализующий чтение метаданных из набора типов файлов.

file_types = {}

Набор типов файлов, для которых следует запускать этот плагин. Например: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

version = (8, 5, 0)

Версия этого плагина в виде 3 выпусков (основной, дополнительный, ревизия)

author = 'Kovid Goyal'

Автор этого плагина

type = 'Чтение метаданных'

Тип плагина. Используется для категоризации в графической оболочке.

get_metadata(*stream*, *type*)

Возвращает метаданные для файла, представленного потоком (объект, подобный файлу, который поддерживает чтение). Вызывает исключение при ошибке во входных данных.

Параметры

type – Тип файла. Гарантированно будет одной из записей в *file_types* (страница 268).

Результат

Объект `calibre.ebooks.metadata.book.Metadata`

class calibre.customize.**MetadataWriterPlugin**(**args*, ***kwargs*)

Базовые классы: *Plugin* (страница 264)

Плагин, реализующий чтение метаданных из набора типов файлов.

file_types = {}

Набор типов файлов, для которых следует запускать этот плагин. Например: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

version = (8, 5, 0)

Версия этого плагина в виде 3 выпусков (основной, дополнительный, ревизия)

author = 'Kovid Goyal'

Автор этого плагина

type = 'Запись метаданных'

Тип плагина. Используется для категоризации в графической оболочке.

set_metadata(*stream*, *mi*, *type*)

Установить метаданные для файла, представленного потоком (объект, подобный файлу, который поддерживает чтение). Вызвать исключение при ошибке во входных данных.

Параметры

- **type** – Тип файла. Гарантированно будет одной из записей в *file_types* (страница 269).
- **mi** – Объект `calibre.ebooks.metadata.book.Metadata`

12.1.4 Плагины каталогизации

class calibre.customize.**CatalogPlugin**(*plugin_path*)

Базовые классы: *Plugin* (страница 264)

Плагин, реализующий генератор каталогов.

file_types = {}

Тип выходного файла, для которого следует запустить этот плагин. Например: „epub“ или „xml“.

type = 'Генератор каталога'

Тип плагина. Используется для категоризации в графической оболочке.

cli_options = []

Параметры CLI парсера, специфичные для этого плагина, объявленные как *namedtuple Option*:

из коллекций импортировать 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 в calibre.db.cli.cmd_catalog:option_parser()

initialize()

Если подключаемый модуль не является встроенным, скопируйте файлы .ui и .py подключаемого модуля из ZIP-файла в $TMPDIR. Вкладка будет динамически сгенерирована и добавлена в диалог Параметры каталога в calibre.gui2.dialogs.catalog.py:Catalog.

run(*path_to_output*, *opts*, *db*, *ids*, *notification=None*)

Запустить плагин. Должно быть реализовано в подклассах. Он должен сгенерировать каталог в формате, указанном в file_types, и вернуть абсолютный путь к сгенерированному файлу каталога. Если обнаружена ошибка, должно возникнуть исключение.

Сгенерированный файл каталога должен быть создан с помощью метода `temporary_file()`.

Параметры

- **path_to_output** – Абсолютный путь к сгенерированному файлу каталога.
- **opts** – Словарь аргументов ключевых слов
- **db** – Объект LibraryDatabase2

12.1.5 Плагины загрузки метаданных

class calibre.ebooks.metadata.sources.base.**Source**(**args*, ***kwargs*)

Базовые классы: *Plugin* (страница 264)

type = 'Источник метаданных'

Тип плагина. Используется для категоризации в графической оболочке.

author = 'Kovid Goyal'

Автор этого плагина

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

capabilities = frozenset({})

Набор возможностей, поддерживаемых этим плагином. Полезные возможности: „identify“, „cover“

touched_fields = frozenset({})

Список полей метаданных, которые потенциально могут быть загружены этим плагином на этапе идентификации

has_html_comments = False

Установите значение True, если ваш плагин возвращает комментарии в формате HTML.

supports_gzip_transfer_encoding = False

Установка этого значения в True означает, что объект браузера укажет, что он поддерживает кодировку передачи gzip. Это может ускорить загрузку, но сначала убедитесь, что источник действительно поддерживает кодировку передачи gzip.

ignore_ssl_errors = False

Установите значение True, чтобы игнорировать ошибки сертификата HTTPS при подключении к этому источнику.

cached_cover_url_is_reliable = True

Кешированные URL-адреса обложек иногда могут быть ненадежными (т.е. загрузка может завершиться сбоем или возвращенное изображение может быть поддельным. Если такое часто случается с этим источником, установите значение False

options = ()

Список объектов `Option`. Они будут использоваться для автоматического создания виджета конфигурации для этого плагина.

config_help_message = None

Строка, которая отображается в верхней части виджета конфигурации для этого плагина.

can_get_multiple_covers = False

Если True, этот источник может вернуть несколько обложек для данного запроса.

auto_trim_covers = False

Если установлено значение True, обложки, загруженные этим плагином, автоматически обрезаются.

prefer_results_with_isbn = True

Если установлено значение True, и этот источник возвращает несколько результатов для запроса, с ISBN и без, результаты без ISBN будут проигнорированы.

is_configured()

Верните False, если ваш плагин нужно настроить, прежде чем его можно будет использовать. Например, может потребоваться имя пользователя/пароль/ключ API.

customization_help()

Вернуть строку с информацией о том, как настроить этот плагин. По умолчанию вызывается `NotImplementedError`, что указывает на то, что плагин не требует настройки.

Если вы повторно реализуете этот метод в своем подклассе, пользователю будет предложено ввести строку в качестве настройки для этого плагина. Строка настройки будет доступна как `self.site_customization`.

Настройкой сайта может быть что угодно, например, путь к необходимому двоичному файлу на компьютере пользователя.

Параметры

gui – Если True, вернуть HTML-справку, иначе вернуть текстовую справку.

config_widget()

Реализуйте этот метод и *`save_settings()`* (страница 271) в своем плагине, чтобы использовать настраиваемое диалоговое окно конфигурации, а не полагаться на простую настройку по умолчанию на основе строк.

Этот метод, если он реализован, должен возвращать QWidget. Виджет может иметь необязательный метод validate(), который не принимает аргументов и вызывается сразу после того, как пользователь нажимает кнопку OK. Изменения применяются тогда и только тогда, когда метод возвращает True.

Если по какой-либо причине вы не можете выполнить настройку в это время, верните кортеж из двух строк (сообщение, подробности), они будут отображаться как диалоговое окно с предупреждением для пользователя, и процесс будет прерван.

save_settings(*config_widget*)

Сохраните настройки, указанные пользователем, с помощью config_widget.

Параметры

config_widget – Виджет, возвращаемый *config_widget()* (страница 271).

get_author_tokens(*authors*, *only_first_author=True*)

Взять список авторов и вернуть список токенов, полезных для поискового запроса AND. Эта функция пытается вернуть токены в порядке имени, отчества, фамилии, предполагая, что если в имени автора стоит запятая, имя находится в фамилии, в другой форме.

get_title_tokens(*title*, *strip_joiners=True*, *strip_subtitle=False*)

Взять заголовок и веруть список токенов, полезных для поискового запроса AND. Исключает связки (необязательно) и знаки препинания.

split_jobs(*jobs*, *num*)

Разделить список заданий на максимальное количество групп, как можно равномернее.

test_fields(*mi*)

Вернуть первое поле из self.touched_fields, которое имеет значение null для объекта mi

clean_downloaded_metadata(*mi*)

Вызовите этот метод в методе идентификации вашего плагина, чтобы нормализовать метаданные перед помещением объекта Metadata в result_queue. Можно использовать собственный алгоритм, подходящий для вашего источника метаданных.

get_book_url(*identifiers*)

Вернуть 3-кортеж или None. Кортеж из трёх элементов имеет форму: (identifier_type, identifier_value, URL). URL-адрес - это URL-адрес книги, идентифицированный идентификаторами в этом источнике. identifier_type, identifier_value указывают идентификатор, соответствующий URL-адресу. Этот URL-адрес должен быть доступен для просмотра человеком, использующим браузер. Он предназначен для предоставления пользователю интерактивной ссылки, чтобы пользователь мог легко перейти на страницу с книгами в этом источнике. Если URL-адрес не найден, вернуть None. Этот метод должен быть быстрым и последовательным, поэтому используйте его только в том случае, если возможно построить URL-адрес из известной схемы с заданными идентификаторами.

get_book_url_name(*idtype*, *idval*, *url*)

Вернуть человекочитаемое имя из возвращаемого значения get_book_url ().

get_book_urls(*identifiers*)

Переопределите этот метод, если вы хотите вернуть несколько URL-адресов для этой книги. Вернуть список 3-tuples (кортеж из трех элементов). По умолчанию этот метод просто вызывает *get_book_url()* (страница 272).

get_cached_cover_url(*identifiers*)

Возвращает кешированный URL-адрес обложки для книги, идентифицированной словарем идентификаторов, или None, если такой URL-адрес не существует.

Обратите внимание, что этот метод должен возвращать только проверенные URL-адреса, то есть не те URL-адреса, которые могут привести к общему изображению обложки или ошибке «Не найдено».

id_from_url(*url*)

Проанализировать URL-адрес и вернуть кортеж в форме: (identifier_type, identifier_value). Если URL-адрес не соответствует шаблону для источника метаданных, вернуть None.

identify_results_keygen(*title=None*, *authors=None*, *identifiers={}*)

Вернуть функцию, которая используется для генерации ключа, который может сортировать объекты метаданных по их релевантности с учетом поискового запроса (заголовок, авторы, идентификаторы).

Эти ключи используются для сортировки результатов вызова *identify()* (страница 273).

Подробнее об алгоритме по умолчанию см. *InternalMetadataCompareKeyGen* (страница 274). Повторно реализуйте эту функцию в своем плагине, если алгоритм по умолчанию не подходит.

identify(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*)

Определить книгу по её названию/автору/ISBN/и т.д.

Если идентификаторы указаны и совпадений не найдено, и этот источник метаданных не хранит все связанные идентификаторы (например, все ISBN книги), этот метод должен повторить попытку, указав только заголовок и автора (при условии, что они были указаны).

Если этот источник метаданных также предоставляет обложки, URL-адрес обложки должен быть кэширован, чтобы при последующем вызове API получения обложек с тем же ISBN/специальным идентификатором не нужно было снова получать URL-адрес обложки. Используйте для этого API кеширования.

Каждый объект Metadata, помещенный в result_queue этим методом, должен иметь атрибут source_relevance, который является целым числом, указывающим порядок, в котором результаты были возвращены источником метаданных для этого запроса. Это целое число будет использоваться `compare_identify_results()`. Если порядок не важен, установите его в 0 (ноль) для каждого результата.

Прежде чем объект Metadata будет помещен в result_queue, убедитесь, что вся информация сопоставления обложки/ISBN закэширована.

Параметры

- **log** – Объект журнала, используйте его для вывода отладочной информации/ошибок
- **result_queue** – Очередь результатов, результаты должны быть помещены в неё. Каждый результат - это объект метаданных.
- **abort** – Если abort.is_set() возвращает True, прервать дальнейшую обработку и вернутся как можно скорее.
- **title** – Название книги может быть None
- **authors** – Список авторов книги может быть None
- **identifiers** – Словарь других идентификаторов, чаще всего {„isbn“: „1234 …“}
- **timeout** – Тайм-аут в секундах, ни один сетевой запрос не должен зависать дольше таймаута.

Результат

None, если ошибок не было, иначе - представление ошибки в Юникоде, подходящее для показа пользователю.

download_cover(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*, *get_best_cover=False*)

Скачать обложку и поместить её в result_queue. Все параметры имеют то же значение, что и для *identify()* (страница 273). Поместить (self, cover_data) в result_queue.

Этот метод должен использовать кешированные URL-адреса обложек для повышения эффективности, при возможности. Когда кэшированных данных нет, большинство плагинов просто вызывают идентификацию и используют её результаты.

Если параметр get_best_cover имеет значение True и этот плагин может получить несколько обложек, он должен получить только «лучшую».

class calibre.ebooks.metadata.sources.base.**InternalMetadataCompareKeyGen**(*mi*, *source_plugin*, *title*, *authors*, *identifiers*)

Сгенерировать ключ сортировки для сравнения релевантности объектов метаданных с учётом поискового запроса. Используется только для сравнения результатов из одного источника метаданных, но не из разных источников.

Ключ сортировки гарантирует, что сортировка в порядке возрастания является сортировкой по убыванию релевантности.

Алгоритм такой:

- Предпочитать результаты, у которых есть хотя бы один идентификатор, такой же как у запроса
- Предпочитать результаты с кешированным URL обложки
- Предпочитать результаты со всеми заполненными полями
- Предпочитать результаты на том же языке, что и текущий язык пользовательского интерфейса
- Предпочитать результаты, которые точно соответствуют названию запроса
- Предпочитать результаты с более длинными комментариями, более чем на 10% длиннее
- **Используйте релевантность результата, полученную в результате поиска источника метаданных**
  движок

12.1.6 Плагины преобразования

class calibre.customize.conversion.**InputFormatPlugin**(**args*)

Базовые классы: *Plugin* (страница 264)

InputFormatPlugins отвечают за преобразование документа в HTML + OPF + CSS + и т. д. Результаты преобразования *должны* быть закодированы в UTF-8. Основное действие происходит в *convert()* (страница 275).

type = 'Ввод для конвертации'

Тип плагина. Используется для категоризации в графической оболочке.

can_be_disabled = False

При False пользователь не сможет отключить этот плагин. Используйте с осторожностью.

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

file_types = {}

Набор типов файлов, для которых следует запускать этот плагин. Например: `set (['azw', 'mobi', 'prc'])`

is_image_collection = False

Если True, этот плагин ввода генерирует коллекцию изображений, по одному на файл HTML. Устанавливается динамически в методе convert, если входные файлы могут быть как коллекциями изображений, так и коллекциями без изображений. Если вы установите для него значение True, то реализуйте метод get_images(), который возвращает список изображений.

core_usage = 1

Количество ядер процессора, используемых этим плагином. Значение -1 означает, что он использует все доступные ядра.

for_viewer = False

Если установлено значение True, плагин ввода будет выполнять специальную обработку, чтобы сделать его вывод подходящим для просмотра.

output_encoding = 'utf-8'

Кодировка, в которой этот модуль ввода создает файлы. Значение None означает, что кодировка не определена и должна определяться индивидуально.

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Параметры, общие для всех плагинов формата ввода. Не переопределять в подклассах. Вместо этого используйте *options* (страница 275). Каждая опция должна быть экземпляром `OptionRecommendation`.

options = {}

Параметры для настройки поведения этого плагина. Каждая опция должна быть экземпляром `OptionRecommendation`.

recommendations = {}

Набор из 3-tuples (трех кортежей) формы (option_name, recommended_value, recommendation_level)

get_images()

Вернуть список абсолютных путей к изображениям, если этот входной плагин представляет коллекцию изображений. Список изображений находится в том же порядке, что и корешок и оглавление.

convert(*stream*, *options*, *file_ext*, *log*, *accelerators*)

Этот метод должен быть реализован в подклассах. Он должен возвращать путь к созданному файлу OPF или экземпляру `OEBBook`. Весь вывод должен содержаться в текущей папке. Если этот плагин создает файлы вне текущей папки, их необходимо удалить/пометить для удаления, прежде чем этот метод вернёт значение.

Параметры

- **stream** – Файловый объект, содержащий входной файл.
- **options** – Параметры для настройки процесса конвертации. Гарантированно наличие атрибутов, соответствующих всем параметрам, заявленным этим плагином. Кроме того, будет подробный атрибут, принимающий целые значения от нуля и выше. Большие числа означают более подробную информацию. Ещё один полезный атрибут - input_profile, который является экземпляром :class:calibre.customize.profiles.InputProfile.
- **file_ext** – Расширение (без .) входного файла. Гарантируется, что это будет один из *file_types*, поддерживаемых этим плагином.
- **log** – Объект `calibre.utils.logging.Log`. Весь вывод должен использовать этот объект.
- **accelerators** – Словарь различной информации, которую плагин ввода может легко получить, что ускорит последующие этапы преобразования.

postprocess_book(*oeb*, *opts*, *log*)

Вызывается, чтобы разрешить плагину ввода выполнять постобработку после анализа книги.

specialize(*oeb*, *opts*, *log*, *output_fmt*)

Вызывается, чтобы позволить плагину ввода специализировать анализируемую книгу для определенного формата вывода. Вызывается после postprocess_book и перед выполнением любых преобразований в проанализированной книге.

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

Вызывается для создания виджета, используемого для настройки этого плагина в графическом интерфейсе calibre. Виджет должен быть экземпляром класса PluginWidget. См. примеры встроенных модулей ввода.

class calibre.customize.conversion.**OutputFormatPlugin**(**args*)

Базовые классы: *Plugin* (страница 264)

OutputFormatPlugins отвечают за преобразование документа OEB (OPF + HTML) в выходную электронную книгу.

Предполагается, что документ OEB закодирован в UTF-8. Основное действие происходит в *convert()* (страница 276).

type = 'Вывод для конвертации'

Тип плагина. Используется для категоризации в графической оболочке.

can_be_disabled = False

При False пользователь не сможет отключить этот плагин. Используйте с осторожностью.

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

file_type = None

Тип файла (расширение без начальной точки), который выводит этот плагин.

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Параметры, общие для всех плагинов формата ввода. Не переопределять в подклассах. Вместо этого используйте *options* (страница 276). Каждая опция должна быть экземпляром `OptionRecommendation`.

options = {}

Параметры для настройки поведения этого плагина. Каждая опция должна быть экземпляром `OptionRecommendation`.

recommendations = {}

Набор из 3-tuples (трех кортежей) формы (option_name, recommended_value, recommendation_level)

property description

str(object="") -> str str(bytes_or_buffer[, encoding[, errors]]) -> str

Создать новый строковый объект из данного объекта. Если указаны кодировка или ошибки, то объект должен предоставить буфер данных, который будет декодирован с использованием данной кодировки и обработчика ошибок. В противном случае возвращает результат object.__str__() (если определено) или repr(object). кодировка по умолчанию „utf-8“. ошибки по умолчанию „strict“.

convert(*oeb_book*, *output*, *input_plugin*, *opts*, *log*)

Отрендерить содержимое *oeb_book* (который является экземпляром `calibre.ebooks.oeb.OEBBook`) в файл, указанный в output.

Параметры

- **output** – Либо файл, подобный объекту, либо строка. Если это строка, это путь к папке, которая может существовать, а может и не существовать. Плагин вывода должен

записывать свой вывод в эту папку. Если это объект, подобный файлу, плагин вывода должен записывать свой вывод в файл.

- **input_plugin** – Плагин ввода, который использовался в начале конвейера конвертации.
- **opts** – Опции конвертации. Гарантировано наличие атрибутов, соответствующих рекомендациям OptionRecommendations этого плагина.
- **log** – Логгер. Печатать отладочные/информационные сообщения и т. д.

specialize_options(*log*, *opts*, *input_fmt*)

Может использоваться для изменения значений параметров конвертации, используемых конвейером конвертации.

specialize_css_for_output(*log*, *opts*, *item*, *stylizer*)

Может использоваться для внесения изменений в CSS во время процесса уплощения CSS.

Параметры

- **item** – HTML-файл элемента, который обрабатывается
- **stylizer** – Объект Stylizer, содержащий уплощённые стили для элемента. Вы можете получить стиль для любого элемента с помощью stylizer.style(element).

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

Вызывается для создания виджета, используемого для настройки этого плагина в графическом интерфейсе calibre. Виджет должен быть экземпляром класса PluginWidget. См. примеры встроенных модулей ввода.

12.1.7 Драйверы устройств

Базовый класс для всех драйверов устройств *DevicePlugin* (страница 277). Однако, если ваше устройство представляет себя для операционной системы как USBMS-накопитель, следует использовать вместо него класс USBMS, поскольку он реализует всю логику, необходимую для поддержки таких типов устройств.

class calibre.devices.interface.**DevicePlugin**(*plugin_path*)

Базовые классы: *Plugin* (страница 264)

Определяет интерфейс, который должен быть реализован серверными модулями, которые взаимодействуют с устройством для чтения электронных книг.

type = 'Интерфейс устройства'

Тип плагина. Используется для категоризации в графической оболочке.

FORMATS = ['lrf', 'rtf', 'pdf', 'txt']

Упорядоченный список поддерживаемых форматов

VENDOR_ID = 0

VENDOR_ID может быть целым числом, списком целых чисел или словарем. Если это словарь, то это должен быть словарь словарей в форме:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

Целое число или список целых чисел

BCD = None

BCD может иметь значение None, чтобы не различать устройства на основе BCD, или это может быть список BCD номеров всех устройств, поддерживаемых этим драйвером.

THUMBNAIL_HEIGHT = 68

Высота миниатюр на устройстве

THUMBNAIL_COMPRESSION_QUALITY = 75

Качество сжатия миниатюр. Значение ближе к 100 - миниатюры лучшего качества с минимумом артефактов сжатия. Размер файлов миниатюр увеличивается.

WANTS_UPDATED_THUMBNAILS = False

Значение параметра True, если устройство поддерживает обновление миниатюр обложек во время sync_booklists. При значении True у device.py запрашивается обновление миниатюр обложек при сопоставлении книг.

CAN_SET_METADATA = ['title', 'authors', 'collections']

Можно ли установить метаданные книг через графический интерфейс.

CAN_DO_DEVICE_DB_PLUGBOARD = False

Может ли устройство обрабатывать коммутации метаданных device_db

path_sep = '/'

Разделитель путей для путей к книгам на устройстве

icon = 'reader.png'

Значок для этого устройства

UserAnnotation

псевдоним для `Annotation`

OPEN_FEEDBACK_MESSAGE = None

В графическом интерфейсе это отображается как сообщение, если в строке статуса не None. Полезно, если открытие может занять много времени

VIRTUAL_BOOK_EXTENSIONS = frozenset({})

Набор расширений, которые являются «виртуальными книгами» на устройстве и поэтому не могут быть просмотрены/сохранены/добавлены в библиотеку. Например: `frozenset(['kobo'])`

VIRTUAL_BOOK_EXTENSION_MESSAGE = None

Сообщение, отображаемое пользователю для расширений виртуальных книг.

NUKE_COMMENTS = None

Удалять ли комментарии в копии книги, отправленной на устройство. Если не None, это должна быть короткая строка, которой будут заменены комментарии.

MANAGES_DEVICE_PRESENCE = False

Если True указывает, что этот драйвер полностью управляет обнаружением устройства, извлечением и т. д. При установке для него значение True, вы *должны* реализовать методы detect_managed_devices и debug_managed_device_detection. Драйвер, для которого установлено значение True, отвечает за обнаружение устройств, управление черным списком устройств, списком извлеченных устройств и т. д. calibre будет периодически вызывать метод detect_managed_devices(), и если он возвращает обнаруженное устройство, calibre вызовет open(). open() будет вызываться каждый раз, когда устройство возвращается, даже если предыдущие вызовы open() завершились неудачно, поэтому драйвер должен поддерживать свой собственный черный список отказавших устройств. Точно так же при извлечении calibre вызовет eject(), а затем, предполагая, что следующий вызов detect_managed_devices() вернет None, вызовет post_yank_cleanup().

SLOW_DRIVEINFO = False

Если установлено True, calibre вызовет метод *get_driveinfo()* (страница 281) после загрузки списков книг для получения информации о диске.

ASK_TO_ALLOW_CONNECT = False

Если установлено значение True, при первом обнаружении устройства calibre спросит пользователя, хочет ли он управлять устройством с помощью calibre. Если установить параметру значение True, то вы должны реализовать *get_device_uid()* (страница 284) и *ignore_connected_device()* (страница 284) и *get_user_blacklisted_devices()* (страница 284) и *set_user_blacklisted_devices()* (страница 284)

user_feedback_after_callback = None

Установите значение как словарь формы {„title“: title, „msg“: msg, „det_msg“: detail_msg}, чтобы calibre отображал всплывающее сообщение для пользователя после выполнения некоторых обратных вызовов (в настоящее время только upload_books). Не спамьте пользователя потоком сообщений. Эта переменная проверяется после *каждого* обратного вызова, поэтому устанавливайте её только при необходимости.

classmethod get_open_popup_message()

GUI отображает это как немодальное всплывающее окно. Должен быть экземпляром OpenPopupMessage

classmethod model_metadata() → tuple[ModelMetadata, ...]

Метаданные обо всех реальных моделях устройств, поддерживаемых этим драйвером

is_usb_connected(*devices_on_system*, *debug=False*, *only_presence=False*)

Вернуть True, device_info, если в данный момент подключено устройство, управляемое этим плагином.

Параметры

devices_on_system – Список подключенных устройств

detect_managed_devices(*devices_on_system*, *force_refresh=False*)

Вызывается, только если у MANAGES_DEVICE_PRESENCE значение True.

Просканировать устройства, с которыми может работать этот драйвер. Должен возвращать объект устройства, если устройство найдено. Этот объект будет передан методу open() как connected_device. Если устройство не найдено, вернуть None. Возвращаемый объект может быть любым, calibre его не использует, он передается только в open().

Этот метод периодически вызывается графическим интерфейсом пользователя, поэтому убедитесь, что он не слишком ресурсоемкий. Используйте кеш, чтобы избежать повторного сканирования системы.

Параметры

- **devices_on_system** – Набор USB-устройств, обнаруженных в системе.
- **force_refresh** – Если True и драйвер использует кеш для предотвращения повторного сканирования, то кеш должен быть очищен.

debug_managed_device_detection(*devices_on_system*, *output*)

Вызывается, только если у MANAGES_DEVICE_PRESENCE значение True.

Должен записывать информацию об устройствах, обнаруженных в системе в вывод, представляющий собой файл, подобный объекту.

Должен возвращать True, если устройство было обнаружено и успешно открыто, в противном случае - False.

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Параметры

- **key** – Ключ для разблокировки устройства
- **log_packets** – Если true, поток пакетов к/от устройства логируется.
- **report_progress** – Функция, которая вызывается с % прогресса (число от 0 до 100) для различных задач. Если она вызывается с -1, это означает, что задача не имеет информации о ходе выполнения.
- **detected_device** – Device information from the device scanner

can_handle_windows(*usbdevice*, *debug=False*)

Необязательный метод для выполнения дальнейших проверок устройства, чтобы узнать, способен ли этот драйвер обрабатывать его. Если это не так, он должен вернуть False. Этот метод вызывается только после того, как поставщик, идентификаторы продуктов и bcd совпадают, поэтому он может выполнять некоторые относительно трудоемкие проверки. Реализация по умолчанию возвращает True. Этот метод вызывается только в Windows. См. также *can_handle()* (страница 280).

Для устройств на базе USBMS этот метод по умолчанию делегирует *can_handle()* (страница 280). Таким образом, вам нужно только переопределить *can_handle()* (страница 280) в подклассе USBMS.

Параметры

usbdevice – USB-устройство, возвращенное `calibre.devices.winusb.scan_usb_devices()`

can_handle(*device_info*, *debug=False*)

Версия для Unix *can_handle_windows()* (страница 280).

Параметры

device_info – Кортеж из (vid, pid, bcd, производитель, продукт, серийный номер)

open(*connected_device*, *library_uuid*)

Выполнить любую инициализацию конкретного устройства. Вызывается после обнаружения устройства, но перед любыми другими функциями, которые обмениваются данными с устройством. Например: для устройств, которые представляют себя как запоминающие устройства USB, этот метод будет отвечать за установку устройства или, если устройство было автоматически смонтировано, за определение места его установки. Метод *calibre.devices.usbms.device.Device.open()* (страница 289) имеет реализацию этой функции, которая должна служить хорошим примером для запоминающих устройств USB.

Этот метод может вызвать исключение OpenFeedback для отображения сообщения пользователю.

Параметры

- **connected_device** – Устройство, которое мы пытаемся открыть. Это кортеж из (идентификатор поставщика, идентификатор продукта, bcd, название производителя, название продукта, серийный номер устройства). Однако у некоторых устройств нет серийного номера, и в Windows присутствуют только первые три поля, остальные - None.
- **library_uuid** – UUID текущей библиотеки calibre. Может иметь значение None, если библиотеки нет (например, при использовании из командной строки).

eject()

Размонтировать/извлечь устройство из ОС. Не проверяется есть ли ожидающие задания графического интерфейса пользователя, которым необходимо взаимодействовать с устройством.

ПРИМЕЧАНИЕ. Этот метод нельзя вызывать в том же потоке, что и остальные методы устройства.

post_yank_cleanup()

Вызывается, если пользователь дергает устройство, не извлекая его предварительно.

set_progress_reporter(*report_progress*)

Установить функцию для сообщения информации о ходе выполнения.

Параметры

report_progress – Функция, которая вызывается с % прогресса (число от 0 до 100) для различных задач. Если она вызывается с -1, это означает, что задача не имеет информации о ходе выполнения.

get_device_information(*end_session=True*)

Спросить у устройства информацию об устройстве. См. L{DeviceInfoQuery}.

Результат

(имя устройства, версия устройства, версия программного обеспечения на устройстве, тип MIME). Кортеж может дополнительно иметь пятый элемент, который является словарем информации о накопителе. См. пример usbms.driver.

get_driveinfo()

Вернуть словарь driveinfo. Обычно вызывается из get_device_information(), но если загрузка информации о диске для этого драйвера происходит медленно, то следует установить SLOW_DRIVEINFO. В этом случае этот метод будет вызываться calibre после загрузки списков книг. Обратите внимание, что он не вызывается в потоке устройства, поэтому драйвер должен кэшировать информацию о диске в методе books(), и эта функция должна возвращать кэшированные данные.

card_prefix(*end_session=True*)

Возвращает список из 2 элементов префикса для путей на карточках. Если карта отсутствует, для префикса карты устанавливается значение None. НАПРИМЕР („/place“, „/place2“) (None, „place2“) („place“, None) (None, None)

total_space(*end_session=True*)

Получить общее пространство, доступное на точках монтирования:

1. Основная память
2. Карта памяти A
3. Карта памяти B

Результат

Список из трех элементов с общим объёмом в байтах (1, 2, 3). Если на конкретном устройстве нет ни одного из этих местоположений, оно должно вернуть 0.

free_space(*end_session=True*)

Получить свободное место на точках монтирования:

1. Основная память
2. Карта 1
3. Карта 2

Результат

Список из трех элементов со свободным пространством в байтах (1, 2, 3). Если конкретное устройство не имеет ни одного из этих местоположений, оно должно вернуть -1.

books(*oncard=None*, *end_session=True*)

Вернуть список электронных книг на устройстве.

Параметры

oncard – Если „carda“ или „cardb“ возвращают список электронных книг на определенной карте памяти, в противном случае возвращает список электронных книг в основной памяти устройства. Если карта указана и на ней нет книг, вернуть пустой список.

Результат

СписокКниг.

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)

Загрузите список книг на устройство. Если файл уже существует на устройстве, его следует заменить. Этот метод должен вызывать `FreeSpaceError`, если на устройстве недостаточно свободного места. Текст FreeSpaceError должен содержать слово «card», если `on_card` не равно None, в противном случае он должен содержать слово «memory».

Параметры

- **files** – Список путей
- **names** – Список имен файлов книг которые были уже загружены на устройство. len(names) == len(files)
- **metadata** – Если не None, это список объектов `Metadata`. Идея состоит в том, чтобы использовать метаданные, чтобы определить, где на устройстве разместить книгу. len(metadata) == len(files). Помимо обычной обложки (путь к обложке), также может быть атрибут эскиза, который следует использовать в предпочтении. Атрибут эскиза имеет форму (ширина, высота, данные обложки в формате jpeg).

Результат

Список трехэлементных кортежей. Список предназначен для передачи в *add_books_to_metadata()* (страница 282).

classmethod add_books_to_metadata(*locations*, *metadata*, *booklists*)

Добавить местоположения в списки книг. Эта функция не должна связываться с устройством.

Параметры

- **locations** – Результат звонка L {upload_books}
- **metadata** – Список объектов `Metadata`, такой же, как для *upload_books()* (страница 282).
- **booklists** – Кортеж, содержащий результат вызовов (`books (oncard = None)()`,: meth:*books(oncard=“carda“)*,: meth`books(oncard=“cardb“)`).

delete_books(*paths*, *end_session=True*)

Удалять книги по путям на устройстве.

classmethod remove_books_from_metadata(*paths*, *booklists*)

Удалить книги из списка метаданных. Эта функция не должна связываться с устройством.

Параметры

- **paths** – пути к книгам на устройстве.
- **booklists** – Кортеж, содержащий результат вызовов (`books (oncard = None)()`,: meth:*books(oncard=“carda“)*,: meth`books(oncard=“cardb“)`).

sync_booklists(*booklists*, *end_session=True*)

Обновить метаданные на устройстве.

Параметры
booklists – Кортеж, содержащий результат вызовов (`books (oncard = None)`(),: meth:*books(oncard="carda")*,: meth`books(oncard="cardb")`).

get_file(*path*, *outfile*, *end_session=True*)

Прочитать файл по `пути` на устройстве и записать его в outfile.

Параметры
outfile – файловый объект, например `sys.stdout`, или результат вызова *open()* (страница 280).

classmethod config_widget()

Должен вернуть QWidget. QWidget содержит настройки интерфейса устройства.

classmethod save_settings(*settings_widget*)

Должно сохранить настройки на диск. Принимает виджет, созданный в *config_widget()* (страница 283), и сохраняет все настройки на диск.

classmethod settings()

Должно вернуть объект opts. Объект opts должен иметь как минимум один атрибут format_map, представляющий собой упорядоченный список форматов для устройства.

set_plugboards(*plugboards*, *pb_func*)

предоставить драйверу текущий набор коммутационных панелей и функцию для выбора конкретной коммутационной панели. Этот метод вызывается непосредственно перед add_books и sync_booklists.

pb_func - это вызываемый объект со следующей подписью:
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).

Результат
Нет или один экземпляр коммутационной панели.

set_driveinfo_name(*location_code*, *name*)

Задать имя устройства в файле информации о диске „name". Этот параметр будет действовать до тех пор, пока файл не будет создан заново или имя не будет изменено снова.

Недисковые устройства должны реализовывать этот метод на основе кодов местоположения, возвращаемых методом get_device_information().

prepare_addable_books(*paths*)

Учитывая список путей, возвращает другой список путей. Эти пути указывают на добавляемые версии книг.

Если при подготовке книги произошла ошибка, то вместо пути в возвращаемом списке для этой книги должна быть позиция из трёх кортежей: (исходный_путь, экземпляр исключения, трассировка)

startup()

Вызывается, когда calibre запускает устройство. Выполните любую требуемую инициализацию. Обратите внимание, что можно создать несколько экземпляров класса, и, следовательно, __init__ можно вызывать несколько раз, но только один экземпляр будет вызывать этот метод. Этот метод вызывается в потоке устройства, а не в потоке графического интерфейса пользователя.

shutdown()

Вызывается, когда calibre выключается навсегда или при подготовке к перезапуску. Сделайте любую необходимую очистку. Этот метод вызывается в потоке устройства, а не в потоке графического интерфейса пользователя.

get_device_uid()

Должен возвращать уникальный идентификатор для текущего подключенного устройства (вызывается сразу после успешного вызова open()). Вы должны реализовать этот метод, если вы установите ASK_TO_ALLOW_CONNECT = True

ignore_connected_device(*uid*)

Должно игнорировать устройство, идентифицированное uid (результат вызова get_device_uid()) в будущем. Вы должны реализовать этот метод, если вы установите ASK_TO_ALLOW_CONNECT = True. Обратите внимание, что эта функция вызывается сразу после open(), поэтому, если open() кэширует какое-то состояние, драйвер должен сбросить это состояние.

get_user_blacklisted_devices()

Вернуть маппинг uid устройства к понятному имени для всех устройств, которые пользователь попросил игнорировать.

set_user_blacklisted_devices(*devices*)

Задать список идентификаторов устройств, которые должны игнорироваться этим драйвером.

specialize_global_preferences(*device_prefs*)

Реализуйте этот метод, если ваше устройство хочет переопределить конкретную настройку. Вы должны убедиться, что все сайты вызовов, которым требуется переопределение настроек, используют device_prefs[„something“] вместо prefs[„something“]. Ваш метод должен вызывать device_prefs.set_overrides(pref = val, pref = val, …). В настоящее время используется для: управления метаданными (prefs[„manage_device_metadata“])

set_library_info(*library_name*, *library_uuid*, *field_metadata*)

Реализуйте этот метод, если вам нужна информация о текущей библиотеке calibre. Этот метод вызывается при запуске и при изменении библиотеки calibre при подключении.

is_dynamically_controllable()

Вызывается диспетчером устройств при запуске плагинов. Если этот метод возвращает строку, то а) он поддерживает интерфейс динамического управления диспетчера устройств и б) это имя должно использоваться при разговоре с плагином.

Этот метод можно вызвать в потоке графического интерфейса. Драйвер, реализующий этот метод, должен быть потокобезопасным.

start_plugin()

Этот метод вызывается для запуска плагина. Плагин должен начать принимать подключения устройств, если он это делает. Если плагин уже принимает соединения, ничего не делайте.

Этот метод можно вызвать в потоке графического интерфейса. Драйвер, реализующий этот метод, должен быть потокобезопасным.

stop_plugin()

Этот метод вызывается для остановки плагина. Плагин больше не должен принимать соединения и должен очищаться за собой. Вероятно этот метод должен вызывать выключение. Если плагин уже не принимает подключения, ничего не делайте.

Этот метод можно вызвать в потоке графического интерфейса. Драйвер, реализующий этот метод, должен быть потокобезопасным.

get_option(*opt_string*, *default=None*)

Вернуть значение параметра, указанного в opt_string. Этот метод можно вызвать, когда плагин не запущен. Вернуть None, если параметр не существует.

Этот метод можно вызвать в потоке графического интерфейса. Драйвер, реализующий этот метод, должен быть потокобезопасным.

set_option(*opt_string*, *opt_value*)

Установить значение параметра, обозначенного opt_string. Этот метод можно вызвать, когда плагин не запущен.

Этот метод можно вызвать в потоке графического интерфейса. Драйвер, реализующий этот метод, должен быть потокобезопасным.

is_running()

Вернуть True, если плагин запущен, иначе false

Этот метод можно вызвать в потоке графического интерфейса. Драйвер, реализующий этот метод, должен быть потокобезопасным.

synchronize_with_db(*db*, *book_id*, *book_metadata*, *first_call*)

Вызывается во время сопоставления книг, когда книга на устройстве сопоставляется с книгой в базе данных calibre. Метод отвечает за синхронизацию данных с устройства в базу данных calibre (при необходимости).

Метод должен возвращать кортеж с двумя значениями. Первое значение - это набор идентификаторов книг calibre, измененных, если база данных calibre была изменена, или None, если база данных не была изменена. Если первое значение является пустым набором, то метаданные для книги на устройстве обновляются метаданными calibre и возвращаются на устройство, но обновление графического интерфейса этой книги не выполняется. Полезно, если данные calibre верны, но их необходимо отправить на устройство.

Второе значение само по себе является кортежем с двумя значениями. Первое значение в кортеже указывает, следует ли отправлять на устройство формат книги. Цель состоит в том, чтобы разрешить проверку того, что книга на устройстве совпадает с книгой в calibre. Это значение должно быть None, если книга не отправляется, в противном случае возвращается базовое имя файла на устройстве (строка, вроде foobar.epub). Обязательно укажите расширение в имени. Подсистема устройства создаст задание send_books для всех книг, не возвращающих значений not-None. Примечание: кроме последующего извлечения расширения, имя игнорируется в тех случаях, когда устройство использует шаблон для генерации имени файла, в большинстве случаев так и происходит. Второе значение в возвращенном кортеже указывает, датирован ли формат будущей датой. Вернуть True, если это так, в противном случае вернуть False. calibre отобразит для пользователя диалоговое окно со списком всех книг датированных будущим.

Чрезвычайно важно: этот метод вызывается в потоке графического интерфейса. Он должен быть потокобезопасным по отношению к потоку диспетчера устройств.

book_id: идентификатор calibre книги в базе данных. book_metadata: объект метаданных для книги, поступающий с устройства. first_call: True, если это первый вызов во время синхронизации, в противном случае - False.

class calibre.devices.interface.**BookList**(*oncard*, *prefix*, *settings*)

Базовые классы: `list`

Список книг. Каждый объект Book должен иметь поля

1. название
2. авторы

3. размер (размер файла книги)
4. datetime (кортеж времени в формате UTC)
5. путь (путь на устройстве к книге)
6. thumbnail (может быть None). thumbnail - это либо объект str/bytes с данными изображения, либо он должен иметь атрибут image_path, в котором хранится абсолютный (собственный для платформы) путь к изображению.
7. теги (список строк, может быть пустым).

supports_collections()

Вернуть True, если устройство поддерживает коллекции для этого списка книг.

add_book(*book*, *replace_metadata*)

Добавить книгу в список книг. Намерение состоит в том, чтобы поддерживать любые внутренние метаданные устройства. Вернуть True, если необходимо синхронизировать списки книг

remove_book(*book*)

Удалить книгу из списка книг. Исправить любые метаданные устройства одновременно

get_collections(*collection_attributes*)

Вернуть словарь коллекций, созданных из collection_attributes. Каждая запись в словаре имеет название коллекции форм:[список книг]

Список книг отсортирован по названию, за исключением сборников, созданных из серий, в этом случае используется series_index.

Параметры

collection_attributes – Список атрибутов объекта Book

Устройства на базе USB Mass Storage

Базовый класс для таких устройств *calibre.devices.usbms.driver.USBMS* (страница 290). Этот класс, в свою очередь, наследует часть своих функций от своих основ, описанных ниже. Типичный базовый драйвер на основе USBMS выглядит так:

```
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):
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        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*)

Базовые классы: DeviceConfig, *DevicePlugin* (страница 277)

Этот класс обеспечивает логику, общую для всех драйверов для устройств, которые экспортируют себя как запоминающие устройства USB. Предоставляет реализации для установки/извлечения устройств USBMS на всех платформах.

VENDOR_ID = 0

VENDOR_ID может быть целым числом, списком целых чисел или словарем. Если это словарь, то это должен быть словарь словарей в форме:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

Целое число или список целых чисел

BCD = None

BCD может иметь значение None, чтобы не различать устройства на основе BCD, или это может быть список BCD номеров всех устройств, поддерживаемых этим драйвером.

WINDOWS_MAIN_MEM = None

Строка, определяющая основную память устройства в строках идентификатора Windows PnP. Это может быть None, строка, список строк или скомпилированное регулярное выражение.

WINDOWS_CARD_A_MEM = None

Строка, идентифицирующая первую карту устройства в строках идентификатора Windows PnP. Это может быть None, строка, список строк или скомпилированное регулярное выражение.

WINDOWS_CARD_B_MEM = None

Строка, идентифицирующая вторую карту устройства в строках идентификатора Windows PnP. Это может быть None, строка, список строк или скомпилированное регулярное выражение.

OSX_MAIN_MEM_VOL_PAT = None

Используется при обнаружении нового драйвера для устранения неоднозначности основной памяти и карт памяти. Должно быть регулярное выражение, которое соответствует точке монтирования основной памяти, назначенной macOS.

MAX_PATH_LEN = 250

Максимальная длина путей, созданных на устройстве

NEWS_IN_FOLDER = True

Поместите новости в отдельную папку

classmethod model_metadata() → tuple[ModelMetadata, ...]

Метаданные обо всех реальных моделях устройств, поддерживаемых этим драйвером

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Параметры

- **key** – Ключ для разблокировки устройства
- **log_packets** – Если true, поток пакетов к/от устройства логируется.
- **report_progress** – Функция, которая вызывается с % прогресса (число от 0 до 100) для различных задач. Если она вызывается с -1, это означает, что задача не имеет информации о ходе выполнения.
- **detected_device** – Device information from the device scanner

set_progress_reporter(*report_progress*)

Установить функцию для сообщения информации о ходе выполнения.

Параметры

report_progress – Функция, которая вызывается с % прогресса (число от 0 до 100) для различных задач. Если она вызывается с -1, это означает, что задача не имеет информации о ходе выполнения.

card_prefix(*end_session=True*)

Возвращает список из 2 элементов префикса для путей на карточках. Если карта отсутствует, для префикса карты устанавливается значение None. НАПРИМЕР („/place“, „/place2“) (None, „place2“) („place“, None) (None, None)

total_space(*end_session=True*)

Получить общее пространство, доступное на точках монтирования:

1. Основная память
2. Карта памяти A
3. Карта памяти B

Результат

Список из трех элементов с общим объёмом в байтах (1, 2, 3). Если на конкретном устройстве нет ни одного из этих местоположений, оно должно вернуть 0.

free_space(*end_session=True*)

Получить свободное место на точках монтирования:

1. Основная память
2. Карта 1
3. Карта 2

Результат

Список из трех элементов со свободным пространством в байтах (1, 2, 3). Если конкретное устройство не имеет ни одного из этих местоположений, оно должно вернуть -1.

windows_sort_drives(*drives*)

Вызывается для устранения неоднозначности основной памяти и карты памяти для устройств, которые не различают их на основе *WINDOWS_CARD_NAME*. Например: EB600

can_handle_windows(*usbdevice*, *debug=False*)

Необязательный метод для выполнения дальнейших проверок устройства, чтобы узнать, способен ли этот драйвер обрабатывать его. Если это не так, он должен вернуть False. Этот метод вызывается только после того, как поставщик, идентификаторы продуктов и bcd совпадают, поэтому он может выполнять некоторые относительно трудоемкие проверки. Реализация по умолчанию возвращает True. Этот метод вызывается только в Windows. См. также `can_handle()`.

Для устройств на базе USBMS этот метод по умолчанию делегирует `can_handle()`. Таким образом, вам нужно только переопределить `can_handle()` в подклассе USBMS.

Параметры

usbdevice – USB-устройство, возвращенное `calibre.devices.winusb.scan_usb_devices()`

open(*connected_device*, *library_uuid*)

Выполнить любую инициализацию конкретного устройства. Вызывается после обнаружения устройства, но перед любыми другими функциями, которые обмениваются данными с устройством. Например: для устройств, которые представляют себя как запоминающие устройства USB, этот метод будет отвечать за установку устройства или, если устройство было автоматически смонтировано, за определение места его установки. Метод *calibre.devices.usbms.device.Device.open()* (страница 289) имеет реализацию этой функции, которая должна служить хорошим примером для запоминающих устройств USB.

Этот метод может вызвать исключение OpenFeedback для отображения сообщения пользователю.

Параметры

- **connected_device** – Устройство, которое мы пытаемся открыть. Это кортеж из (идентификатор поставщика, идентификатор продукта, bcd, название производителя, название продукта, серийный номер устройства). Однако у некоторых устройств нет серийного номера, и в Windows присутствуют только первые три поля, остальные - None.
- **library_uuid** – UUID текущей библиотеки calibre. Может иметь значение None, если библиотеки нет (например, при использовании из командной строки).

eject()

Размонтировать/извлечь устройство из ОС. Не проверяется есть ли ожидающие задания графического интерфейса пользователя, которым необходимо взаимодействовать с устройством.

ПРИМЕЧАНИЕ. Этот метод нельзя вызывать в том же потоке, что и остальные методы устройства.

post_yank_cleanup()

Вызывается, если пользователь дергает устройство, не извлекая его предварительно.

sanitize_callback(*path*)

Обратный вызов, позволяющий отдельным драйверам устройств отменять очистку пути, используемую `create_upload_path()`.

filename_callback(*default*, *mi*)

Обратный вызов, позволяющий драйверам изменять имя файла по умолчанию, установленное `create_upload_path()`.

sanitize_path_components(*components*)

Выполнить любую специальную очистку устройства в компонентах пути для файлов, которые будут загружены на устройство.

get_annotations(*path_map*)

Разрешить path_map в annotation_map файлов, найденных на устройстве

add_annotation_to_library(*db*, *db_id*, *annotation*)

Добавить аннотацию в библиотеку calibre

class calibre.devices.usbms.cli.**CLI**

class calibre.devices.usbms.driver.**USBMS**(*plugin_path*)

Базовые классы: *CLI* (страница 290), *Device* (страница 287)

Базовый класс для всех устройств USBMS. Реализует логику отправки/получения/обновления метаданных/кеширования метаданных и т. д.

description = 'Соединиться с читалкой.'

Короткая строка, описывающая, что делает этот плагин

author = 'John Schember'

Автор этого плагина

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

booklist_class

псевдоним для `BookList`

book_class

псевдоним для `Book`

FORMATS = []

Упорядоченный список поддерживаемых форматов

CAN_SET_METADATA = []

Можно ли установить метаданные книг через графический интерфейс.

get_device_information(*end_session=True*)

Спросить у устройства информацию об устройстве. См. L{DeviceInfoQuery}.

Результат

(имя устройства, версия устройства, версия программного обеспечения на устройстве, тип MIME). Кортеж может дополнительно иметь пятый элемент, который является словарем информации о накопителе. См. пример usbms.driver.

set_driveinfo_name(*location_code*, *name*)

Задать имя устройства в файле информации о диске „name“. Этот параметр будет действовать до тех пор, пока файл не будет создан заново или имя не будет изменено снова.

Недисковые устройства должны реализовывать этот метод на основе кодов местоположения, возвращаемых методом get_device_information().

books(*oncard=None*, *end_session=True*)

Вернуть список электронных книг на устройстве.

Параметры

oncard – Если „carda“ или „cardb“ возвращают список электронных книг на определенной карте памяти, в противном случае возвращает список электронных книг в основной памяти устройства. Если карта указана и на ней нет книг, вернуть пустой список.

Результат

СписокКниг.

upload_books (*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)

Загрузите список книг на устройство. Если файл уже существует на устройстве, его следует заменить. Этот метод должен вызывать `FreeSpaceError`, если на устройстве недостаточно свободного места. Текст FreeSpaceError должен содержать слово «card», если `on_card` не равно None, в противном случае он должен содержать слово «memory».

Параметры

- **files** – Список путей
- **names** – Список имен файлов книг которые были уже загружены на устройство. len(names) == len(files)
- **metadata** – Если не None, это список объектов `Metadata`. Идея состоит в том, чтобы использовать метаданные, чтобы определить, где на устройстве разместить книгу. len(metadata) == len(files). Помимо обычной обложки (путь к обложке), также может быть атрибут эскиза, который следует использовать в предпочтении. Атрибут эскиза имеет форму (ширина, высота, данные обложки в формате jpeg).

Результат

Список трехэлементных кортежей. Список предназначен для передачи в *add_books_to_metadata()* (страница 291).

upload_cover (*path*, *filename*, *metadata*, *filepath*)

Загрузить обложку книги на устройство. Реализация по умолчанию ничего не делает.

Параметры

- **path** – Полный путь к папке, в которой находится соответствующая книга.
- **filename** – Имя файла книги без расширения.
- **metadata** – метаданные, принадлежащие книге. Используйте metadata.thumbnail для обложки
- **filepath** – Полный путь к файлу электронной книги

add_books_to_metadata (*locations*, *metadata*, *booklists*)

Добавить местоположения в списки книг. Эта функция не должна связываться с устройством.

Параметры

- **locations** – Результат звонка L {upload_books}
- **metadata** – Список объектов `Metadata`, такой же, как для *upload_books()* (страница 290).
- **booklists** – Кортеж, содержащий результат вызовов (`books (oncard = None)()`,: meth:*books(oncard=“carda“)*,: meth`books(oncard=“cardb“)`).

delete_books (*paths*, *end_session=True*)

Удалять книги по путям на устройстве.

remove_books_from_metadata (*paths*, *booklists*)

Удалить книги из списка метаданных. Эта функция не должна связываться с устройством.

Параметры

- **paths** – пути к книгам на устройстве.
- **booklists** – Кортеж, содержащий результат вызовов (`books (oncard = None)()`,: meth:*books(oncard=“carda“)*,: meth`books(oncard=“cardb“)`).

sync_booklists(*booklists*, *end_session=True*)

Обновить метаданные на устройстве.

Параметры

booklists – Кортеж, содержащий результат вызовов (`books (oncard = None)()`,: meth:*books(oncard="carda")*,: meth`books(oncard="cardb")`).

classmethod normalize_path(*path*)

Вернуть путь с нативными разделителями путей платформы

12.1.8 Действия в пользовательском интерфейсе

Если вы добавляете свой собственный плагин в ZIP-файл, вы должны создать подкласс как InterfaceActionBase, так и InterfaceAction. Метод `load_actual_plugin()` вашего подкласса InterfaceActionBase должен возвращать экземпляр объекта вашего подкласса InterfaceBase.

class calibre.gui2.actions.**InterfaceAction**(*parent*, *site_customization*)

Базовые классы: `QObject`

Плагин, представляющий «действие», которое может быть выполнено в графическом пользовательском интерфейсе. Все элементы на панели инструментов и контекстных меню реализованы этими плагинами.

Обратите внимание, что этот класс является базовым классом для этих подключаемых модулей, однако, чтобы интегрировать подключаемый модуль с системой подключаемых модулей calibre, вам необходимо создать класс-оболочку, который ссылается на фактический подключаемый модуль. См. Примеры модуля `calibre.customize.builtins`.

Если два объекта *InterfaceAction* (страница 292) имеют одинаковое имя, приоритет имеет объект с более высоким приоритетом.

Подклассы должны реализовывать методы *genesis()* (страница 294), *library_changed()* (страница 295), *location_selected()* (страница 294), *shutting_down()* (страница 295), *initialization_complete()* (страница 295) и *tag_browser_context_action()* (страница 295)

После инициализации этот плагин получает доступ к графическому интерфейсу основного calibre через член `gui`. Вы можете получить доступ к другим плагинам по имени, например:

```
self.gui.iactions['Save To Disk']
```

Чтобы получить доступ к собственному плагину, используйте атрибут `interface_action_base_plugin`, этот атрибут становится доступным только после инициализации плагина. Полезно, если вы хотите использовать методы из класса плагина, например do_user_config().

QAction, указанный в *action_spec* (страница 292), создается автоматически и становится доступным как self.qaction.

name = 'Implement me'

Название плагина. Если присутствуют два плагина с одинаковым именем, приоритет имеет тот, который имеет более высокий приоритет.

priority = 1

Приоритет плагина. Если присутствуют два плагина с одинаковым именем, приоритет имеет тот, который имеет более высокий приоритет.

popup_type = 1

Тип всплывающего меню, когда этот плагин добавляется на панель инструментов.

auto_repeat = False

Следует ли автоматически повторять это действие при удерживании его сочетания клавиш.

action_spec = ('text', 'icon', None, None)

Форма: (текст, путь к значку, всплывающая подсказка, сочетание клавиш) значок, всплывающая подсказка и сочетание клавиш могут иметь значение None, сочетание клавиш должно быть строкой, None или кортежем сочетаний клавиш. Если None, сочетание клавиш, соответствующее действию, не регистрируется. Если вы передаете пустой кортеж, ярлык регистрируется без привязки клавиш по умолчанию.

action_shortcut_name = None

Если не None, используется для имени, отображаемого пользователю при настройке сочетаний клавиш для указанной выше спецификации действия вместо action_spec[0]

action_add_menu = False

Если True, меню создается автоматически и добавляется в self.qaction.

action_menu_clone_qaction = False

Если True, клон self.qaction добавляется в меню self.qaction. Если вы хотите, чтобы текст этого действия отличался от текста self.qaction, установите для этой переменной новый текст.

dont_add_to = frozenset({})

Набор локаций, в которые нельзя добавлять это действие. См. `all_locations` для списка возможных местоположений.

dont_remove_from = frozenset({})

Набор локаций, из которых нельзя удалять это действие. См. `all_locations` для списка возможных местоположений.

action_type = 'global'

Тип действия „current“ означает действие в текущем представлении, „global“ означает действие, которое действует не на текущее представление, а, скорее, на calibre в целом.

accepts_drops = False

Если True, то у этого InterfaceAction будет возможность взаимодействовать с событиями перетаскивания. Подробнее см. в методах *accept_enter_event()* (страница 293), `accept_drag_move_event()`, *drop_event()* (страница 293).

accept_enter_event(*event*, *mime_data*)

Этот метод должен возвращать True, если это действие интерфейса способно обрабатывать событие перетаскивания. Не вызывайте accept/ignore для события, о котором позаботится пользовательский интерфейс calibre.

accept_drag_move_event(*event*, *mime_data*)

Этот метод должен возвращать True, если это действие интерфейса способно обрабатывать событие перетаскивания. Не вызывайте accept/ignore для события, о котором позаботится пользовательский интерфейс calibre.

drop_event(*event*, *mime_data*)

Этот метод должен выполнять некоторые полезные действия и возвращать True, если это действие интерфейса способно обрабатывать событие drop. Не вызывайте accept/ignore для события, о котором позаботится пользовательский интерфейс calibre. В этой функции не следует выполнять блокирующие/длительные операции. Вместо этого отправьте сигнал или используйте QTimer.singleShot и быстро вернитесь. См. примеры встроенных действий.

create_menu_action(*menu*, *unique_name*, *text*, *icon=None*, *shortcut=None*, *description=None*, *triggered=None*, *shortcut_name=None*, *persist_shortcut=False*)

Удобный метод простого добавления действий в QMenu. Возвращает созданный QAction. Это действие имеет один дополнительный атрибут calibre_shortcut_unique_name, который, если не None, относится к уникальному имени, под которым это действие зарегистрировано диспетчером клавиатуры.

Параметры

- **menu** – QMenu вновь созданное действие будет добавлено в
- **unique_name** – Уникальное имя для этого действия, оно должно быть уникальным во всем мире, поэтому сделайте его как можно более описательным. Если сомневаетесь, добавьте к нему UUID.
- **text** – Текст действия.
- **icon** – Либо QIcon, либо имя файла. Имя файла передается встроенной функции QIcon.ic(), поэтому вам не нужно передавать полный путь к папке с изображениями.
- **shortcut** – Строка, список строк, None или False. Если значение равно False, для этого действия не регистрируется комбинация клавиш. Если None, то регистрируется сочетание клавиш без привязки клавиш по умолчанию. Строка и список строк регистрируют ярлык с привязкой клавиш по умолчанию, как указано.
- **description** – Описание этого действия. Используется для установки всплывающих подсказок.
- **triggered** – Вызываемый объект, связанный с сработавшим сигналом созданного действия.
- **shortcut_name** – Текст, отображаемый пользователю при настройке сочетаний клавиш для этого действия. По умолчанию установлено значение text.
- **persist_shortcut** – Ярлыки для действий, которые не всегда появляются или зависят от библиотеки, могут исчезнуть при редактировании других сочетаний клавиш, если для параметра persist_shortcut установлено значение True.

load_resources(*names*)

Если этот плагин поставляется в виде ZIP-файла (плагин, добавленный пользователем), этот метод позволит вам загружать ресурсы из ZIP-файла.

Например, чтобы загрузить изображение:

```
pixmap = QPixmap()
pixmap.loadFromData(tuple(self.load_resources(['images/icon.png']).
↪values())[0])
icon = QIcon(pixmap)
```

Параметры

names – Список путей к ресурсам в ZIP-файле с использованием разделителя /

Результат

Словарь формы „{name: file_contents}“. Любые имена, которые не были найдены в ZIP-файле, не будут присутствовать в словаре.

genesis()

Установка этого плагина. Вызывается только один раз во время инициализации. self.gui доступен. Действие, указанное в *action_spec* (страница 292), доступно как `self.qaction`.

location_selected(*loc*)

Вызывается при изменении отображаемого списка книг в calibre. В настоящее время значения loc следующие: `library, main, card и cardb`.

Этот метод должен включать/отключать это действие и его вспомогательные действия в зависимости от местоположения.

library_about_to_change(*olddb*, *db*)

Вызывается при изменении текущей библиотеки.

Параметры

- **olddb** – LibraryDatabase, соответствующая предыдущей библиотеке.
- **db** – LibraryDatabase, соответствующая новой библиотеке.

library_changed(*db*)

Вызывается при изменении текущей библиотеки.

Параметры
db – LibraryDatabase, соответствующая текущей библиотеке.

gui_layout_complete()

Вызывается один раз за действие, когда макет основного графического интерфейса завершен. Если ваше действие требует внесения изменений в макет, они должны быть выполнены здесь, а не в *initialization_complete()* (страница 295).

initialization_complete()

Вызывается один раз для каждого действия после завершения инициализации основного графического интерфейса.

tag_browser_context_action(*index*)

Вызывается при отображении контекстного меню в браузере тегов. index - это QModelIndex, который указывает на элемент браузера тегов, нажатый правой кнопкой мыши. Проверьте его на достоверность с помощью index.valid () и получите базовый объект TagTreeItem с помощью index.data (Qt.ItemDataRole.UserRole). Любые объекты действия, полученные этим методом, будут добавлены в контекстное меню.

shutting_down()

Вызывается один раз для каждого плагина, когда основной графический интерфейс находится в процессе завершения работы. Освободите все используемые ресурсы, но постарайтесь не блокировать выключение на длительные периоды времени.

class calibre.customize.**InterfaceActionBase**(**args*, ***kwargs*)

Базовые классы: *Plugin* (страница 264)

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

author = 'Kovid Goyal'

Автор этого плагина

type = 'Действие в пользовательском интерфейсе'

Тип плагина. Используется для категоризации в графической оболочке.

can_be_disabled = False

При False пользователь не сможет отключить этот плагин. Используйте с осторожностью.

load_actual_plugin(*gui*)

Этот метод должен возвращать фактический объект плагина действия интерфейса.

12.1.9 Плагины параметров

class calibre.customize.**PreferencesPlugin**(*plugin_path*)

Базовые классы: *Plugin* (страница 264)

Плагин, представляющий виджет, отображаемый в диалоговом окне «Настройки».

У этого плагина есть только один важный метод *create_widget()* (страница 296). Различные поля плагина определяют его категоризацию в пользовательском интерфейсе.

supported_platforms = ['windows', 'osx', 'linux']

Список платформ, на которых работает этот плагин. Например: `['windows', 'osx', 'linux']`

author = 'Kovid Goyal'

Автор этого плагина

type = 'Параметры'

Тип плагина. Используется для категоризации в графической оболочке.

can_be_disabled = False

При False пользователь не сможет отключить этот плагин. Используйте с осторожностью.

config_widget = None

Путь импорта к модулю, который содержит класс с именем ConfigWidget, который реализует интерфейс ConfigWidgetInterface. Используется *create_widget()* (страница 296).

category_order = 100

Где в списке категорий должна быть *category* (страница 296) этого плагина

name_order = 100

Где в списке имен в категории должен быть *gui_name* (страница 296) этого плагина

category = None

Категория, в которую должен входить этот плагин

gui_category = None

Название категории, отображаемое пользователю для этого плагина.

gui_name = None

Имя, отображаемое пользователю для этого плагина.

icon = None

Значок этого плагина должен быть абсолютным путем

description = None

Описание, используемое для всплывающих подсказок и т.п.

create_widget(*parent=None*)

Создать и вернуть фактический виджет Qt, используемый для установки этой группы предпочтений. Виджет должен реализовывать *calibre.gui2.preferences.ConfigWidgetInterface* (страница 296).

Реализация по умолчанию использует *config_widget* (страница 296) для создания экземпляра виджета.

class calibre.gui2.preferences.**ConfigWidgetInterface**

Этот класс определяет интерфейс, который должны реализовывать все виджеты, отображаемые в диалоговом окне «Настройки». Смотрите *ConfigWidgetBase* (страница 297), чтобы узнать о базовом классе, который реализует этот интерфейс и также определяет различные удобные методы.

changed_signal = None

Этот сигнал должен излучаться всякий раз, когда пользователь изменяет значение в этом виджете.

supports_restoring_to_defaults = True

Установить значение True, если реализован метод `restore_to_defaults()`.

restore_defaults_desc = 'Восстановление параметров по умолчанию. Вы должны нажать кнопку Применить, чтобы сохранить параметры по умолчанию.'

Подсказка для кнопки «Восстановить по умолчанию»

restart_critical = False

Если True, диалоговое окно Preferences не позволит пользователю устанавливать какие-либо другие предпочтения. Действует только в том случае, если *commit()* (страница 297) возвращает True.

genesis(*gui*)

Вызывается один раз перед отображением виджета, должен выполнить все необходимые настройки.

Параметры
gui – Главный графический интерфейс пользователя calibre

initialize()

Следует установить для всех значений конфигурации их начальные значения (значения, хранящиеся в файлах конфигурации). Оператор возврата не является обязательным. Вернуть False, если диалоговое окно не должно отображаться.

restore_defaults()

Должно установить для всех значений конфигурации значения по умолчанию.

commit()

Сохранить все изменённые настройки. Вернуть True, если изменения требуют перезапуска, в противном случае - False. Вызвать исключение `AbortCommit`, чтобы указать, что произошла ошибка. Вы несете ответственность за предоставление пользователю отзыва о том, в чём заключается ошибка и как её исправить.

refresh_gui(*gui*)

Вызывается один раз после фиксации этого виджета. Отвечает за то, что пользовательский интерфейс перечитывает любые измененные настройки. Обратите внимание, что по умолчанию графический интерфейс в любом случае повторно инициализирует различные элементы, поэтому большинству виджетов не нужно использовать этот метод.

initial_tab_changed()

Вызывается, если изначально отображаемая вкладка изменена до показа виджета, но после его инициализации.

class calibre.gui2.preferences.**ConfigWidgetBase**(*parent=None*)

Базовый класс, содержащий код для простого добавления стандартных виджетов конфигурации, таких как флажки, поля со списком, текстовые поля и т. д. См. метод *register()* (страница 298).

Этот класс автоматически обрабатывает уведомление об изменении, сброс до значения по умолчанию, перевод между объектами gui и объектами конфигурации и т. д. для зарегистрированных настроек.

Если ваш виджет конфигурации наследуется от этого класса, но включает в себя незарегистрированные параметры, вам следует переопределить методы *ConfigWidgetInterface* (страница 296) и вызвать методы базового класса внутри переопределений.

changed_signal

Этот сигнал должен излучаться всякий раз, когда пользователь изменяет значение в этом виджете.

supports_restoring_to_defaults = True

Установить значение True, если реализован метод `restore_to_defaults()`.

restart_critical = False

Если True, диалоговое окно Preferences не позволит пользователю устанавливать какие-либо другие предпочтения. Действует только в том случае, если *commit()* (страница 298) возвращает True.

register(*name*, *config_obj*, *gui_name=None*, *choices=None*, *restart_required=False*, *empty_string_is_None=True*, *setting=<class 'calibre.gui2.preferences.Setting'>*)

Зарегистрировать настройки.

Параметры

- **name** – Название настройки
- **config_obj** – Объект конфигурации, который читает/записывает параметр
- **gui_name** – Имя объекта GUI, представляющего интерфейс для изменения настройки. По умолчанию предполагается, что это `'opt_' + name`.
- **choices** – Если этот параметр основан на множественном выборе (поле со списком), то это - список вариантов. Список представляет собой список из двух кортежей элементов формы: `` [(имя графического интерфейса, значение), …] ``
- **setting** – Класс, отвечающий за управление этим параметром. Класс по умолчанию обрабатывает почти все случаи, поэтому этот параметр используется редко.

initialize()

Следует установить для всех значений конфигурации их начальные значения (значения, хранящиеся в файлах конфигурации). Оператор возврата не является обязательным. Вернуть False, если диалоговое окно не должно отображаться.

commit(**args*)

Сохранить все изменённые настройки. Вернуть True, если изменения требуют перезапуска, в противном случае - False. Вызвать исключение `AbortCommit`, чтобы указать, что произошла ошибка. Вы несете ответственность за предоставление пользователю отзыва о том, в чём заключается ошибка и как её исправить.

restore_defaults(**args*)

Должно установить для всех значений конфигурации значения по умолчанию.

12.2 Переменные окружения

- `CALIBRE_CONFIG_DIRECTORY` - установить папку, в которой файлы конфигурации хранятся/читаются.
- `CALIBRE_TEMP_DIR`- установить временную папку, используемую calibre
- `CALIBRE_CACHE_DIRECTORY` - установить папку calibre, используемую для кэширования постоянных данных между сеансами
- `CALIBRE_OVERRIDE_DATABASE_PATH` - указать полный путь к metadata.db. Используя эту переменную, вы можете хранить metadata.db вне папки библиотеки. Полезно, если папка вашей библиотеки находится на сетевом диске, который не поддерживает блокировку файлов.
- `CALIBRE_DEVELOP_FROM` - используется для запуска из среды разработки calibre. Смотрите *Настройка среды разработки calibre* (страница 355).
- `CALIBRE_OVERRIDE_LANG` - используется для принудительного выбора языка интерфейса (ISO 639 код языка)

- `CALIBRE_TEST_TRANSLATION` - используется для тестирования файла перевода .po (должен быть путь к файлу .po)
- `CALIBRE_NO_NATIVE_FILEDIALOGS` - заставляет calibre не использовать собственные файловые диалоги для выбора файлов/папок.
- `CALIBRE_NO_NATIVE_MENUBAR` - заставляет calibre не создавать собственное (глобальное) меню в Ubuntu Unity и аналогичных средах рабочего стола Linux. Вместо этого меню, как обычно, помещается внутри окна.
- `CALIBRE_USE_SYSTEM_THEME` - по умолчанию в Linux calibre использует собственный встроенный стиль Qt. Это сделано для того, чтобы избежать сбоев и зависаний, вызванных несовместимостью между версией Qt calibre, на которой построена версия, и общесистемной Qt. Обратной стороной является то, что calibre может не соответствовать внешнему виду системы. Если вы установите эту переменную среды в Linux, это приведет к тому, что calibre будет использовать системную тему - остерегайтесь сбоев и зависаний.
- `CALIBRE_SHOW_DEPRECATION_WARNINGS` - заставляет calibre печатать предупреждения об устаревании в стандартный вывод. Полезно для разработчиков calibre.
- `CALIBRE_NO_DEFAULT_PROGRAMS` - не позволять calibre автоматически регистрировать типы файлов, с которыми он может работать в Windows.
- `CALIBRE_USE_SYSTEM_CERTIFICATES` - заставьте calibre использовать системное хранилище сертификатов для проверки SSL-сертификатов вместо собственного хранилища сертификатов в Windows и macOS.
- `CALIBRE_NO_ICONS_IN_MENUS` - Отключить значки в меню
- `QT_QPA_PLATFORM` - в Linux установите для этого параметра значение `wayland`, чтобы заставить calibre использовать Wayland, и `xcb`, чтобы принудительно использовать X11.
- `SYSFS_PATH` - использовать, если sysfs монтируется где-то, кроме /sys
- `http_proxy`, `https_proxy` - используется в Linux для указания HTTP(S) прокси

Смотрите Как настроить переменные окружения в Windows[86]. Если вы на macOS можно описать переменные окружения в `~/Library/Preferences/calibre/macos-env.txt` указав по одной переменной окружения на строку, к примеру:

```
CALIBRE_DEVELOP_FROM=$HOME/calibre-src/src
CALIBRE_NO_NATIVE_FILEDIALOGS=1
CALIBRE_CONFIG_DIRECTORY=~/.config/calibre
```

12.3 Твики

Твики — это небольшие изменения, через которые вы можете управлять различными аспектами поведения calibre. Вы можете изменить их, перейдя в меню «Параметры->Расширенные->Твики». Ниже приведены значения настроек по умолчанию

```
#!/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.
```

(продолжается на следующей странице)

[86] https://www.computerhope.com/issues/ch000549.htm

(продолжение с предыдущей страницы)

```
#: 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'
#  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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
# 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 in Tag browser
# Set which author field to display in the Tag browser (the list of authors,
# series, publishers etc on the left hand side). The choices are author and
# author_sort. This tweak affects only what is displayed under the authors
# category in the Tag browser and Content server. Please note that if you set this
# to author_sort, it is very possible to see duplicate names in the list because
# although it is guaranteed that author names are unique, there is no such
# guarantee for author_sort values. Showing duplicates won't break anything, but
# it could lead to some confusion. When using 'author_sort', the tooltip will
# show the author's name.
# Examples:
#   categories_use_field_for_author_name = 'author'
#   categories_use_field_for_author_name = 'author_sort'
categories_use_field_for_author_name = 'author'

#: Control partitioning of Tag browser
# When partitioning the Tag browser, the format of the subcategory label is
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
# 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)
#  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␣
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
↪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
# '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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
# 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+',
                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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        '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
# 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:|(|)}'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
#
# Examples: The first three examples assume that the second tweak
# has not been changed.
#
# 1) I want three series columns to be merged into one set of collections. The
# column lookup names are 'series', '#series_1' and '#series_2'. I want nothing
# in the parenthesis. The value to use in the tweak value would be:
#    sony_collection_renaming_rules={'series':'', '#series_1':'', '#series_2':''}
#
# 2) I want the word '(Series)' to appear on collections made from series, and
# the word '(Tag)' to appear on collections made from tags. Use:
#   sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#
# 3) I want 'series' and '#myseries' to be merged, and for the collection name
# to have '(Series)' appended. The renaming rule is:
#   sony_collection_renaming_rules={'series':'Series', '#myseries':'Series'}
#
# 4) Same as example 2, but instead of having the category name in parentheses
# and appended to the value, I want it prepended and separated by a colon, such
# as in Series: Darkover. I must change the template used to format the category name
#
# The resulting two tweaks are:
#    sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#    sony_collection_name_template='{category:||: }{value}'
sony_collection_renaming_rules = {}
sony_collection_name_template = '{value}{category:| (|)}'

#: Specify how SONY collections are sorted
# Specify how SONY collections are sorted. This tweak is only applicable if
# metadata management is set to automatic. You can indicate which metadata is to
# be used to sort on a collection-by-collection basis. The format of the tweak
# is a list of metadata fields from which collections are made, followed by the
# name of the metadata field containing the sort value.
# Example: The following indicates that collections built from pubdate and tags
# are to be sorted by the value in the custom column '#mydate', that collections
# built from 'series' are to be sorted by 'series_index', and that all other
# collections are to be sorted by title. If a collection metadata field is not
# named, then if it is a series- based collection it is sorted by series order,
# otherwise it is sorted by title order.
# [(['pubdate', 'tags'],'#mydate'), (['series'],'series_index'), (['*'], 'title')]
# Note that the bracketing and parentheses are required. The syntax is
# [ ( [list of fields], sort field ) , ( [ list of fields ] , sort field ) ]
# Default: empty (no rules), so no collection attributes are named.
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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
# library after certain operations such as searches or device insertion. Each
# sort level adds a performance penalty. If the database is large (thousands of
# books) the penalty might be noticeable. If you are not concerned about multi-
# level sorts, and if you are seeing a slowdown, reduce the value of this tweak.
maximum_resort_levels = 5

#: Choose whether dates are sorted using visible fields
# Date values contain both a date and a time. When sorted, all the fields are
# used, regardless of what is displayed. Set this tweak to True to use only
# the fields that are being displayed.
sort_dates_using_visible_fields = False

#: Fuzz value for trimming covers
# The value used for the fuzz distance when trimming a cover.
# Colors within this distance are considered equal.
# The distance is in absolute intensity units.
cover_trim_fuzz_value = 10

#: Control behavior of the book list
# You can control the behavior of double clicks and pressing Enter on the books
# list. Choices: open_viewer, do_nothing, show_book_details,
# show_locked_book_details, edit_cell, edit_metadata. Selecting anything other
# than open_viewer, show_book_details, or show_locked_book_details has the side
# effect of disabling editing a field using a single click.
# Default: open_viewer.
# Example: doubleclick_on_library_view = 'do_nothing'
# You can also control whether the book list scrolls per item or
# per pixel. Default is per item.
doubleclick_on_library_view = 'open_viewer'
enter_key_behavior = 'do_nothing'
horizontal_scrolling_per_column = False
vertical_scrolling_per_row = False

#: Language to use when sorting
# Setting this tweak will force sorting to use the
# collating order for the specified language. This might be useful if you run
# calibre in English but want sorting to work in the language where you live.
# Set the tweak to the desired ISO 639-1 language code, in lower case.
# You can find the list of supported locales at
# https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
# Default: locale_for_sorting = '' -- use the language calibre displays in
# Example: locale_for_sorting = 'fr' -- sort using French rules.
# Example: locale_for_sorting = 'nb' -- sort using Norwegian rules.
locale_for_sorting = ''

#: The number of seconds to wait before sending emails
# The number of seconds to wait before sending emails when using a
# public email server like GMX/Hotmail/Gmail. Default is: 5 minutes
# 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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
# in the list below will be considered a public email server.
public_smtp_relay_delay = 301
public_smtp_relay_host_suffixes = ['gmail.com', 'live.com', 'gmx.com', 'outlook.com']

#: The maximum width and height for covers saved in the calibre library
# All covers in the calibre library will be resized, preserving aspect ratio,
# to fit within this size. This is to prevent slowdowns caused by extremely
# large covers
maximum_cover_size = (1650, 2200)

#: Where to send downloaded news
# When automatically sending downloaded news to a connected device, calibre
# will by default send it to the main memory. By changing this tweak, you can
# control where it is sent. Valid values are "main", "carda", "cardb". Note
# that if there isn't enough free space available on the location you choose,
# the files will be sent to the location with the most free space.
send_news_to_device_location = 'main'

#: Unified toolbar on macOS
# If you enable this option and restart calibre, the toolbar will be 'unified'
# with the titlebar as is normal for macOS applications. However, doing this has
# various bugs, for instance the minimum width of the toolbar becomes twice
# what it should be and it causes other random bugs on some systems, so turn it
# on at your own risk!
unified_title_toolbar_on_osx = False

#: Save original file when converting/polishing from same format to same format
# When calibre does a conversion from the same format to the same format, for
# example, from EPUB to EPUB, the original file is saved, so that in case the
# conversion is poor, you can tweak the settings and run it again. By setting
# this to False you can prevent calibre from saving the original file.
# Similarly, by setting save_original_format_when_polishing to False you can
# prevent calibre from saving the original file when polishing.
save_original_format = True
save_original_format_when_polishing = True

#: Number of recently viewed books to show
# Right-clicking the "View" button shows a list of recently viewed books. Control
# how many should be shown, here.
gui_view_history_size = 15

#: Change the font size of the Book details panel in the interface
# Change the font size at which book details are rendered in the side panel and
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0

#: What format to default to when using the "Unpack book" feature
# The "Unpack book" feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
# 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.
# 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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
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'
#   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
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
#: 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 Переопределение значков, шаблонов и т. д.

Примечание

У calibre есть прямая поддержка тем значков, несколько из этих тем можно использовать, перейдя в *Параметры → Интерфейс → Оформление → Изменить набор значков*. Темы значков используют описанный ниже механизм для переопределения статических ресурсов.

calibre позволяет переопределять статические ресурсы, такие как значки, JavaScript и шаблоны для оболочки метаданных, каталогов и т. д., с помощью настраиваемых версий, которые вам нравятся. Все статические ресурсы хранятся в подпапке ресурсов места установки calibre. В Windows это обычно `C:\Program Files\Calibre2\app\resources`. В macOS `/Applications/calibre.app/Contents/Resources/resources/`. В Linux, если вы используете бинарный установщик с веб-сайта калибра, это будет `/opt/calibre/resources`. Эти пути могут меняться в зависимости от того, где вы решили установить calibre.

Вы не должны изменять файлы в этой папке ресурсов, так как ваши изменения будут перезаписаны при следующем обновлении calibre. Вместо этого перейдите в *Preferences-Advanced-Miscellaneous* и щелкните *Открыть папку конфигурации calibre*. В этой папке конфигурации создайте подпапку с именем resources и поместите в неё файлы, которые вы хотите переопределить. Поместите файлы в соответствующие подпапки, например поместите изображения в `resources/images` и т. д. calibre автоматически будет использовать ваш пользовательский файл вместо встроенного при следующем запуске.

Например, если вы хотите изменить значок для действия :guilabel: *Remove books*, вы сначала должны заглянуть во встроенную папку ресурсов и увидеть, что соответствующий файл `resources/images/remove_books.png `. Предполагая, что у вас есть альтернативный значок в формате PNG с именем :file:`my_remove_books.png`, вы должны сохранить его в папке конфигурации как `resources/images/remove_books.png`. Все значки, используемые пользовательским интерфейсом calibre, находятся в `resources/images` и её подпапках. Размещение файла переопределения здесь будет иметь даже более высокий приоритет, чем пользовательская тема значков.

12.5 Создание собственной темы значков для calibre

Если вы создали красивый набор значков и хотите поделиться ими с другими пользователями calibre с помощью встроенной поддержки тем значков calibre, вы можете легко упаковать свои значки в тему. Для этого перейдите в *Preferences-Miscellaneous-Create icon theme*, выберите папку, в которую вы поместили свои значки (обычно это папка `resources/images` в папке конфигурации calibre, как описано выше) . Затем заполните метаданные темы и нажмите OK. Создастся ZIP-файл, содержащий значки темы. Вы можете загрузить его на форум calibre по адресу Mobileread[87], а затем я сделаю вашу тему доступной через встроенную систему тем значков calibre. По умолчанию только что созданная тема значков также будет установлена как текущая тема в calibre. Если вы тестируете свою тему, не забудьте удалить изображения из папки `resources/images`, чтобы использовались значки из темы.

Начиная с calibre 6, у вас могут быть собственные значки для светлого и тёмного режима. Просто создайте две версии значка и назовите файлы с суффиксом `-for-dark-theme` и `-for-light-theme`. Например, `modified-for-dark-theme.png` и `modified-for-light-theme.png`. Затем calibre автоматически использует соответствующий значок в зависимости от текущей темы.

12.6 Настройка calibre с помощью плагинов

calibre имеет модульную конструкцию. Практически все функциональные возможности calibre поставляются в виде плагинов. Плагины используются для преобразования форматов, для загрузки новостей (хотя и называются рецептами), для различных компонентов пользовательского интерфейса, для подключения к различным устройствам, для обработки файлов при добавлении их в calibre и так далее. Вы можете получить полный список всех встроенных в calibre плагинов, перейдя в *Параметры → Расширенные → Плагины*.

Вы можете писать свои плагины чтобы настроить и расширить поведение calibre. Архитектура плагинов calibre очень проста, ознакомьтесь с руководством *Создание собственных плагинов для расширения функциональности calibre* (страница 232).

После того, как вы написали плагин, вы можете загрузить его на форум плагинов calibre по адресу Mobileread[88], и он будет доступен через встроенный апдейтер плагинов calibre.

[87] https://www.mobileread.com/forums/forumdisplay.php?f=166
[88] https://www.mobileread.com/forums/forumdisplay.php?f=237

ГЛАВА 13

Командная строка

Примечание

В macOS инструменты командной строки находятся в пакете Calibre, например, если вы установили Calibre в `/Applications` инструменты командной строки находятся в `/Applications/calibre.app/Contents/MacOS/`. Так, например, для запуска `ebook-convert` вы можете использовать `/Applications/calibre.app/Contents/MacOS/ebook-convert`.

13.1 Документированные команды

13.1.1 `calibre`

```
calibre [параметры] [path_to_ebook или calibre url ...]
```

Запустить основной графический интерфейс пользователя и, при желании, добавить электронную книгу по path_to_ebook в базу данных. Также возможно указать URL-адреса **`calibre`** для выполнения других действий. Например:

`calibre`://view-book/test_library/1842/epub

откроет книгу с id 1842 в формате EPUB из библиотеки «test_library» в просмотрщике электронных книг. Имена библиотек - это имена папок библиотеки с пробелами, заменёнными подчеркиванием. Полное описание действий на основе URL-адресов приведены в Руководстве пользователя.

Когда вы передаёте аргументы с пробелами в **`calibre`**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--detach

Отвязаться от управляющего терминала (только Linux)

--help, -h

показать это сообщение и выйти

--ignore-plugins

Игнорировать пользовательские плагины. Полезно, если вы установили плагин, препятствующий запуску Calibre

--no-update-check

Не проверять наличие обновлений

--shutdown-running-calibre, -s

Завершает работу запущенного экземпляра calibre, если такой имеется. Обратите внимание, что прервутся все выполняющиеся задания, поэтому используйте с осторожностью.

--start-in-tray

Запускать свёрнутым в системный лоток.

--verbose, -v

Не учитывается, поэтому не используйте. Оставлено только для совместимости

--version

показать версию программы и выйти

--with-library

Использовать библиотеку по указанному пути.

13.1.2 calibre-customize

```
calibre-customize [опции]
```

Настроить calibre с помощью загрузки внешних плагинов.

Когда вы передаёте аргументы с пробелами в **calibre-customize**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--add-plugin, -a

Добавить плагин, указав путь к содержащему его ZIP файлу.

--build-plugin, -b

Для разработчиков плагинов: путь к папке, в которой вы разрабатываете плагин. Эта команда автоматически заархивирует плагин и обновит его в calibre.

--customize-plugin

Настроить плагин. Укажите имя плагина и строку настройки через запятую. Строка настройки такая же, как и при настройке плагина в графическом интерфейсе.

--disable-plugin

Отключить указанный плагин

--enable-plugin

Включить указанный плагин

--help, -h

показать это сообщение и выйти

--list-plugins, -l

Список всех установленных плагинов

--remove-plugin, -r

Удалить пользовательский плагин по имени. Не влияет на встроенные плагины

--version

показать версию программы и выйти

13.1.3 calibre-debug

```
calibre-debug [options]
```

Разные интерфейсы командной строки, полезные для отладки calibre. Без вариантов, эта команда запускает встроенный интерпретатор Python. Также можно запустить основной GUI calibre, просмотрщик электронных книг calibre и редактор calibre в режиме отладки.

Он также содержит интерфейсы для различных опций calibre, для которых нет отдельных инструментов командной строки, например выделение подмножества шрифтов, инструмент сравнения электронных книг и т. д.

Вы также можете использовать **calibre-debug** для запуска автономных скриптов. Для этого используйте его так:

calibre-debug -e myscript.py -- --option1 --option2 file1 file2 …

Всё, что после -- передаётся в скрипт. Вы также можете использовать **calibre-debug** как шебанг (shebang) в скриптах, вот так:

#!/usr/bin/env -S **calibre-debug** -e -- –

Когда вы передаёте аргументы с пробелами в **calibre-debug**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--add-simple-plugin

Добавить простой плагин (т.е. плагин, который состоит только из .py файла), указав путь к .py файлу, содержащему код плагина.

--command, -c

Запустить Python код

--debug-device-driver, -d

Отладка обнаружения устройства

--default-programs

Добавить (убрать) Calibre в список программ по умолчанию Windows *--default-programs* (страница 315) = (register|unregister)

--diff

Запустить инструмент сравнения. Например: calibre-debug *--diff* (страница 315) file1 file2

--edit-book

Запустите инструмент "Редактировать книгу" в режиме отладки.

--exec-file, -e

Запустить Python код в файле.

--explode-book, -x

Взорвать книгу в указанную папку. Использование: -x file.epub output_dir Экспортирует книгу как набор файлов HTML и метаданных, которые можно редактировать с помощью стандартных инструментов редактирования HTML. Работает с файлами EPUB, AZW3, HTMLZ и DOCX.

--export-all-calibre-data

Экспортировать все данные calibre (книги/настройки/плагины). Обычно вас попросят указать папку экспорта и библиотеки для экспорта. Вы также можете указать их в качестве аргументов командной строки, чтобы пропустить вопросы. Используйте абсолютные пути для папки экспорта и библиотек. Специальное ключевое слово all можно использовать для экспорта всех библиотек. Примеры: calibre-debug *--export-all-calibre-data* (страница 316) # для интерактивного использования calibre-debug `--export-all-cale-data` /путь/к/пустому/экспорту/папке /путь/к/библиотеке/папке1 /путь/к/библиотеке2 calibre-debug `--export-all-dication-data` /export/folder all # экспортировать все известные библиотеки

--gui, -g

Запустить графический интерфейс с включенной отладкой. Отладочная информация будет выводиться в потоки stdout и stderr

--gui-debug

Запустить графический интерфейс с отладочной консолью, фиксируя указанный путь. Только для внутреннего использования. Используйте ключ -g чтобы запустить графический интерфейс в отладочном режиме.

--help, -h

показать это сообщение и выйти

--implode-book, -i

Исправить ранее развернутую книгу. Использование: -i output_dir file.epub Импортировать книгу из файлов в output_dir, которые созданы предыдущим вызовом -explode-book. Используйте тот же тип файла, который был при разворачивании.

--import-calibre-data

Импорт ранее экспортированных данных calibre

--inspect-mobi, -m

Посмотреть MOBI файл(файлы) по указанному пути(путях)

--kepubify

Конвертировать указанный файл EPUB в KEPUB без полной конвертации. Это то, что делает драйвер Kobo при отправке файлов на устройство.

--paths

Вывести пути, необходимые для установки окружения для calibre

--run-plugin, -r

Запустить плагин, обеспечивающий интерфейс командной строки. Например: calibre-debug -r "Plugin name" -- file1 `--option1` Всё после -- будет передано плагину в качестве параметров.

--run-test, -t

Запустить упомянутый(-е) тест(-ы). Специальное значение all - запустить все тесты. Если имя теста начинается с точки, предполагается, что это имя модуля. Если имя теста начинается с @, предполагается, что это имя категории.

--run-without-debug

Не запускать с установленным флагом DEBUG

--shutdown-running-calibre, -s

Завершает работу запущенного экземпляра calibre, если такой имеется. Обратите внимание, что прервутся все выполняющиеся задания, поэтому используйте с осторожностью.

--subset-font, -f

Сократить указанный шрифт. Добавьте -- после параметра, чтобы передать параметр программе выделения подмножества шрифта.

--test-build

Тест бинарных модулей в сборке

--un-kepubify

Конвертировать указанный файл KEPUB в EPUB без полной конвертации. Это то, что делает драйвер Kobo при импорте файлов с устройства.

--version

показать версию программы и выйти

--viewer, -w

Запустить читалку в режиме отладки

13.1.4 calibre-server

```
calibre-server [опции] [путь к папке библиотеки…]
```

Запустить контент-сервер calibre. Контент-сервер показывает ваши библиотеки calibre через интернет. Вы можете указать пути к папкам библиотек в качестве аргументов для **calibre-server**. Если вы не укажете какие-либо пути, будут использоваться все библиотеки, о которых знает основная программа calibre.

Когда вы передаёте аргументы с пробелами в **calibre-server**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--access-log

Путь к файлу журнала доступа. Журнал содержит информацию о клиентах, подключающихся к серверу и делающих запросы. По умолчанию журнал доступа не ведётся.

--ajax-timeout

Время (в секундах), ожидания ответа от сервера при выполнении запросов.

--auth-mode

Выбрать тип используемой авторизации. Установка используемого сервером режима проверки подлинности HTTP. Выберите «basic», если размещаете этот сервер за SSL-прокси. Или оставьте «auto», который будет использовать «basic», если SSL настроен, а в противном случае — «digest».

--auto-reload

Автоматическая перезагрузка сервера при изменении исходного кода. Полезно в разработке. Вам также следует указать малое значение таймаута выключения.

--ban-after

Количество неудачных попыток входа в систему. Количество неудачных входов, после которого IP-адрес запрещается

--ban-for

Запретить IP-адреса с повторными ошибками входа. Временно запрещает доступ для IP-адресов с повторными ошибками входа в течение указанного количества минут. Полезно для предотвращения попыток подбора паролей. Если установлено в ноль, запрет не выполняется.

--book-list-mode

Выберите умолчальный режим списка книг. Режим списка книг по умолчанию, используемый для новых пользователей. Отдельные пользователи могут переопределить настройки по умолчанию в своих настройках. По умолчанию используется сетка покрытия.

--compress-min-size

Мин. размер, для использования сжатия данных в ответах (в байтах).

--custom-list-template

Путь к файлу JSON, содержащему шаблон для настраиваемого режима списка книг. Простейший способ создать такой файл шаблона - перейти в calibre Настройки -> Обмен через сеть -> Шаблон списка книг , создать шаблон и экспортировать его.

--daemonize

Запустить процесс в фоне как демон (только Linux).

--displayed-fields

Ограничить отображаемые пользовательские поля. Список разделенных пользователем полей метаданных, которые будут отображаться контент-сервером в представлениях /opds и /mobile. Если вы укажете этот параметр, любые поля, не входящие в этот список, не будут отображаться. Например: my_rating,my_tags

--enable-allow-socket-preallocation, --disable-allow-socket-preallocation

Предварительное выделение сокета, например, systemd активация сокета. По умолчанию эта опция включена.

--enable-auth, --disable-auth

Аутентификация на основе пароля для доступа к серверу. Как правило, сервер неограничен, что позволяет всем получить к нему доступ. Эта опция ограничит доступ предопределенными пользователями. По умолчанию эта опция отключена.

--enable-fallback-to-detected-interface, --disable-fallback-to-detected-interface

Откат к автоматически определяемому интерфейсу. Если по какой-то причине сервер неспособен присоединиться к интерфейсу, заданному в настройке listen_on, он попробует определить интерфейс, который подключён к внешнему миру, и присоединиться к нему. По умолчанию эта опция включена.

--enable-local-write, --disable-local-write

Разрешить не аутентифицированным локальным подключениям вносить изменения. Обычно, если вы не включаете аутентификацию, сервер работает в режиме только для чтения, чтобы анонимные пользователи не вносили изменений в ваши библиотеки. Эта опция позволяет любому пользователю подключаться к серверу calibre для внесения изменений. Небезопасно запускать сервер без проверки подлинности, с использованием calibredb для внесения изменений в ваши библиотеки. Обратите внимание, что включение этой опции означает, что ЛЮБАЯ программа, запущенная на компьютере, может вносить изменения в ваши библиотеки. По умолчанию эта опция отключена.

--enable-log-not-found, --disable-log-not-found

Логировать HTTP 404 (Not Found) запросы. Обычно сервер регистрирует все HTTP-запросы для ресурсов, которые не найдены.Это может привести к большому количеству спама в журнале, если на ваш сервер нацелились боты. Используйте этот параметр, чтобы отключить логирование. По умолчанию эта опция включена.

--enable-use-bonjour, --disable-use-bonjour

Объявлять фиды OPDS через BonJour. Объявлять фиды OPDS через сервис BonJour, чтобы приложения, оперирующие с OPDS могли их определить и подключиться к серверу автоматически. По умолчанию эта опция включена.

--enable-use-sendfile, --disable-use-sendfile

Zero-copy передача файлов для увеличения производительности. Включение zero-copy для передач внутри ядра при отправке файлов по сети с целью увеличения производительности. Однако, это может привести к возникновению ошибок при передаче файлов в неверные файловые системы. Если вы столкнулись с повреждением файлов при передаче, отключите эту опцию. По умолчанию эта опция включена.

--help, -h

показать это сообщение и выйти

--ignored-fields

Игнорировать определённые пользователем поля метаданных. Список разделенных запятой пользовательских полей метаданных, не отображаемые контент-сервером в представлениях /opds и /mobile. Например: my_rating,my_tags

--listen-on

Интерфейс для прослушивания входящих соединений. По умолчанию прослушивание осуществляется на всех доступных интерфейсах IPv6 и IPv4. Вы можете изменить это, например, на "127.0.0.1", чтобы прослушивать только IPv4-соединения с локальной машины, или на "0.0.0.0", чтобы прослушивать все входящие IPv4-соединения.

--log

Путь к файлу журнала сервера. Журнал содержит информацию сервера и ошибки, а не журналы доступа. По умолчанию пишется в stdout.

--manage-users

Управляйте базой данных пользователей, которым разрешено подключаться к этому серверу. Возможно использование в автоматическом режиме добавив –. Подробнее см. calibre-server *--manage-users* (страница 319) -- help. См. также параметр *--userdb* (страница 320).

--max-header-line-size

Макс. размер одного заголовка HTTP (в КБ).

--max-job-time

Макс. время жизни рабочих процессов. Максимально допустимое время жизни запущенного рабочего процесса (в минутах). Установите в ноль для снятия ограничения по времени.

--max-jobs

Максимальное количество рабочих процессов. Рабочие процессы запускаются по мере необходимости для больших заданий, таких как подготовка книги для просмотра, добавление книг, конвертирование и т.п. Обычно макс. количество таких процессов зависит от количества ядер ЦП. Вы можете управлять этим параметром.

--max-log-size

Максимальный размер файла журнала (в МБ). Максимальный размер файлов журнала /лог-файлов/, создаваемых сервером. Как только размер лог-файла превышает заданный, содержимое файла автоматически ротируется. Установка в 0 отключает ротацию.

--max-opds-items

Максимальное число книг в фиде OPDS. Максимальное число книг, которое сервер вернёт в одном принимающем фиде OPDS.

--max-opds-ungrouped-items

Максимальное число несгруппированных элементов в фидах OPDS. Группировать элементы по категориям, таким как автор/теги по первой букве, когда элементов больше, чем это число. Установите в 0 для отключения.

--max-request-body-size

Макс. допустимый размер файлов, загружаемых на сервер (в МБ).

--num-per-page

Количество книг для показа на одной странице. Количество книг, отображаемых на одной странице в браузере.

--pidfile

Записать PID процесса в указанный файл

--port

Порт для входящих соединений.

--search-the-net-urls

Путь к файлу JSON, содержащему URL-адреса для функции Поиск в Интернет. Самый простой способ создать такой файл - перейти в Настройки -> Обмен через сеть -> Поиск в Интернет, создать URL-адреса и экспортировать их.

--shutdown-timeout

Общее время ожидания «чистого» отключения, в секундах .

--ssl-certfile

Путь к файлу сертификата SSL.

--ssl-keyfile

Путь к файлу секретного ключа SSL.

--timeout

Время (в секундах) до отключения холостого соединения.

--trusted-ips

Разрешить неаутентифицированным соединениям с определенных IP-адресов вносить изменения. Обычно, если вы не включаете аутентификацию, сервер работает в режиме только для чтения, чтобы анонимные пользователи не вносили изменений в ваши библиотеки. Эта опция позволяет любому пользователю подключаться к серверу calibre для внесения изменений. Должен быть список адресов или спецификаций сетей через запятую. Небезопасно запускать сервер без проверки подлинности, с использованием calibredb для внесения изменений в ваши библиотеки. Обратите внимание, что включение этой опции означает, что любой, кто подключается с указанных IP-адресов, может вносить изменения в ваши библиотеки.

--url-prefix

Префикс, добавляемый ко всем URL-ам. Полезно, если вы хотите запустить этот сервер за обратным прокси. Например, используйте /calibre в качестве префикса URL.

--userdb

Путь к пользовательской базе данных, используемой для аутентификации. База данных представляет собой SQLite-файл. Для его создания используйте *--manage-users* (страница 319). Вы можете узнать больше об управлении пользователями по адресу: https://manual.calibre-ebook.com/ru/server.html#managing-user-accounts-from-the-command-line-only

--version

показать версию программы и выйти

--worker-count

Количество рабочих потоков для обработки запросов.

13.1.5 calibre-smtp

```
calibre-smtp [опции] [от_кого кому текст]
```

Отправить почту по протоколу SMTP. В **calibre-smtp** два режима работы. В режиме сочинения вы указываете «от_кого», «кому» и «текст» — всё это используется для создания и отправки почтового сообщения. В режиме фильтра **calibre-smtp** считывает полное сообщение электронной почты из STDIN и отправляет его.

«Текст» — это содержимое сообщения электронной почты. Если «текст» не задан, полное сообщение электронной почты считывается из STDIN. «От_кого» — адрес отправителя, а «кому» — адрес электронной почты получателя. Когда полное сообщение читается из STDIN, «от_кого» и «кому» используются только в согласовании SMTP, заголовки сообщения не меняются.

Когда вы передаёте аргументы с пробелами в **calibre-smtp**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--fork, -f

Ветвление и доставка сообщений в фоне. Если вы используете этот вариант, вы должны также использовать *--outbox* (страница 321) для обработки отказов доставки.

--help, -h

показать это сообщение и выйти

--localhost, -l

Имя локального хоста. Используется при подключении к серверу SMTP.

--outbox, -o

Путь к папке почты для хранения недоставленных сообщений.

--timeout, -t

Таймаут для соединения

--verbose, -v

Быть многословнее

--version

показать версию программы и выйти

COMPOSE MAIL

Параметры для составления письма. Игнорируются, если текст не задан

--attachment, -a

Файл для прикрепления к электронному письму

--subject, -s

Тема письма

SMTP RELAY

Параметры для отправки почты через SMTP релей сервер. calibre попытается отправить письмо напрямую, если –relay не задан.

--cafile

Путь к файлу конкатенированных сертификатов CA в формате PEM, используемому для проверки сертификата сервера при использовании TLS. По умолчанию используются системные CA сертификаты.

--dont-verify-server-certificate

Не проверять сертификат сервера при подключении с помощью TLS. Раньше это было поведение по умолчанию в calibre до 3.27. Если вы используется релей с самоподписанным или иным образом недействительным сертификатом, вы можете использовать этот параметр для восстановления поведения до 3.27

--encryption-method, -e

Метод шифрования при подключении к релею. Выбор из TLS, SSL и NONE. По-умолчанию TLS. ВНИМАНИЕ: опция NONE крайне небезопасна

--password, -p

Пароль для релея

--port

Порт подключения к почтовому релей серверу. По-умолчанию 465, если используется SSL шифрование, иначе 25.

--relay, -r

SMTP релей сервер для отправки почты.

--username, -u

Имя для релея

13.1.6 calibredb

```
calibredb команда [опции] [аргументы]
```

calibredb - это интерфейс командной строки к базе данных calibre. Оно имеет несколько подкоманд, описанных ниже.

calibredb может использоваться для управления любой базой данных calibre указывается путь или calibre *Content server*, работающий на на локальном компьютере или через Интернет. Вы можете запустить calibre *Content server* с помощью команды **calibre-server** или в главной программе calibre нажав *Подключить/поделиться → Запустить Content server*. Так как **calibredb** может вносить изменения в ваш calibre, вы должны сначала настроить аутентификацию на сервере. Есть два способа сделать это:

- Если вы планируете подключаться только к серверу, работающему на том же компьютере,

вы можете просто использовать параметр –enable-local-write в контент-сервере, чтобы любая программа, включая calibredb, работала на локальный компьютер, чтобы внести изменения в данные calibre. При запуске сервера из главной программы calibre, эта опция есть в *Preferences-Sharing over the net-Advanced*.

- Если вы хотите разрешить доступ через Интернет, вам следует настроить

учетные записи пользователей на сервере и использовать `--username` (страница 324) и :option:` –password` options to **calibredb**, чтобы предоставить доступ. Вы можете настроить аутентификацию пользователей для **calibre-server** с помощью` –enable-auth` вариант и используя `--manage-users` для создания учетных записей пользователей. Если вы запускаете сервер из главной программы calibre, используйте *Preferences-Sharing over the net-Require username / password*.

Чтобы подключиться к работающему серверу содержимого, передайте URL-адрес сервера в параметр: option: *–with-library*, см. документацию по этому параметру.

- *ГЛОБАЛЬНЫЕ НАСТРОЙКИ* (страница 323)
- *list* (страница 324)
- *add* (страница 325)
  - *ДОБАВЛЕНИЕ ИЗ ПАПОК* (страница 326)
- *remove* (страница 326)
- *add_format* (страница 326)
- *remove_format* (страница 327)
- *show_metadata* (страница 327)
- *set_metadata* (страница 327)
- *export* (страница 328)
- *catalog* (страница 329)
  - *ОПЦИИ EPUB* (страница 329)
- *saved_searches* (страница 331)
- *add_custom_column* (страница 331)
- *custom_columns* (страница 331)
- *remove_custom_column* (страница 332)
- *set_custom* (страница 332)
- *restore_database* (страница 332)
- *check_library* (страница 332)
- *list_categories* (страница 333)
- *backup_metadata* (страница 333)
- *clone* (страница 334)
- *embed_metadata* (страница 334)
- *search* (страница 334)
- *fts_index* (страница 334)
- *fts_search* (страница 335)

ГЛОБАЛЬНЫЕ НАСТРОЙКИ

--help, -h

показать это сообщение и выйти

--library-path, --with-library

Путь к библиотеке calibre. По умолчанию используется путь, сохранённый в настройках. Также можно подключиться к контент-серверу calibre для выполнения действий в удалённых библиотеках. Для этого используйте URL-адрес: http://hostname:port/#library_id, например, http://localhost:8080/#mylibrary.

Library_id - это идентификатор библиотеки библиотеки, к которой вы хотите подключиться, на контент-сервере. Можно использовать специальное значение library_id для получения списка идентификаторов библиотеки, доступных на сервере. Подробнее о настройке доступа через контент-сервер см. в разделе https://manual.calibre-ebook.com/ru/generated/ru/calibredb.html.

--password

Пароль для подключения к контент-серверу calibre. Чтобы прочитать пароль со стандартного ввода, используйте специальное значение: <stdin>. Чтобы прочитать пароль из файла, используйте: <f:/path/to/file> (т.е. <f: за ним следует полный путь к файлу и закрывающий >). Требуются угловые скобки в приведённом выше разделе, не забудьте экранировать их или использовать кавычки в вашем шелле.

--timeout

Тайм-аут в секундах при подключении к библиотеке calibre по сети. По умолчанию - две минуты.

--username

Имя пользователя для соединения с контент-сервером calibre

--version

показать версию программы и выйти

list

```
calibredb список [options]
```

Список книг в базе данных calibre.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--ascending

Сортировать результаты по возрастанию

--fields, -f

Поля, отображаемые при перечислении книг в базе данных. Должен быть разделённый запятыми список полей. Доступные поля: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid По умолчанию: title,authors. Специальное поле "all" можно использовать для выбора всех полей. В дополнение к встроенным полям, пользовательские поля также доступны как *field_name, например, для настраиваемого поля #rating, используйте имя: *rating

--for-machine

Генерирует вывод в формате JSON, более подходящем для машинного разбора. Игнорирует настройки ширины строки и разделителя.

--limit

Максимальное число результатов для отображения. По умолчанию: все

--line-width, -w

Максимальная ширина строки вывода. По умолчанию - размер экрана.

--prefix

Префикс для всех путей. По умолчанию - абсолютный путь к директории библиотеки.

--search, -s

Фильтровать результаты запроса. С форматом запроса можно ознакомиться в соответствующей документации в руководстве пользователя. По умолчанию не фильтровать.

--separator

Разделитель полей. По умолчанию это пробел.

--sort-by

Поле, по которому сортируются результаты. Вы можете указать несколько полей, разделив их запятыми. Доступные поля: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid По умолчанию: id

--template

Шаблон, который нужно запустить, если в списке полей есть "template". Обратите внимание, что шаблоны игнорируются при подключении к серверу calibre. По умолчанию: Нет

--template_file, -t

Путь к файлу, содержащему шаблон для запуска, если в списке полей есть 'template'. По умолчанию: Нет

--template_heading

Заголовок столбца шаблона. По умолчанию: template. Этот параметр игнорируется, если установлен параметр *--for-machine* (страница 324)

add

```
calibredb добавить [параметры] файл1 файл2 файл3 ...
```

Добавить указанные файлы как книги в базу данных. Вы также можете указать папки, см. параметры, относящиеся к папке, ниже.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--authors, -a

Указать авторов добавляемых книг

--automerge, -m

Если будут найдены книги с похожими названиями и авторами, автоматически объединить входящие форматы (файлы) с существующими книжными записями. Значение "ignore" - повторяющиеся форматы отбрасываются. Значение "overwrite" - повторяющиеся форматы в библиотеке перезаписываются вновь добавленными файлами. Значение "new_record" - повторяющиеся форматы помещаются в новую запись книги.

--cover, -c

Путь к обложке для использования в добавленной книге

--duplicates, -d

Добавлять книги в базу данных, даже если они уже есть. Сравнение - по названиям книг и авторам. Параметр *--automerge* (страница 325) имеет приоритет.

--empty, -e

Добавить пустую книгу (книгу без форматов)

--identifier, -I

Задать идентификаторы книги, напр. -I asin:XXX -I isbn:YYY

--isbn, -i

Указать ISBN добавляемых книг

--languages, -l

Список языков через запятую (желательно использовать языковую кодировку ISO639, хотя могут быть распознаны и некоторые обычные названия языков)

--series, -s

Указать серию добавляемых книг

--series-index, -S

Указать номер в серии у добавляемых книг

--tags, -T

Указать теги добавляемых книг

--title, -t

Установить название добавляемой книги (книг)

ДОБАВЛЕНИЕ ИЗ ПАПОК

Опции для управления добавлением книг из директорий. По умолчанию добавляются только файлы с расширениями известных форматов электронных книг.

--add

Шаблон имени файла (glob), файлы, соответствующие этому шаблону, будут добавлены при сканировании папок на наличие файлов, даже если они не относятся к известному типу файлов электронных книг. Можно указать несколько раз для нескольких шаблонов.

--ignore

Шаблон имени файла (glob), файлы, соответствующие этому шаблону, будут игнорироваться при сканировании папок на наличие файлов. Можно указать несколько раз для нескольких шаблонов. Например: *.pdf игнорирует все файлы PDF.

--one-book-per-directory, -1

Предположим, что в каждой папке есть только одна логическая книга и что все файлы в ней представляют собой электронные книги разных форматов этой книги.

--recurse, -r

Рекурсивно обрабатывать папки

remove

```
calibredb remove ids
```

Удалить книги, с определёнными идентификаторами из базы данных. Id должен быть разделенным запятыми списком идентификационных номеров (вы можете получить номера идентификаторов с помощью команды поиска). Например, 23,34,57-85 (при указании диапазона последнее число в диапазоне не включено).

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--permanent

Не использовать Корзина

add_format

```
calibredb add_format [опции] id файл_электронной_книги
```

Добавьте файл_электронной_книги в доступные форматы логической книги, определённой по id. Идентификатор можно получить с помощью команды search (искать). Если формат уже существует, он заменяется, если не указана опция –dont-replace (не заменять).

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--as-extra-data-file

Добавьте файл как дополнительный файл данных в книгу, а не как электронную книгу.

--dont-replace

Не заменять формат, если он уже существует

remove_format

```
calibredb remove_format [параметры] id fmt
```

Удалить формат fmt из логической книги с идентификатором id. Получить id - команда search. fmt должен быть расширением файла, например LRF, или TXT, или EPUB. Если логическая книга не имеет доступного fmt не делать ничего.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

show_metadata

```
calibredb show_metadata [параметры] id
```

Показывает метаданные, хранящиеся в базе данных calibre для книги, заданной идентификатором id. id - числовой идентификатор из команды search.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--as-opf

Напечатать метаданные в OPF из (XML)

set_metadata

```
calibredb set_metadata [options] book_id [/path/to/metadata.opf]
```

Установить метаданные, хранящиеся в базе данных calibre для книги, указанной book_id из файла metadata.opf OPF. book_id — это идентификационный номер книги из команды поиска. Можно получить представление о формате OPF, используя –as-opf переключившись на команду show_metadata. Также можно установить метаданные отдельных полей опцией –field. Если вы используете опцию –field, нет необходимости указывать файл OPF.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--field, -f

Поле для установки. Формат название_поля:значение. Например: *--field* (страница 327) tags:tag1,tag2. Используйте *--list-fields* (страница 327) чтобы получить список названий полей. Вы можете указать эту настройку несколько раз для установки нескольких полей. Помните: для языков вы должны использовать ISO639 языковую кодировку (то есть en для English, fr для French и т.д.). Для идентификаторов, синтаксис *--field* (страница 327) identifiers:isbn:XXXX,doi:YYYYY. Для булевых (да/нет) полей используйте true и false или yes и no.

--list-fields, -l

Список названий полей метаданных, которые могут быть использованы с опцией *--field* (страница 327).

export

```
calibredb export [options] ids
```

Экспортировать книги, указанные идентификаторами (список, разделённый запятыми), в файловую систему. Операция экспорта сохраняет все форматы книги, её обложку и метаданные (в OPF-файл). Любые дополнительные файлы данных, связанные с книгой, также сохраняются. Вы можете получить номера идентификаторов с помощью команды поиска (search).

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--all

Экспортировать все книги в базу данных, игнорируя список id.

--dont-asciiize

calibre преобразует все неанглийские символы в английские эквиваленты имен файлов. Это полезно при сохранении в устаревшую файловую систему без полной поддержки имен файлов Unicode. Указание этого переключателя отключит данный режим.

--dont-save-cover

Обычно, calibre будет сохранять обложку в отдельном файле рядом с файлами электронной книги. Указание этого переключателя отключит данный режим.

--dont-save-extra-files

Сохранить все файлы данных, связанные с книгой, при сохранении книги. Указание этого переключателя отключит данный режим.

--dont-update-metadata

По умолчанию calibre обновляет метаданные в сохранённых файлах в библиотеке. Может замедлить запись на диск. Указание этого переключателя отключит данный режим.

--dont-write-opf

Обычно, calibre будет писать метаданные в отдельный файл OPF рядом с файлом электронной книги. Указание этого переключателя отключит данный режим.

--formats

Список форматов, разделенных запятыми, для сохранения для каждой книги. По умолчанию все доступные форматы сохраняются.

--progress

Прогресс отчёта

--replace-whitespace

Заменить пробел символом подчёркивания.

--single-dir

Экспорт всех книг в одну папку

--template

Шаблон для управления именем файла и структурой папок сохраненных файлов. По умолчанию это "{author_sort}/{title}/{title} - {authors}", при котором книги будут сохраняться в подпапку для каждого автора с именами файлов, содержащими название и автора. Доступные элементы управления: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt

Формат отображения дат. %d - день, %b - месяц, %m - номер месяца, %Y - год. По умолчанию: %b, %Y

--to-dir

Экспорт книг в указанную папку. По умолчанию .

--to-lowercase

Преобразовать пути в нижний регистр.

catalog

```
%prкаталог rog /path/to/destination.(csv|epub|mobi|xml ...) [параметры]
```

Экспортировать каталог в формате, указанном расширением путь/к/назначению. Параметры управляют тем, как записи отображаются в сгенерированном выводе каталога. Обратите внимание, что разные форматы каталогов поддерживают разные наборы параметров. Чтобы просмотреть различные параметры, укажите имя выходного файла, а затем –help опцию.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--ids, -i

Разделённый запятыми список идентификаторов из базы данных для каталога. Если указано, *--search* (страница 329) игнорируется. По умолчанию: all

--search, -s

Фильтровать результаты по поисковому запросу. О формате поискового запроса читайте в связанной с поиском документации в руководстве пользователя. По умолчанию: no filtering (не фильтровать)

--verbose, -v

Показать детальную информацию. Используется для отладки.

ОПЦИИ EPUB

--catalog-title

Название сгенерированного каталога, используемого как название в метаданных. По умолчанию: 'My Books' Применяется к: AZW3, EPUB, MOBI форматам вывода

--cross-reference-authors

Создание перекрестных ссылок в разделе «Авторы» для книг с несколькими авторами. По умолчанию: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

--debug-pipeline

Сохранить вывод из разных этапов конвейера преобразования в указанную папку. Полезно, если вы не уверены, на каком этапе процесса преобразования возникает ошибка. По умолчанию: 'None' Применимо к: выходным форматам AZW3, EPUB, MOBI

--exclude-genre

Regex, описывающий теги для исключения в качестве жанров. По умолчанию: '[.+]|^+$' исключает теги в квадратных скобках, напр. '[Project Gutenberg]', и '+', умолчальный тег для чтения книг. Применяется к: AZW3, EPUB, MOBI форматам вывода

--exclusion-rules

Задать правила, используемые для исключения книг из сгенерированного каталога. Модель для правила исключения: ('<rule name>','Tags','<comma-separated list of tags>') или ('<rule name>','<custom column>','<pattern>'). Например: (('Archived books','#status','Archived'),) исключит книгу со значением 'Archived' в польз. столбце 'status'. Когда определено несколько правил, все они применятся. По умолчанию: '(('Catalogs','Tags','Catalog'),)' Применяется к: AZW3, EPUB, MOBI форматам вывода

--generate-authors

Включить раздел «Авторы» в каталог. По умолчанию: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

--generate-descriptions

Включить раздел «Примечания» в каталог. По умолчанию: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

--generate-genres

Включить раздел «Жанры» в каталог. По умолчанию: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

--generate-recently-added

Включить раздел «Недавно добавленные» в каталог. По умолчанию: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

--generate-series

Включить раздел «Серии» в каталог. По умолчанию: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

--generate-titles

Включить раздел «Названия» в каталог. По умолчанию: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

--genre-source-field

Исходное поле для секции 'Жанры'. Умолчание: 'Теги' Применяется к: AZW3, EPUB, MOBI форматам вывода

--header-note-source-field

Пользовательское поле, содержащее текст примечания для вставки в заголовок описания. Умолчание: '' Применяется к: AZW3, EPUB, MOBI форматам вывода

--merge-comments-rule

#<custom field>: [before|after]:[True|False] с указанием: <custom field> Пользовательское поле, содержащее примечания для объединения с комментариями [before|after] Размещение примечаний относительно комментариев [True|False] - между примечаниями и комментариями вставляется горизонтальная линия. По умолчанию: '::' Применимо к выходным форматам AZW3, EPUB, MOBI

--output-profile

Определить профиль вывода. В некоторых случаях для оптимизации каталога для устройства требуется профиль вывода. Например 'kindle' or 'kindle_dx' создаст структурированное Оглавление с Разделами и Авторами. Умолчание: 'None' Применяется к: AZW3, EPUB, MOBI форматам вывода

--prefix-rules

Задать правила включения префиксов, указывающих чтение книг, элементы списка пожеланий и др. пользх префиксов. Модель для префиксного правила ('<rule name>','<source field>','<pattern>','<prefix>'). Когда определено несколько правил, будет использоваться первое правило сопоставления. Умолчание: '(('Read books','tags','+','✓'),('Wishlist item','tags','Wishlist','×'))' Применяется к: AZW3, EPUB, MOBI форматам вывода

--preset

Использовать именованный пресет, созданный в GUI конструкторе каталогов Пресет задает все настройки для создания каталога. Умолчание: 'None' Применяется к: AZW3, EPUB, MOBI форматам вывода

--thumb-width

Размер подсказки (в дюймах) для обложки книг в каталоге. Диапазон: 1.0 - 2.0 Умолчание: '1.0' Применяется к: AZW3, EPUB, MOBI форматам вывода

--use-existing-cover

Заменить существующую обложку при генерации каталога. Умолчание: 'False' Применяется к: AZW3, EPUB, MOBI форматам вывода

saved_searches

```
calibredb saved_searches [options] (list|add|remove)
```

Управление сохраненными в этой базе данных поисками. При добавлении запроса с уже существующим именем, он будет заменён.

Синтаксис для добавления:

calibredb **saved_searches** add search_name search_expression

Синтаксис для удаления:

calibredb **saved_searches** remove search_name

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

add_custom_column

```
calibredb add_custom_column [опции] ярлык имя тип_данных
```

Создаёт пользовательский столбец, где «ярлык» — удобное для компьютера название столбца, без пробелов и двоеточий; «имя» — удобное для человека название столбца; «тип_данных» один из: bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--display

Словарь настроек для указания того, как данные в столбцах будут интерпретированы. Это JSON строка. Для перечисления столбцов используйте *--display* (страница 331)"{\ "enum_values\ ":[\ "val1\ ", \ "val2\ "]}" Есть много настроек, которые можно использовать в переменной отображения. Эти настройки по типу столбца бывают: составными: composite_template, composite_sort, make_category,contains_html, use_decorations даты и времени: date_format перечисление: enum_values, enum_colors, use_decorations целочисленные или с плавающей точкой: number_format текстовые: is_names, use_decorations Для поиска правильных сочетаний лучше всего создать пользовательский столбец соответствующего типа в GUI и посмотреть резервный OPF книги (убедитесь что новый OPF был создан после добавления столбца). Вы увидите JSON для "отображения" для нового столбца в OPF.

--is-multiple

Этот столбец хранит теги как данные (например, несколько значений, разделенных запятыми). Применяется только, если тип данных текст.

custom_columns

```
calibredb custom_columns [options]
```

Список доступных пользовательских столбцов. Показ меток столбцов и идентификаторы.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--details, -d

Показать подробности для каждой колонки.

remove_custom_column

```
calibredb remove_custom_column [options] labelУдалить пользовательский столбец,␣
↪идентифицированный меткой. Вы можете увидеть доступные столбцы с командой custom_
↪columns.
```

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--force, -f

Не запрашивать подтверждение

set_custom

```
calibredb set_custom [options] column id value
```

Задать значение настраиваемого столбца для книги определяемой по id. Список идентификаторов получается командой поиска. Список пользовательских имен столбцов получается командой custom_columns

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--append, -a

Если столбец содержит несколько значений, добавить указанное значение к существующим, а не заменять их.

restore_database

```
calibredb restore_database [параметры]
```

Восстановить эту базу данных из метаданных, хранящихся в файлах OPF в каждом папка библиотеки calibre. Это полезно, если ваш файл metadata.db был поврежден.

ВНИМАНИЕ: Эта команда полностью регенерирует вашу базу данных. Теряются все сохраненные поисковые запросы, категории пользователей, плагины, сохраненные конвертации для каждой книги настройки и пользовательские рецепты. Восстановленные метаданные будут иметь точность как в файлах OPF.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--really-do-it, -r

Действительно выполнить восстановление. Команда не запустится пока эта опция не определена.

check_library

```
calibredb check_library [настройки]
```

Выполняет некоторые проверки файловой системы представляющей библиотеку. Отчёты invalid_titles, extra_titles, invalid_authors, extra_authors, missing_formats, extra_formats, extra_files, missing_covers, extra_covers, failed_folders

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--csv, -c
Вывод в CSV

--ignore_extensions, -e
Разделённый запятыми список игнорируемых расширений. По умолчанию: все (all).

--ignore_names, -n
Разделённый запятыми список игнорируемых названий. По умолчанию: все (all).

--report, -r
Разделённый запятыми список отчётов. По умолчанию: все (all).

--vacuum-fts-db
Сжать (командой Vacuum) базу данных полнотекстового поиска. Если база данных велика, это может занять много времени и памяти.

list_categories

```
calibredb list_categories [настройки]
```

Генерирует отчёт по категориям в базе данных. Информация соответствует той, что отображается в Браузере тегов.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--categories, -r
Разделённый запятыми список имён поиска категорий. По умолчанию: all

--csv, -c
Вывод в CSV

--dialect
Тип файла CSV для создания. Варианты: excel, excel-tab, unix

--item_count, -i
Выводить только общее число записей в категории вместо поэлементного подсчёта.

--width, -w
Максимальная ширина строки вывода. По умолчанию - размер экрана.

backup_metadata

```
calibredb backup_metadata [параметры]
```

Резервное копирование метаданных, хранящихся в базе данных, в отдельные файлы OPF в каждой папке с книгами. Обычно это происходит автоматически, но вы можете запустить эту команду для принудительного повторного создания файлов OPF с параметром –all.

Обратите внимание, что обычно в этом нет необходимости, так как файлы OPF сохраняются. автоматически, каждый раз при изменении метаданных.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--all
Обычно команда работает только с книгами, имеющими устаревшие OPF файлы. Эта опция заставляет её работать со всеми книгами.

clone

```
calibredb clone path/to/new/library
```

Создать клон текущей библиотеки. Создаётся новая пустая библиотека, содержащая все пользовательские столбцы, виртуальные библиотеки и прочие параметры, аналогично текущей библиотеке.

Клонированная библиотека не содержит книг. Чтобы создать полную копию со всеми книгами просто скопируйте папку библиотеки.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

embed_metadata

```
calibredb embed_metadata [параметры] book_id
```

Обновление метаданных в файлах текущей книги, хранящейся в библиотеке calibre, из метаданных в базе данных calibre. Обычно метаданные обновляются только во время экспорта файлов из calibre, эта команда полезна, если вы хотите, чтобы файлы были обновлены на месте. Заметьте, что различные форматы файлов поддерживают разное количество метаданных. Вы можете использовать специальное значение „all“ для book_id, чтобы обновить метаданные во всех книгах. Вы можете также задать несколько идентификаторов книг, разделённых пробелами, и диапазон идентификаторов, разделённых дефисами. Например: calibredb **embed_metadata** 1 2 10-15 23

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--only-formats, -f

Обновлять только метаданные в файлах заданного формата. Задайте несколько раз для нескольких форматов. По умолчанию, обновляются все форматы.

search

```
calibredb search [опции] поисковое выражение
```

Искать в библиотеке указанный поисковый запрос, возвращая разделенный запятыми список идентификаторов книг, соответствующих поисковому запросу. Формат вывода полезен для подачи в другие команды, которые принимают список идентификаторов в качестве входных данных.

Выражение поиска может быть любым из языка поисковых запросов calibre, например: calibredb **search** author:asimov „title:»i robot»“

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--limit, -l

Максимальное число возвращаемых результатов. По умолчанию - все результаты.

fts_index

```
calibredb fts_index [параметры] enable/disable/status/reindex
```

Управление индексом для полнотекстового поиска.

enable

Включить индексирование для данной библиотеки

disable
Включить индексирование для данной библиотеки

status
Показать состояние индексирования

reindex
Переиндексировать отдельные книги или библиотеку целиком. Для отдельных книг, укажите их идентификаторы после команды reindex. Если идентификаторы не указаны, то переиндексируется вся библиотека.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--indexing-speed
Скорость индексации. Быстрое индексирование использует максимум ресурсов компьютера, медленное - менее ресурсоёмко. Скорость сбрасывается на медленную после каждого вызова.

--wait-for-completion
Ждать, пока все книги будут проиндексированы, периодически показывая ход индексации.

fts_search

```
calibredb fts_search [параметры] выражение поиска
```

Полнотекстовый поиск в библиотеке или ее фрагменте.

Когда вы передаёте аргументы с пробелами в calibredb, заключайте их в кавычки. Например: «/some path/with spaces»

--do-not-match-on-related-words
Точное совпадение вместо родственных слов, напр. correction не совпадает с correcting.

--include-snippets
Выводить фрагменты текста вокруг совпадений. Это существенно замедлит поиск.

--indexing-threshold
Сколько процентов библиотеки должно быть проиндексировано для включения полнотекстового поиска. По умолчанию 90

--match-end-marker
Маркер конца совпавшего слова во фрагменте

--match-start-marker
Маркер начала совпавшего слова во фрагменте

--output-format
Формат вывода результатов: "text" для обычного текста или "json".

--restrict-to
Ограничьте искомые книги, используя поисковое выражение или идентификаторы. Например: ids:1,2,3 для ограничения по идентификаторам или search:tag:foo для ограничения книг с тегом foo.

13.1.7 ebook-convert

```
ebook-convert input_file output_file [параметры]
```

Преобразовать электронную книгу из одного формата в другой.

input_file - это вход, а output_file - это выход. Оба должны быть указаны в качестве первых двух аргументов команды.

Выходной формат электронной книги определяется расширением файла output_file. Выходной_файл также может иметь специальный формат .EXT, где EXT - это расширение выходного файла. В этом случае имя выходного файла происходит от имени входного файла. Обратите внимание, что имена файлов не должны начинаться с дефиса. Наконец, если output_file не имеет расширения, он рассматривается как папка, и в эту папку записывается «открытая электронная книга» (OEB), состоящая из файлов HTML. Эти файлы - это файлы, которые обычно передаются в плагин вывода.

После указания файла ввода и вывода вы можете настроить конвертацию, указав различные параметры. Доступные параметры зависят от типов входных и выходных файлов. Чтобы получить справку по ним, укажите входной и выходной файл, а затем используйте параметр -h.

Полную документацию по системе конвертации см. *Конвертирование книг* (страница 61)

Когда вы передаёте аргументы с пробелами в **ebook-convert**, заключайте их в кавычки. Например: «/some path/with spaces»

Параметры и значения по умолчанию для свойств/ параметров изменяются в зависимости и их от входного и выходного форматов, так что Вы всегда должны выполнять проверку :: ebook-convert myfile.input_format myfile.output_format -h Ниже приведены параметры общие для всех преобразований, а за ними следуют параметры, специфичные для каждого входного и выходного формата.

- *ВХОДНЫЕ НАСТРОЙКИ* (страница 337)
- *ВЫХОДНЫЕ НАСТРОЙКИ* (страница 337)
- *ВИД И ФУНКЦИИ* (страница 338)
- *ЭВРИСТИЧЕСКАЯ ОБРАБОТКА* (страница 341)
- *НАЙТИ И ЗАМЕНИТЬ* (страница 341)
- *ОПРЕДЕЛЕНИЕ СТРУКТУРЫ* (страница 342)
- *ОГЛАВЛЕНИЕ* (страница 343)
- *МЕТАДАННЫЕ* (страница 344)
- *ОТЛАДКА* (страница 345)

--help, -h

показать это сообщение и выйти

--input-profile

Указать профиль ввода. Профиль ввода указывает конвертеру, как интерпретировать различную информацию внутри исходного документа. Например, длина зависит от разрешения (длина в пикселях). Варианты: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes

Список встроенных имен рецептов. Вы можете создать электронную книгу из встроенного рецепта, подобного этому: ebook-convert "Recipe Name.recipe" output.epub

--output-profile

Выбор профиля вывода. Профиль вывода определяет параметры оптимизации преобразуемого документа для выбранного типа устройств. В некоторых случаях (редко), профиль вывода определяет

параметры оптимизации для конкретного устройства. Доступные профили:cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, ipad, ipad3, irexdr1000, irexdr800, jetbook5, kindle, kindle_dx, kindle_fire, kindle_oasis, kindle_pw, kindle_pw3, kindle_scribe, kindle_voyage, kobo, msreader, mobipocket, nook, nook_color, nook_hd_plus, pocketbook_inkpad3, pocketbook_lux, pocketbook_hd, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet, generic_eink, generic_eink_hd, generic_eink_large

--version

показать версию программы и выйти

ВХОДНЫЕ НАСТРОЙКИ

Опции для контроля обработки входного файла mobi

--input-encoding

Укажите кодировку исходного документа. Если установить эту опцию, то любая исходная кодировка будет переопределена. Особенно полезно для документов, в которых кодировка не указана или указана неверно.

ВЫХОДНЫЕ НАСТРОЙКИ

Настройки для контроля обработки выходного файла epub

--dont-split-on-page-breaks

Выключить разбиение при разрыве страниц. Обычно входные файлы автоматически разбиваются на каждый разрыв страницы на два файла. Это дает выходную электронную книгу, которая может анализироваться быстрее и с меньшими ресурсами. Однако разбиение происходит медленно, и если ваш исходный файл содержит очень большое количество разрывов страниц, вы должны отключить разбиение на разрыве страниц.

--epub-flatten

Этот параметр нужен только если EPUB-файл будет использован с FBReaderJ. Файловая структура внутри EPUB файла будет сплющена, все файлы разместятся в корне.

--epub-inline-toc

Добавить встроенное оглавление, которое появится как часть основного содержания книги.

--epub-max-image-size

Максимальный размер изображения (ширина x высота). Значение none означает использование размера экрана из выходного профиля. Значение profile - максимальный размер не указан. Например, значение 100x200 - изменение размера всех изображений так чтобы их ширина была не больше 100 пикселей, а высота - не больше 200 пикселей. Обратите внимание, что это влияет только на размер самих файлов изображений. Любое данное изображение может отображаться в другом размере в зависимости от стиля, примененного к нему в документе.

--epub-toc-at-end

Поместить встроенное оглавление в конце книги, а не в начале.

--epub-version

Версия файла EPUB для генерации. EPUB 2 - наиболее широко совместим, используйте EPUB 3, только если вы знаете, что он вам действительно нужен.

--extract-to

Распакуйте содержимое сгенерированной книги в указанную папку. Содержимое папки сначала удаляется, так что будьте осторожны.

--flow-size

Разбить все HTML файлы, размер которых превышает заданный (в КБ). Это необходимо сделать, поскольку большинство читалок EPUB не могут обрабатывать большие файлы. По умолчанию 260KB определяется как размер, необходимый для Adobe Digital Editions. Установите параметр в 0, чтобы отключить разбиение файлов.

--no-default-epub-cover

Обычно, если входной файл не имеет обложки, и вы её не указали, то по умолчанию обложка генерируется из названия, авторов и т.д. Этот параметр отключает генерацию такой обложки.

--no-svg-cover

Не использовать файл SVG в качестве обложки. Используйте эту опцию только, если файл EPUB будет использоваться на устройстве не поддерживающем SVG, таком как iPhone или JetBook Lite. Без включенной опции, подобные устройства будут отображать обложку в виде пустой страницы.

--preserve-cover-aspect-ratio

При использовании обложки в формате SVG, настройка растянет обложку по всей площади экрана, но сохранит соотношения сторон. Т.е. могут появиться белые поля у границ изображения, но останется без искажений. Без включенной опции, изображение обложки может быть слегка искажено, но у него не будет пустых полей по краям.

--pretty-print

Если отмечено, модуль вывода попытается форматировать вывод в удобном для чтения виде. Может не работать на некоторых модулях вывода.

--toc-title

Заголовок любого сгенерированного встроенного оглавления.

ВИД И ФУНКЦИИ

Опции для контроля за внешним видом выходного файла

--asciiize

Транслитерировать символы Unicode в ASCII. Используйте с осторожностью, символы Unicode будут заменены на ASCII. Например, `'Pelé'` заменится на `'Pele'`. Также обратите внимание, что в случаях, когда существует несколько представлений символа (например, символов, используемых в китайском и японском языках), будет использоваться представление, основанное на текущем языке интерфейса calibre.

--base-font-size

Размер базового шрифта в pt. Все размеры шрифтов в выпущенной книге будут перемасштабированы на основе этого размера. Выбирая больший размер, вы можете увеличить шрифты на выходе и наоборот. По умолчанию, когда значение равно нулю, размер базового шрифта выбирается на основе выбранного профиля вывода.

--change-justification

Изменить выравнивание текста. Значение `"Влево"` преобразует все выравнивание текста из источника в выравнивание по левому краю (то есть не выравненому). Значение `"Выровнять"` преобразует весь не выровненный текст в выровненный. Значение `"оригинал"` (по умолчанию) не изменяет выравнивание в исходном файле. Обратите внимание, что только некоторые выходные форматы поддерживают выравнивание.

--disable-font-rescaling

Запретить масштабирование шрифтов.

--embed-all-fonts

Встроить каждый шрифт, который содержится в исходном документе, но ещё не встроен. Используемые шрифты - искать в вашей системе и встроить их, если найдены. Встраивание будет работать только если формат, в который вы конвертируете, поддерживает встраиваемые шрифты. Например EPUB, AZW3,

DOCX и PDF. Убедитесь, что у вас есть лицензия на встраиваемые шрифты, используемые в этом документе.

--embed-font-family

Встроить определённое семейство шрифтов в книгу. Устанавливает «основной» шрифт книги. Если в исходном документе заданы собственные шрифты, они, вероятно, переопределят основной шрифт. Вы можете использовать опцию фильтрации стилей для удаления шрифтов исходного документа. Заметьте, что встраивание шрифтов работает только с некоторыми выходными форматами, преимущественно EPUB, AZW3 и DOCX.

--expand-css

По умолчанию calibre будет использовать сокращенную форму для различных свойств CSS, таких как margin, padding, border и т.п. Этот параметр включает использование полной расширенной формы. Обратите внимание, что CSS всегда расширяется при создании файлов EPUB с профилем вывода на Nook, поскольку Nook не может обрабатывать сокращенный CSS.

--extra-css

Введите путь к таблице стилей CSS или чистый CSS. Эта CSS будет включена в правила стиля исходного файла, поэтому она может быть использована для переопределения этих правил.

--filter-css

Разделённый запятой список CSS-свойств, которые должны быть удалены изо всех правил CSS. Это полезно, если настройка одного из стилей мешает ему быть изменённым на вашем устройстве. Например: font-family, color, margin-left, margin-right

--font-size-mapping

Сопоставления имён шрифтов и размеров (в пунктах) из CSS. Например, строка настроек выглядит так: 12,12,14,16,18,20,22,24. Это сопоставления для размеров от xx-small до xx-large, с последним размером для огромных шрифтов. Алгоритм масштабирования шрифтов использует эти размеры для правильного масштабирования. По умолчанию используются настройки выбранного профиля.

--insert-blank-line

Вставить пустую строку между абзацами. Не будет работать, если в исходном документе абзацы (теги <p> или) не используются.

--insert-blank-line-size

Установить высоту вставляемых пустых строк (в em). Высота строк между абзацами будет в два раза превышать заданное значение.

--keep-ligatures

Сохраните лигатуры, присутствующие во входном документе. Лигатура — это комбинированный символ пары символов, таких как ff, fi, fl и т. д. Большинство программ чтения не поддерживают лигатуры в шрифтах по умолчанию, поэтому они вряд ли будут отображаться правильно. По умолчанию calibre превратит лигатуру в соответствующую пару обычных символов. Обратите внимание, что под лигатурами здесь подразумеваются только лигатуры Юникода, а не лигатуры, созданные с помощью CSS или стилей шрифта. Вместо этого эта опция сохранит их.

--line-height

Высота строки в пунктах. Определяет расстояние между двумя соседними строками текста. Применяется только к элементам, у которых не определена высота строки. В большинстве случаев лучше всего оставлять минимальное значение. По умолчанию никакие действия с высотой строк не производятся.

--linearize-tables

Некоторые плохо форматированные документы используют таблицы для управления расположением текста на странице. При конвертации этих документов текст зачастую становится неудобочитаемым. Эта опция указывает, что нужно извлекать содержимое из таблиц и представлять его в линейном виде.

--margin-bottom

Установить нижнюю границу в pt (пунктах). Умолчание - 5.0. Установка значения меньше нуля приведет к отсутствию границы (настройка полей в исходном документе будет сохранена). Примечание. Ориентированные на страницы форматы, такие как PDF и DOCX, имеют свои собственные настройки полей.

--margin-left

Установить левую границу в pt (пунктах). Умолчание - 5.0. Установка значения меньше нуля приведет к отсутствию границы (настройка полей в исходном документе будет сохранена). Примечание. Ориентированные на страницы форматы, такие как PDF и DOCX, имеют свои собственные настройки полей.

--margin-right

Установить правую границу в pt (пунктах). Умолчание - 5.0. Установка значения меньше нуля приведет к отсутствию границы (настройка полей в исходном документе будет сохранена). Примечание. Ориентированные на страницы форматы, такие как PDF и DOCX, имеют свои собственные настройки полей.

--margin-top

Установить верхнюю границу в pt (пунктах). Умолчание - 5.0. Установка значения меньше нуля приведет к отсутствию границы (настройка полей в исходном документе будет сохранена). Примечание. Ориентированные на страницы форматы, такие как PDF и DOCX, имеют свои собственные настройки полей.

--minimum-line-height

Минимальная высота строки в процентах от вычисленного размера шрифта элемента. Calibre позаботится о том, чтобы каждый элемент имел высоту строки не менее заданной, независимо от того, что указано во входящем документе. Для отключения задайте ноль. По умолчанию — 120%. Использование данной настройки предпочтительнее, чем прямое указание высоты строки, кроме случая, когда Вы точно понимаете свои действия. Например, «двойной интервал» получается при значении 240.

--remove-paragraph-spacing

Удалить промежутки между абзацами и установить отступ абзаца в 1.5em. Удаление промежутков не будет работать, если в исходном документе абзацы (теги <p> или) не используются.

--remove-paragraph-spacing-indent-size

Когда Calibre удаляет пустые строки между абзацами, то автоматически расставляет красные строки, чтобы абзацы визуально выделялись. Эта опция устанавливает величину отступа (в em). Если установлена отрицательная величина, используется размер отступа, указанный в документе.

--smarten-punctuation

Преобразовать обычные кавычки, тире и троеточия к их типографическим эквивалентам. Более подробно, см. https://daringfireball.net/projects/smartypants

--subset-embedded-fonts

Сократить все внедрённые шрифты. Каждый внедрённый шрифт будет содержать только используемые в документе символы. Это уменьшает размер файла шрифтов. Полезно если вы внедряете объёмный шрифт с большим количеством неиспользуемых символов.

--transform-css-rules

Путь к файлу, содержащему правила для преобразования стилей CSS в этой книге. Самый простой способ создать такой файл - использовать мастер для создания правил в графическом интерфейсе calibre. Получите доступ к нему в разделе Оформление->Стили трансформации из диалогового окна преобразования. После создания правил вы можете использовать кнопку «Экспорт», чтобы сохранить их в файле.

--transform-html-rules

Путь к файлу, содержащему правила трансформации HTML в этой книге. Простейший способ создать такой файл - использовать мастер для создания правил в графическом интерфейсе calibre. Доступ к нему можно получить в разделе «Внешний вид - Преобразование HTML» диалогового окна конвертации. После создания правил вы можете использовать кнопку «Экспорт», чтобы сохранить их в файл.

--unsmarten-punctuation

Преобразовать нестандартные кавычки, тире и многоточия в их обычные эквиваленты.

ЭВРИСТИЧЕСКАЯ ОБРАБОТКА

Изменить текст документа и структуру используя общие шаблоны. По умолчанию выключено. Используйте –enable-heuristics чтобы включить. Отдельные действия могут быть отключены с помощью настройки –disable-*.

--disable-dehyphenate

Анализировать переносы слов в документе. В качестве словаря для определения, оставить или убрать перенос, используется сам документ.

--disable-delete-blank-paragraphs

Убрать пустые абзацы, когда они встречаются между другими абзацами

--disable-fix-indents

Преобразовать отступы, состоящие из повторяющихся неразрывных пробелов, в CSS-отступы.

--disable-format-scene-breaks

Выравненные по левому краю разрывы сцены выравниваются по центру. Заменяет мягкий разрыв сцены из нескольких пустых линий на горизонтальные линии.

--disable-italicize-common-cases

Искать общие слова и шаблоны, обозначающие курсив, и преобразовывать в курсивное написание.

--disable-markup-chapter-headings

Определять неформатированные заголовки разделов и подзаголовки. Менять их на теги h2 и h3. Эта настройка не создаёт оглавление, но может быть использована совместно с функцией определения структуры для его создания.

--disable-renumber-headings

Искать последовательные вхождения тегов <h1> или <h2>. Теги перенумеровываются, чтобы предотвратить разбиение заголовков разделов посередине.

--disable-unwrap-lines

Отмена переноса строк с учётом знаков препинания и другой информации о форматировании.

--enable-heuristics

Разрешить эвристическую обработку. Эту опцию нужно включить для выполнения любого вида эвристической обработки.

--html-unwrap-factor

Коэффициент для определения длины, при которой строка должна быть соединена. Допускаются десятичные значения между 0 и 1. Значение по умолчанию составляет 0,4, чуть меньше средней длины строки. Если только несколько строк в документе требуют соединения, то это значение должно быть уменьшено.

--replace-scene-breaks

Заменить разрывы сцен определённым текстом. По умолчанию используется текст из входного документа.

НАЙТИ И ЗАМЕНИТЬ

Изменить содержание и структуру документа используя пользовательский шаблон.

--search-replace

Путь к файлу, содержащему регулярные выражения для поиска и замены. Файл должен содержать чередующиеся строки регулярного выражения, за которым следует шаблон замены (который может быть пустой строкой). Регулярное выражение должно быть в синтаксисе regex Python, и файл должен быть в кодировке UTF-8.

--sr1-replace

Текст, на который будут заменяться найденные тексты, соответствующие первому выражению поиска.

--sr1-search

Шаблон поиска (регулярное выражение), заменяемый на первый текст замены.

--sr2-replace

Текст, на который будут заменяться найденные тексты, соответствующие второму выражению поиска.

--sr2-search

Шаблон поиска (регулярное выражение), заменяемый на второй текст замены.

--sr3-replace

Текст, на который будут заменяться найденные тексты, соответствующие третьему выражению поиска.

--sr3-search

Шаблон поиска (регулярное выражение), заменяемый на третий текст замены.

ОПРЕДЕЛЕНИЕ СТРУКТУРЫ

Управление автоопределением структуры документа.

--add-alt-text-to-img

Если у <img> тега нет атрибута alt, проверьте связанный файл изображения на наличие метаданных, определяющих альтернативный текст, и используйте их для заполнения атрибута alt. Атрибут alt улучшает доступность, предоставляя текстовые описания для пользователей, которые не могут видеть или полностью интерпретировать визуальный контент.

--chapter

Выражение XPath для определения названий глав. По умолчанию следует учитывать теги <h1> или <h2>, содержащие слова "chapter", "book", "section", "prologue", "epilogue" или "part". В качестве заголовков глав, а также любых тегов, которые имеют класс="chapter". Используемое выражение должно оценивать список элементов. Чтобы отключить обнаружение глав, используйте выражение "/". Более подробную информацию об использовании этой функции см. В учебнике по XPath в руководстве пользователя calibre.

--chapter-mark

Указать, как помечать найденные главы. Значение «pagebreak» будет вставлять разрывы страниц после глав. Значение «rule» будет вставлять горизонтальную линию перед главами. Значение «none» отключит разметку глав, а значение «both» будет вставлять и разрывы страниц, и пустые строки.

--disable-remove-fake-margins

В некоторых документах поля страницы задаются с помощью установки левого и правого поля для каждого отдельного абзаца. Calibre постарается найти и удалить такие поля. Иногда это может привести к удалению тех полей, которые удалять не следует. В таком случае вы можете отключить удаление.

--insert-metadata

Вставить метаданные книги в начале книги. Это полезно, если ваша читалка не поддерживает отображение/поиск метаданных напрямую.

--page-breaks-before

XPath-выражение. Разрывы страниц вставляются перед указанными элементами. Для отключения выражения используйте : /

--prefer-metadata-cover

Использовать обложку, полученную из исходного файла, вместо заданной.

--remove-first-image

Удалить первое изображение из входной электронной книги. Полезно, если входной документ имеет изображение, которое не обозначено как обложка. В этом случае, если вы установите обложку в calibre, выходной документ будет иметь две обложки, если вы не укажете эту опцию.

--start-reading-at

Выражение XPath для определения местоположения в документе, с которого следует начать чтение. Некоторые читалки (в первую очередь Kindle) используют это местоположение в качестве позиции для открытия книги. Дополнительную информацию об этой функции см. в учебнике по XPath в руководстве пользователя calibre.

ОГЛАВЛЕНИЕ

Управление автоматическим созданием оглавления. По умолчанию, если в исходном файле есть оглавление, оно предпочитается созданному автоматически.

--duplicate-links-in-toc

Разрешить дублирующиеся записи при создании оглавления из ссылок в исходном документе, т. е. разрешить записи с одинаковым текстом, если они указывают на разные места.

--level1-toc

XPath-выражение, которое определяет все теги, которые должны быть добавлены в оглавление на первом уровне. Если эта опция определена, то она имеет преимущество над другими видами автоматического обнаружения. Смотрите инструкцию к XPath в руководстве пользователя Calibre, чтобы увидеть примеры.

--level2-toc

XPath-выражение, которое определяет все теги, которые должны быть добавлены в оглавление на втором уровне. Каждая запись добавляется под записью предыдущего уровня. Смотрите инструкцию к XPath в руководстве пользователя Calibre, чтобы увидеть примеры.

--level3-toc

XPath-выражение, которое определяет все теги, которые должны быть добавлены в оглавление на третьем уровне. Каждая запись добавляется под записью предыдущего уровня. Смотрите инструкцию к XPath в руководстве пользователя Calibre, чтобы увидеть примеры.

--max-toc-links

Максимальное количество ссылок, вносимых в оглавление. Поставьте 0 для отключения. По умолчанию 50. Ссылки добавляются в оглавление только если количество найденных глав меньше порогового.

--no-chapters-in-toc

Не добавлять автоматически определённые главы в содержание.

--toc-filter

Удалить из оглавления записи, названия которых соответствуют указанному регулярному выражению. Удалятся также все дочерние элементы таких записей.

--toc-threshold

Если обнаружено меньшее число глав, чем указано, то ссылки добавляются в содержание. По умолчанию: 6

--use-auto-toc

Обычно, если в исходном файле уже есть оглавление, оно предпочитается созданному автоматически. С этой опцией всегда используется автоматически созданное оглавление.

МЕТАДАННЫЕ

Настройки для установки метаданных выходного файла

--author-sort

Строка, используемая при сортировке по авторам.

--authors

Указать авторов. Несколько авторов должны быть разделены амперсандами (&).

--book-producer

Установить создателя книги.

--comments

Установить описание электронной книги.

--cover

Установить обложку для указанного файла или URL

--isbn

Указать ISBN книги.

--language

Установить язык.

--pubdate

Установить дату публикации (предполагается, что дата соответствует местному часовому поясу, если не задано)

--publisher

Установить издателя электронной книги.

--rating

Поставьте оценку. Должна быть числом от 1 до 5.

--read-metadata-from-opf, --from-opf, -m

Прочитать метаданные из указанного OPF-файла. Метаданные, прочитанные таким образом, заменят все метаданные исходного документа

--series

Установить серию этой электронной книги.

--series-index

Указать номер книги в серии.

--tags

Указать теги для книги. Должны быть разделены запятыми.

--timestamp

Установить дату книги (больше нигде не используется)

--title

Указать название.

--title-sort

Версия названия, используемая для сортировки.

ОТЛАДКА

Настройки для облегчения отладки преобразования

--debug-pipeline, -d

Сохранить вывод из разных этапов конвейера конвертации в указанную папку. Полезно, если вы не уверены, на каком этапе процесса конвертации возникает ошибка.

--verbose, -v

Уровень детальности. Используйте несколько раз для большей детальности. Использование дважды приведёт к максимальной детальности, единожды - к средней и ни разу - к наименьшей детальности.

13.1.8 ebook-edit

```
ebook-edit [опции] [путь_к_электронной_книге] [имя_файла_внутри_книги …]
```

Запускает инструмент редактирования книги calibre. Также можно задать имена файлов внутри книги, которые автоматически откроются для редактирования.

Когда вы передаёте аргументы с пробелами в **ebook-edit**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--detach

Отвязаться от управляющего терминала (только Linux)

--help, -h

показать это сообщение и выйти

--select-text

Текст, который нужно выделить в книге, когда она открыта для редактирования.

--version

показать версию программы и выйти

13.1.9 ebook-meta

```
ebook-meta файл_электронной_книги [опции]
```

Читать / писать метаданные из / в файлы электронных книг.

Поддерживаемые форматы для чтения метаданных: 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

Поддерживаемые форматы для записи метаданных: azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, kepub, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

Разные типы файлов поддерживают разные виды метаданных. Если вы попытаетесь задать для файла метаданные, которые он не поддерживает, они будут молча проигнорированы.

Когда вы передаёте аргументы с пробелами в **ebook-meta**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--author-sort

Строка для сортировки по автору. Если не указана, а авторы указаны, будет авто-сгенерирована из списка авторов.

--authors, -a

Вписать авторов. Несколько авторов разделяются символом &. Имена должны быть указаны в порядке: Имя Фамилия.

--book-producer, -k

Установить создателя книги.

--category

Установить категорию книги

--comments, -c

Установить описание электронной книги.

--cover

Указать файл обложки.

--date, -d

Установить дату публикации

--disallow-rendered-cover

Для таких форматов, как EPUB, которые используют 'обложку по умолчанию' для первой отображаемой страницы, отключите такие обложки по умолчанию

--from-opf

Прочитать метаданные из указанного файла OPF и использовать его для установки метаданных электронной книги. Метаданные, указанные в командной строке, будут переопределять метаданные, считанные из файла OPF

--get-cover

Получить обложку из электронной книги и сохранить её как указанный файл.

--help, -h

показать это сообщение и выйти

--identifier

Установить идентификаторы книги, могут быть указаны несколько раз. Например: «*--identifier* (страница 346) uri:https://acme.com *--identifier* (страница 346) isbn:12345». Чтобы удалить идентификатор, не указывайте значение: «*--identifier* (страница 346) isbn:». Обратите внимание, что для файлов EPUB идентификатор, помеченный как идентификатор пакета, не может быть удалён.

--index, -i

Указать номер книги в серии.

--isbn

Указать ISBN книги.

--language, -l

Установить язык.

--lrf-bookid

Указать BookID в файлах LRF

--publisher, -p

Установить издателя электронной книги.

--rating, -r

Поставьте оценку. Должна быть числом от 1 до 5.

--series, -s

Установить серию этой электронной книги.

--tags

Указать теги для книги. Должны быть разделены запятыми.

--title, -t

Указать название.

--title-sort

Версия имени для сортировки. Если отсутствует, а имя указано, то будет автоматически сгенерировано из имени.

--to-opf

Укажите имя файла OPF. Метаданные будут сохранены в этом OPF-файле.

--version

показать версию программы и выйти

13.1.10 ebook-polish

```
   ebook-polish [параметры] входной_файл [выходной_файл]


Полировка книг  – все о том, чтобы придать блеск совершенства
```

Вашим тщательно обработанным электронным книгам.

Полировка пытается минимизировать изменения во внутреннем коде вашей электронной книги. В отличие от преобразования, она не уплощает CSS, переименовывает файлы, меняет шрифт размеры, корректирует поля и т.п. Каждое действие выполняет только минимальный набор изменений, необходимых для желаемого эффекта.

Вы должны использовать этот инструмент в качестве последнего шага в процессе создания электронной книги.

> Обратите внимание, что полировка работает только с файлами в форматах AZW3 или EPUB или KEPUB.

Когда вы передаёте аргументы с пробелами в **ebook-polish**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--add-soft-hyphens, -H

Добавляет мягкие переносы во все слова в книге. Это позволит лучше выглядеть книгам с выровненным по ширине текстом в читалках, не поддерживающих переносы.

--compress-images, -i

Сжать иллюстрации для уменьшения объёма, без ухудшения качества изображения.

--cover, -c

Путь к обложке. Изменяет обложку, указанную в электронной книге. Если обложки нет или она неправильно идентифицирована, вставляется новая обложка.

--download-external-resources, -d

Загрузите внешние ресурсы, такие как изображения, таблицы стилей и т. д., которые указывают на URL-адреса, а не на файлы в книге. Все такие ресурсы будут загружены и добавлены в книгу, чтобы в книге больше не было ссылок на внешние ресурсы.

--embed-fonts, -e

Внедрить шрифты, указанные в документе, но ещё не внедрённые. Будет осуществлен поиск шрифтов, доступных на компьютере; если шрифты будут найдены, то они будут внедрены в документ. Убедитесь в наличии соответствующих прав на внедрение шрифтов, используемых в документе.

--help, -h

показать это сообщение и выйти

--jacket, -j

Вставить в начале книги страницу обложки, которая будет содержать все метаданные: название, теги, авторов, серии, комментарии, и т.д. Любые предыдущие обложки будут заменены.

--opf, -o

Путь к OPF файлу. Метаданные в книге обновлены из OPF файла.

--remove-jacket

Удалить предыдущую добавленную обложку.

--remove-soft-hyphens

Удаляет мягкие переносы из всего текста книги.

--remove-unused-css, -u

Удалить все неиспользуемые CSS правила из таблицы стилей и тегов <style>. Некоторые книги, созданные с помощью технических шаблонов, могут иметь множество дополнительных CSS правил, не совпадающих с реальным содержанием. Это замедляет работу читалок при разборе правил.

--smarten-punctuation, -p

Конвертировать текстовые тире, многоточия, кавычки, множественные переносы и т.д. в их типографически верные эквиваленты. Помните, что алгоритм может иногда выдавать неверные результаты, особенно в случае одиночных кавычек в начале сокращений.

--subset-fonts, -f

Сокращение шрифтов означает уменьшение внедрённых шрифтов до необходимого набора символов, используемых в книге. Это значительно уменьшает размер файлов шрифтов (примерно вдвое). Например, если книга использует специальный шрифт для заголовков, то сокращение шрифтов уменьшит этот шрифт до символов содержащихся только в этих заголовках. Или если в книгу включена курсивная версия шрифта, но текст с ним встречается редко или совсем отсутствует, то курсивный шрифт может быть сокращён всего до нескольких символов или полностью удалён. Единственный недостаток сокращения шрифтов в том, если позже вы пожелаете добавить больше текста в ваши книги, то добавленный текст может быть не охвачен сокращённым шрифтом.

--upgrade-book, -U

Обновление внутренней структуры книги, по возможности. К примеру, обновление книг EPUB 2 до EPUB 3.

--verbose

Более подробный вывод обработки. Полезно при отладке.

--version

показать версию программы и выйти

13.1.11 ebook-viewer

```
ebook-viewer [опции] файл
```

Читать электронную книгу.

Когда вы передаёте аргументы с пробелами в **ebook-viewer**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--continue

Продолжить чтение последней открытой книги

--detach

Отвязаться от управляющего терминала (только Linux)

--force-reload

Принудительная перезагрузка всех открытых книг

--full-screen, --fullscreen, -f

Если указано, окно читалки при запуске попытается открыться во весь экран.

--help, -h

показать это сообщение и выйти

--new-instance

Открыть новое окно просмотрщика, даже если установлена опция использования только одного окна просмотрщика

--open-at

Место открытия указанной книги. Местоположение или позицию можно получить, выполнив действия «Перейти->Место» в элементах управления читалкой. Можно использовать форму toc:нечто, и она откроется в месте первой записи оглавления, содержащей строку «нечто». Форма toc-href:нечто будет соответствовать href (внутренней ссылке на место назначения) элементов оглавления. Соответствие точное, если же хотите сопоставить подстроку, воспользуйтесь формой toc-href-contains:нечто. Форма ref:нечто будет использовать Ссылочный режим ссылок. Форма regex:something будет искать регулярное выражение something после открытия книги.

--raise-window

Если указано, окно читалки при запуске попытается выйти на передний план.

--version

показать версию программы и выйти

13.1.12 fetch-ebook-metadata

```
fetch-ebook-metadata [параметры]
```

Получить метаданные книги из онлайн-источников. Вы должны задать по крайней мере название, авторов или ISBN.

Когда вы передаёте аргументы с пробелами в **fetch-ebook-metadata**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--allowed-plugin, -p

Укажите имя используемого плагина загрузки метаданных. По умолчанию будут использоваться все плагины метаданных. Может быть указано несколько раз для нескольких плагинов. Все имена плагинов: Google, Google Images, Amazon.com, Edelweiss, Open Library, Big Book Search

--authors, -a

Автор(ы) книги

--cover, -c

Указать имя файла. Обложка, если доступна, будет сохранена. Без этого параметра обложка не будет загружена.

--help, -h

показать это сообщение и выйти

--identifier, -I

Идентификаторы, такие как ASIN/Goodreads id и т. д. М.б. указаны несколько раз для нескольких идентификаторов. Например:*--identifier* (страница 350) asin:B0082BAJA0

--isbn, -i

ISBN книги

--opf, -o

Вывести метаданные в OPF формат вместо читаемого текста.

--timeout, -d

Таймаут в секундах. По умолчанию 30

--title, -t

Название книги

--verbose, -v

Вывести лог в консоль (stderr)

--version

показать версию программы и выйти

13.1.13 lrf2lrs

```
lrf2lrs книга.lrf
```

Преобразовать файл LRF в файл LRS (XML, кодировка UTF-8)

Когда вы передаёте аргументы с пробелами в **lrf2lrs**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--dont-output-resources

Не сохранять вложенные картинки и шрифты на диск

--help, -h

показать это сообщение и выйти

--output, -o
Выходной файл LRS

--verbose
Быть многословнее

--version
показать версию программы и выйти

13.1.14 lrfviewer

```
lrfviewer [опции] книга.lrf
```

Читать электронную книгу в формате LRF

Когда вы передаёте аргументы с пробелами в **lrfviewer**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--disable-hyphenation
Отключить переносы. Это должно значительно ускорить визуализацию

--help, -h
показать это сообщение и выйти

--profile
Описать визуализатор LRF

--verbose
Вывести больше информации о процессе рендера

--version
показать версию программы и выйти

--visual-debug
Переключиться на визуальные подсказки для отладки движка визуализации

--white-background
По умолчанию фон светлый, так как я нахожу его более удобным для глаз. Используйте эту настройку, чтобы сделать фон совершенно белым.

13.1.15 lrs2lrf

```
lrs2lrf [опции] файл.lrs
```

Собрать файл LRS в файл LRF.

Когда вы передаёте аргументы с пробелами в **lrs2lrf**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--help, -h
показать это сообщение и выйти

--lrs

Преобразование LRS в LRS, необходимо для отладки.

--output, -o

Путь для файла результата

--verbose

Детализация процесса

--version

показать версию программы и выйти

13.1.16 web2disk

```
web2disk URL
```

Где URL, например: https://google.com

Когда вы передаёте аргументы с пробелами в **web2disk**, заключайте их в кавычки. Например: «/some path/with spaces»

[опции]

--base-dir, -d

Базовая папка, в которой сохраняется URL. По умолчанию .

--delay

Минимальный интервал в секундах между последовательными вызовами. По умолчанию: 0 с

--dont-download-stylesheets

Не скачивать файлы стилей CSS.

--encoding

Кодировка вебсайтов, которые вы собираетесь скачивать. По умолчанию делается попытка определения кодировки.

--filter-regexp

Любая ссылка, совпадающая с этим регулярным выражением, будет игнорироваться. Эта опция может быть определена много раз и пока хоть одно выражение совпадает со ссылкой, она будет игнорироваться. По умолчанию ссылки не игнорируются. Если определены одновременно регулярное выражение фильтра и выражение совпадения, то фильтрующее выражение срабатывает первым.

--help, -h

показать это сообщение и выйти

--match-regexp

Будут скачаны только ссылки, соответствующие этому регулярному выражению. Эту опцию можно указывать несколько раз, в этом случае ссылка будет скачиваться тогда, когда она совпадёт хотя бы с одним из регулярных выражений. По умолчанию, никакие ссылки не скачиваются.

--max-files, -n

Максимальное количество файлов для скачивания. Применимо только к файлам из тегов <a href>. По умолчанию: 9223372036854775807

--max-recursions, -r

Максимальное число уровней вложения, т.е. глубина последовательных ссылок. По умолчанию: 1

--timeout, -t

Максимальное время ожидания ответа от сервера. По умолчанию: 10.0 с

--verbose

Показать детальную информацию. Используется для отладки.

--version

показать версию программы и выйти

13.2 Недокументированные команды

- ebook-device
- markdown-calibre

Вы можете узнать, как использовать недокументированные команды, выполнив их без аргументов в терминале.

ГЛАВА 14

Настройка среды разработки calibre

У calibre полностью открытый исходный код, лицензированный GNU GPL v3[89]. То есть вы можете загружать и изменять программу по собственному желанию. В этом разделе вы узнаете, как настроить среду разработки calibre на выбранной операционной системе. В основном calibre написана на Python[90] со вставками кода на C/C++ для повышения скорости и взаимодействия с системой. Обратите внимание, что calibre требует, как минимум, Python 3.8.

Содержание

- *Философия разработки* (страница 356)
  - *Макет кода* (страница 356)
- *Получение кода* (страница 357)
  - *Отправка ваших изменений для включения* (страница 357)
- *Среда разработки под Windows* (страница 358)
- *Среда разработки под macOS* (страница 359)
- *Среда разработки под Linux* (страница 360)
- *Раздельная установка «обычной» и «разрабатываемой» версий calibre на одном компьютере* (страница 361)
- *Советы по отладке* (страница 361)
  - *Использование print statements* (страница 361)
  - *Использование интерактивного интерпретатора Python* (страница 361)
  - *Использовать отладчик Python как удаленный отладчик* (страница 362)
  - *Использование отладчика в вашей любимой Python IDE* (страница 362)

[89] https://www.gnu.org/licenses/gpl.html
[90] https://www.python.org

  - *Выполнение произвольных скриптов в окружении Python calibre* (страница 363)
- *Использование calibre в собственных проектах* (страница 363)
  - *Установка исполняемого файла calibre* (страница 363)
  - *Установка из исходного кода в Linux* (страница 363)
- *Документация API различных частей calibre* (страница 363)

14.1 Философия разработки

Истоки calibre лежат в мире Unix, что означает высокую модульность её дизайна. Модули взаимодействуют друг с другом через чётко определённые интерфейсы. Это заметно облегчает добавление новых возможностей и исправление ошибок, вызывая бурный темп развития. Из-за своих корней calibre имеет полноценный интерфейс командной строки для всех функций, задокументированный в разделе «*Командная строка* (страница 313)».

Модульная конструкция calibre выражается через «Плагины». Существует tutorial по написанию плагинов calibre. Например, добавление поддержки нового устройства в calibre обычно включает в себя написание менее 100 строк кода в виде плагина драйвера устройства. Вы можете просмотреть встроенные драйверы[91]. Точно так же добавление поддержки новых форматов преобразования включает в себя написание плагинов формата ввода/вывода. Другим примером модульного дизайна является recipe system для получения новостей. Дополнительные примеры плагинов, предназначенных для добавления функций в calibre, см. В разделе «Индекс плагинов <https://www.mobileread.com/forums/showthread.php?p=1362767#post1362767>`_.

14.1.1 Макет кода

Весь Python код calibre находится в пакете `calibre`. Этот пакет содержит следующие основные подпакеты

- devices — все драйверы устройств. Просто просмотрите некоторые встроенные драйверы, чтобы понять, как они работают.
  - Подробнее см.: `devices.interface`, определяющий интерфейс, поддерживаемый драйверами устройств, и `devices.usbms`, определяющий универсальный драйвер, который подключается к устройству USBMS. Все драйверы calibre на базе USBMS наследуются от него.
- e-books — весь код преобразования электронных книг/метаданных. Хорошая отправная точка — `calibre.ebooks.conversion.cli`, который является модулем, включающим команду **`ebook-convert`**. Процесс преобразования контролируется через `translation.plumber`. Код, не зависящий от формата, находится в `ebooks.oeb`, а код, зависящий от формата, — в `ebooks.format_name`.
  - Чтение, запись и загрузка метаданных — всё это в `ebooks.metadata`
  - Преобразование происходит в конвейере (о структуре конвейера см. *Предисловие* (страница 63)). Конвейер состоит из входного плагина, различных трансформаций и выходного плагина. Код, который создаёт и управляет конвейером, находится в `plumber.py`. Конвейер работает над представлением электронной книги, похожей на разархивированный epub, с манифестом, spine, toc, guide, html-содержимым и т. д. Класс, который управляет этим представлением, — OEBBook в `ebooks.oeb.base`, различные трансформации, применяемые к книге во время конвертации, находятся в `oeb/transforms/*.py`. А входные и выходные плагины — в `translation/plugins/*.py`.
  - Редактирование электронной книги происходит с использованием другого объекта контейнера. Это задокументировано в *Документация API инструмента редактирования электронных книг* (страница 375).

[91] https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices

- db — серверная часть базы данных. См.: *Документация API интерфейса базы данных* (страница 363) об интерфейсе библиотеки calibre.
- Контент-сервер: `srv` — это контент-сервер calibre.
- gui2 — графический интерфейс пользователя. Инициализация GUI происходит в `gui2.main` и `gui2.ui`. Программа просмотра электронных книг находится в `gui2.viewer`. Редактор электронных книг находится в `gui2.tweak_book`.

Если вы хотите найти точки входа для всех исполняемых файлов calibre различного уровня, посмотрите на структуру `entry_points` в linux.py[92].

Если вам нужна помощь в понимании кода, отправьте сообщение на форум разработчиков[93], и вы, скорее всего, получите помощь от одного из многих разработчиков calibre.

14.2 Получение кода

Получить исходный код calibre можно двумя способами: при помощи системы управления версиями или прямой загрузкой архива tarball[94].

calibre использует Git[95], распределенную систему контроля версий. Git доступен на всех платформах, поддерживаемых calibre. После установки Git вы можете получить исходный код calibre с помощью команды:

```
git clone https://github.com/kovidgoyal/calibre.git
```

В Windows вам потребуется полный путь, который будет выглядеть примерно так `C:\Program Files\Git\git.exe`.

calibre — это очень большой проект с очень длинной историей управления исходным кодом, поэтому описанное выше может занять некоторое время (от 10 минут до часа в зависимости от скорости вашего интернета).

Хотите получить код быстрее - исходный код последней версии всегда доступен в виде archive[96].

Чтобы обновить ветку до последнего кода, используйте команду:

```
git pull --no-edit
```

Вы также можете просмотреть код на GitHub[97]

14.2.1 Отправка ваших изменений для включения

Если вы планируете внести только несколько небольших изменений, вы можете внести свои изменения и создать «merge directive», которую затем можете прикрепить к заявке в calibre bug tracker[98]. Для этого внесите свои изменения, затем запустите:

```
git commit -am "Comment describing your changes"
git format-patch origin/master --stdout > my-changes
```

Это создаст файл `my-changes` в текущей папке, просто прикрепите его к тикету на трекере ошибок calibre <https://bugs.launchpad.net/calibre>`_. Обратите внимание, что сюда будут включены *все* сделанные вами коммиты. Если вы хотите отправить только несколько коммитов, вам нужно изменить `origin/master` выше. Чтобы отправить только последний коммит, используйте:

[92] https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.py
[93] https://www.mobileread.com/forums/forumdisplay.php?f=240
[94] https://calibre-ebook.com/dist/src
[95] https://www.git-scm.com/
[96] https://calibre-ebook.com/dist/src
[97] https://github.com/kovidgoyal/calibre
[98] https://bugs.launchpad.net/calibre

```
git format-patch HEAD~1 --stdout > my-changes
```

Чтобы отправить последние *n* коммитов, замените *1* на *n*, например, для последних 3 коммитов:

```
git format-patch HEAD~3 --stdout > my-changes
```

Будьте осторожны, чтобы не включать слияния при использовании `HEAD~n`.

Если вы планируете много заниматься разработкой, то лучше всего создать учетную запись GitHub[99]. Ниже приведено основное руководство по настройке собственного форка calibre таким образом, чтобы вы могли отправлять пулл-реквесты для включения в основной репозиторий calibre:

- Настройте git на своём компьютере, как описано в статье: Setup Git[100]
- Настройте ssh-ключи для аутентификации на GitHub, как описано здесь: Генерация SSH-ключей[101]
- Перейдите на https://github.com/kovidgoyal/calibre и нажмите кнопку *Fork*.
- В Терминале делай:

  ```
  git clone git@github.com:<username>/calibre.git
  git remote add upstream https://github.com/kovidgoyal/calibre.git
  ```

  Замените <username> выше своим именем пользователя GitHub. Это позволит проверить ваш форк локально.

- Вы можете вносить изменения и коммитить их в любое время. Когда вы будете готовы объединить свою работу, сделайте:

  ```
  git push
  ```

  и перейдите к `https://github.com/<username>/calibre` и нажмите кнопку *Pull Request*, чтобы сгенерировать пул-реквест, который можно объединить.

- Вы можете в любое время обновить свою локальную копию кодом из основного репозитория, выполнив:

  ```
  git pull upstream
  ```

Вам также следует следить за форумом по разработке calibre[102]. Прежде чем вносить серьезные изменения, вы должны обсудить их на форуме или напрямую связаться с Ковидом (его адрес электронной почты указан по всему исходному коду).

14.3 Среда разработки под Windows

Примечание

Вы также должны получить исходный код calibre отдельно, как описано выше.

Установите calibre обычным образом, используя Windows installer[103]. Затем откройте командную строку и перейдите в ранее проверенную папку кода calibre. Например:

[99] https://github.com
[100] https://help.github.com/articles/set-up-git
[101] https://help.github.com/articles/generating-ssh-keys
[102] https://www.mobileread.com/forums/forumdisplay.php?f=240
[103] https://calibre-ebook.com/download_windows

```
cd C:\Users\kovid\work\calibre
```

calibre - это папка, содержащая подпапки src и resources.

Следующий шаг - установка переменной среды CALIBRE_DEVELOP_FROM = абсолютный путь к папке src. Итак, следуя приведенному выше примеру, это будет `C:\Users\kovid\work\calibre\src`. Вот краткое руководство[104] по настройке переменных среды в Windows.

После того, как вы установили переменную окружения, откройте новую командную строку и проверьте, что она была правильно установлена, используя команду:

```
echo %CALIBRE_DEVELOP_FROM%
```

Установка этой переменной окружения означает, что calibre теперь будет загружать весь свой код Python из указанного места.

Это всё! Теперь вы готовы взломать код calibre. Например, откройте файл `src\calibre\__init__.py` в вашем любимом редакторе и добавьте строку:

```
print("Hello, world!")
```

в верхней части файла. Теперь запустите команду **`calibredb`**. Самая первая строка вывода должна быть `Hello, world!`.

Вы также можете настроить окружение разработки calibre в бесплатной Microsoft Visual Studio, если хотите, следуя инструкциям здесь[105].

14.4 Среда разработки под macOS

> **Примечание**
>
> Вы также должны получить исходный код calibre отдельно, как описано выше.

Установите calibre обычным образом, используя предоставленный .dmg[106]. Затем откройте Terminal и перейдите в ранее проверенную папку кода calibre. Например:

```
cd /Users/kovid/work/calibre
```

calibre - это папка, содержащая подпапки src и resources. Инструменты командной строки calibre находятся внутри пакета приложений calibre, в `/Applications/calibre.app/Contents/MacOS`, вы должны добавить эту папку в переменную среды PATH, если вы хотите легко запускать инструменты командной строки.

Следующий шаг - создание сценария bash, который установит для переменной среды `CALIBRE_DEVELOP_FROM` абсолютный путь к папке src при запуске calibre в режиме отладки.

Создайте простой текстовый файл:

```
#!/bin/sh
export CALIBRE_DEVELOP_FROM="/Users/kovid/work/calibre/src"
calibre-debug -g
```

[104] https://docs.python.org/using/windows.html#excursus-setting-environment-variables
[105] https://www.mobileread.com/forums/showthread.php?t=251201
[106] https://calibre-ebook.com/download_osx

Сохраните этот файл как `/usr/local/bin/calibre-develop`, затем установите ему права доступа так, чтобы он мог выполняться:

```
chmod +x /usr/local/bin/calibre-develop
```

Как только вы это сделали, запустите:

```
calibre-develop
```

При запуске calibre вы должны увидеть диагностическую информацию в окне терминала, а в окне графического интерфейса вы увидите звездочку после номера версии, указывающую, что вы запустились из исходников.

14.5 Среда разработки под Linux

Примечание

Вы также должны получить исходный код calibre отдельно, как описано выше.

calibre в основном разработан на Linux. У вас есть два варианта настройки среды разработки. Вы можете установить бинарный файл calibre как обычно и использовать его в качестве среды выполнения для разработки. Этот подход аналогичен тому, который используется в Windows и macOS. Кроме того, вы можете установить calibre из исходных кодов. Инструкции по настройке среды разработки из исходного кода находятся в файле INSTALL в дереве исходного кода. Здесь мы рассмотрим использование двоичного файла в качестве среды выполнения, что является рекомендуемым методом.

Установите calibre обычным образом, используя binary installer[107]. Затем откройте терминал и перейдите в ранее проверенную папку кода calibre. Например:

```
cd /home/kovid/work/calibre
```

calibre - это папка, содержащая подпапки src и resources.

Следующий шаг - установка переменной среды CALIBRE_DEVELOP_FROM = абсолютный путь к папке src. Итак, следуя приведенному выше примеру, это будет /home/kovid/work/calibre/src. Как установить переменные среды, зависит от вашего дистрибутива Linux и используемой оболочки.

Примечание

Рекомендуется использовать бинарный установщик, предоставленный из апстрима. Если вы настаиваете на использовании пакета, предоставленного вашим дистрибутивом, используйте вместо этого переменные `CALIBRE_PYTHON_PATH` и `CALIBRE_RESOURCES_PATH`

После того, как вы установили переменную окружения, откройте новый терминал и проверьте, что она была правильно установлена, используя команду:

```
echo $CALIBRE_DEVELOP_FROM
```

Установка этой переменной окружения означает, что calibre теперь будет загружать весь свой код Python из указанного места.

Это всё! Теперь вы готовы взломать код calibre. Например, откройте файл `src\calibre\__init__.py` в вашем любимом редакторе и добавьте строку:

[107] https://calibre-ebook.com/download_linux

```
print("Hello, world!")
```

в верхней части файла. Теперь запустите команду **calibredb**. Самая первая строка вывода должна быть `Hello, world!`.

14.6 Раздельная установка «обычной» и «разрабатываемой» версий calibre на одном компьютере

Исходное дерево кодов calibre очень стабильно и редко ломается, но если вы чувствуете необходимость запуска из источника в отдельной тестовой библиотеке и запускаете выпущенную версию calibre с вашей повседневной библиотекой, вы можете легко добиться этого, используя файлы .bat или сценарии оболочки для запуска calibre. В приведенном ниже примере показано, как сделать это в Windows с использованием файлов .bat (инструкции для других платформ такие же, просто используйте сценарий оболочки вместо файла .bat)

Чтобы запустить релизную версию calibre с вашей повседневной библиотекой:

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 Советы по отладке

Python — это динамически типизированный язык с отличными возможностями для самоанализа. Ковид написал код ядра calibre без единого использования отладчика. Существует много стратегий для отладки кода calibre:

14.7.1 Использование print statements

Это любимый способ отладки Ковида. Просто вставьте операторы печати в интересующие места и запустите вашу программу в терминале. Например, вы можете запустить графический интерфейс из терминала как:

```
calibre-debug -g
```

Точно так же вы можете запустить просмотрщик электронных книг как:

```
calibre-debug -w /path/to/file/to/be/viewed
```

Редактор электронных книг может быть запущен так:

```
calibre-debug --edit-book /path/to/be/edited
```

14.7.2 Использование интерактивного интерпретатора Python

Вы можете вставить следующие две строки кода, чтобы начать интерактивный сеанс Python в этой точке:

```
from calibre import ipython
ipython(locals())
```

При запуске из командной строки запускается интерактивный интерпретатор Python с доступом ко всем локально определенным переменным (переменным в локальной области). Интерактивная подсказка даже имеет завершение по `Tab` для свойств объекта, и вы можете использовать различные возможности Python для интроспекции, такие как `dir()`, `type()`, `repr()` и т. д.

14.7.3 Использовать отладчик Python как удаленный отладчик

Вы можете использовать встроенный отладчик Python (pdb) в качестве удаленного отладчика из командной строки. Сначала запустите удаленный отладчик в интересующем вас месте кода calibre, например:

```
from calibre.rpdb import set_trace
set_trace()
```

Затем запустите calibre, как обычно, или с помощью одной из команд отладки calibre, описанных в предыдущем разделе. Как только вышеуказанная точка в коде будет достигнута, calibre остановится, ожидая подключения отладчика.

Теперь откройте терминал или командную строку и используйте следующую команду для запуска сеанса отладки:

```
calibre-debug -c "from calibre.rpdb import cli; cli()"
```

Прочитать о том, как использовать отладчик Python, можно в Документации Python stdlib для модуля pdb (англ.)[108].

Примечание

По умолчанию удаленный отладчик будет пытаться подключиться к порту 4444. Вы можете изменить его, передав параметр порта обеим функциям set_trace() и cli(), описанным выше, например: `set_trace(port=1234)` и `cli(port=1234)`.

Примечание

Отладчик Python не может обрабатывать несколько потоков, поэтому вы должны вызывать set_trace один раз для потока, каждый раз с другим номером порта.

14.7.4 Использование отладчика в вашей любимой Python IDE

Можно использовать встроенный отладчик в вашей любимой среде разработки Python, если он поддерживает удаленную отладку. Первый шаг - добавить проверку calibre src в PYTHONPATH в вашей среде IDE. Другими словами, папка, которую вы установили как CALIBRE_DEVELOP_FROM выше, также должна находиться в PYTHONPATH вашей IDE.

Затем поместите модуль удаленного отладчика IDE в подпапку `src` проверки исходного кода calibre. Добавьте любой код, необходимый для запуска удаленного отладчика до нужного уровня, например, в основной функции. Затем запустите calibre как обычно. Теперь ваша IDE должна иметь возможность подключаться к удаленному отладчику, работающему внутри calibre.

[108] https://docs.python.org/library/pdb.html#debugger-commands

14.7.5 Выполнение произвольных скриптов в окружении Python calibre

Команда **calibre-debug** предоставляет несколько удобных ключей для выполнения собственного кода с доступом к модулям calibre:

```
calibre-debug -c "some Python code"
```

отлично подходит для тестирования небольшого фрагмента кода в командной строке. Он работает так же, как ключ -c для интерпретатора Python:

```
calibre-debug myscript.py
```

может быть использован для выполнения вашего собственного скрипта Python. Он работает так же, как и передача скрипта интерпретатору Python, за исключением того, что окружение calibre полностью инициализировано, поэтому вы можете использовать весь код calibre в своем скрипте. Чтобы использовать аргументы командной строки с вашим скриптом, используйте форму:

```
calibre-debug myscript.py -- --option1 arg1
```

`--` заставляет все последующие аргументы быть переданными в ваш скрипт.

14.8 Использование calibre в собственных проектах

Можно напрямую использовать функции/код calibre в вашем проекте Python. Существует два способа сделать это:

14.8.1 Установка исполняемого файла calibre

Если у вас есть бинарная установка calibre, вы можете использовать интерпретатор Python в комплекте с calibre, например:

```
calibre-debug /path/to/your/python/script.py -- arguments to your script
```

14.8.2 Установка из исходного кода в Linux

Помимо использования описанной выше техники, если вы выполняете установку из исходного кода в Linux, вы также можете напрямую импортировать calibre, как показано ниже:

```
import init_calibre
import calibre

print(calibre.__version__)
```

Важно, чтобы вы импортировали модуль init_calibre перед любыми другими модулями/пакетами calibre, поскольку он настраивает интерпретатор для запуска кода calibre.

14.9 Документация API различных частей calibre

14.9.1 Документация API интерфейса базы данных

Это потокобезопасный API (используется схема блокировки чтения-записи). Получить доступ к API можно так:

```
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

Из плагина, являющегося частью основного графического интерфейса calibre, доступ осуществляется так:

```
db = self.gui.current_db.new_api
```

class calibre.db.cache.**Cache**(*backend*, *library_database_instance=None*)

Кэш в памяти файла metadata.db из библиотеки calibre. Этот класс также служит потокобезопасным API для доступа к базе данных. Кэш в памяти поддерживается в нормальной форме для максимальной производительности.

SQLITE используется просто как способ надежного чтения и записи из metadata.db. Переработана вся логика чтения/сортировки/поиска/кеширования таблиц. Это было необходимо для максимальной производительности и гибкости.

class EventType(**values*)

book_created = 4

Когда в базе данных создается новая запись книги с идентификатором книги в качестве единственного аргумента

book_edited = 8

При редактировании формата книги с аргументами: (book_id, fmt)

books_removed = 5

Когда книги удаляются из базы данных со списком идентификаторов книг в качестве единственного аргумента

format_added = 2

Когда в книгу добавляется формат с аргументами: (book_id, формат)

formats_removed = 3

Когда форматы удаляются из книги, с аргументами: (сопоставление идентификатора книги с набором форматов, удаленных из книги)

indexing_progress_changed = 9

Когда меняется прогресс индексации

items_removed = 7

Когда из некоторых книг удаляются такие элементы, как теги или авторы. Аргументы: (имя_поля, идентификаторы затронутых книг, идентификаторы удаленных элементов)

items_renamed = 6

Когда элементы, такие как теги или авторы, переименовываются в некоторых или во всех книгах. Аргументы: (field_name, затронутые идентификаторы книг, сопоставление старого идентификатора элемента с новым идентификатором элемента)

links_changed = 11

При изменении ссылок, связанных с элементами, с аргументами: (field_name, item_ids)

metadata_changed = 1

Когда некоторые метаданные изменяются для некоторых книг, с аргументами: (имя измененного поля, набор затронутых идентификаторов книг)

notes_changed = 10

Когда примечания, связанные с элементом (ами), изменяются с аргументами: (field_name, item_ids)

add_books(*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)

Добавьте указанные книги в библиотеку. Книги должны быть повторяемыми из 2-х кортежей, каждый из 2-х кортежей в форме `(mi, format_map)`, где mi - объект метаданных, а format_map - словарь в форме `{fmt: path_or_stream}`, например: `{'EPUB': '/path/to/file.epub'}`.

Возвращает пару списков: `ids, duplicates`. „ids“ содержит идентификаторы всех новосозданных книг в базе данных. *duplicates* содержит `(mi, format_map)` для книг уже существующих в базе данных, руководствуясь простой эвристикой обнаружения дубликатов, используемой *has_book()* (страница 370).

add_custom_book_data(*name*, *val_map*, *delete_first=False*)

Добавить данные для имени, где val_map - это отображение идентификаторов книг в значения. Если delete_first имеет значение True, все ранее сохраненные данные для имени будут удалены.

add_extra_files(*book_id*, *map_of_relpath_to_stream_or_path*, *replace=True*, *auto_rename=False*)

Добавить дополнительные файлы данных

add_format(*book_id*, *fmt*, *stream_or_path*, *replace=True*, *run_hooks=True*, *dbapi=None*)

Добавить формат в указанную книгу. Вернёт True, если формат был добавлен успешно.

Параметры

- **replace** – If True replace existing format, otherwise if the format already exists, return False.
- **run_hooks** – If True, file type plugins are run on the format before and after being added.
- **dbapi** – Для внутреннего пользования

add_listener(*event_callback_function*, *check_already_added=False*)

Регистрирует функцию обратного вызова, которая будет вызываться после выполнения определённых действий с этой базой данных. Функция должна принимать три аргумента: (*EventType* (страница 364), library_id, event_type_specific_data)

add_notes_resource(*path_or_stream_or_data*, *name: str*, *mtime: float = None*) → int

Добавить указанный ресурс, чтобы его можно было ссылаться в примечаниях и вернуть его хэш-код

all_annotation_types()

Возвращает кортеж из всех типов аннотаций в базе данных.

all_annotation_users()

Возвращает кортеж из всех (user_type, имя пользователя), которые имеют аннотации.

all_annotations(*restrict_to_user=None*, *limit=None*, *annotation_type=None*, *ignore_removed=False*, *restrict_to_book_ids=None*)

Возвращает кортеж всех аннотаций, соответствующих указанным критериям. *ignore_removed* определяет, будут ли также возвращаться удалённые аннотации. Удалённые аннотации - это просто каркас, используемый для объединения аннотаций.

all_annotations_for_book(*book_id*)

Возвращает кортеж, содержащий все аннотации для указанного book_id в виде dict с ключами: *format*, *user_type*, *user*, *annotation*. Здесь annotation - это данные аннотации.

all_book_ids(*type=<class 'frozenset'>*)

Замороженный набор всех известных идентификаторов книг.

all_field_for(*field*, *book_ids*, *default_value=None*)

То же, что field_for, за исключением того, что работает с несколькими книгами одновременно

all_field_ids(*name*)

Замороженный набор идентификаторов для всех значений в поле `name`.

all_field_names(*field*)

Замороженный набор имен всех полей (следует использовать только для полей „много-один“ и „много-много“)

annotation_count_for_book(*book_id*)

Возвращает количество аннотаций для указанной книги, доступных в базе данных.

annotations_map_for_book(*book_id*, *fmt*, *user_type='local'*, *user='viewer'*)

Возвращает сопоставление тип аннотации -> данные аннотации для указанных book_id, format, user и user_type.

author_data(*author_ids=None*)

Вернуть данные об авторе в виде словаря с ключами: имя, сортировка, ссылка

Если авторов с указанными идентификаторами не найдено, возвращается пустой словарь. Если author_ids - None, возвращаются данные для всех авторов.

author_sort_from_authors(*authors*, *key_func=<function make_change_case_func.<locals>.change_case>*)

Учитывая список авторов, вернуть строку author_sort для авторов, предпочитая сортировку по авторам, связанную с автором, а не вычисленную строку.

books_for_field(*name*, *item_id*)

Вернуть все книги, связанные с элементом, идентифицированным с помощью `item_id`, где элемент принадлежит полю `name`.

Возвращаемое значение - это набор идентификаторов книг или пустой набор, если элемент или поле не существует.

books_in_virtual_library(*vl*, *search_restriction=None*, *virtual_fields=None*)

Вернуть набор книг из указанной виртуальной библиотеки

compress_covers(*book_ids*, *jpeg_quality=100*, *progress_callback=None*)

Сжать изображения обложек для указанных книг. Качество сжатия 100 - сжатие без потерь, иначе - сжатие с потерями.

Коллбэк прогресса будет вызываться с book_id и старым и новым размерами для каждой обработанной книги. При возникновении ошибки новый размер - строка с подробными сведениями об ошибке.

copy_cover_to(*book_id*, *dest*, *use_hardlink=False*, *report_file_size=None*)

Копировать обложку в файл как объект `dest`. Возвращает False, если обложки не существует или если dest - тот же файл, что и текущая обложка. dest также может быть путем, и в этом случае обложка копируется по нему тогда и только тогда, когда путь отличается от текущего пути (с учетом чувствительности к регистру).

copy_format_to(*book_id*, *fmt*, *dest*, *use_hardlink=False*, *report_file_size=None*)

Копировать формат fmt в файл как объект dest. Если указанный формат не существует, вызывает ошибку `NoSuchFormat`. dest также может быть путем (к файлу), и в этом случае формат копируется по нему, если путь отличается от текущего пути (с учетом чувствительности к регистру).

cover(*book_id*, *as_file=False*, *as_image=False*, *as_path=False*, *as_pixmap=False*)

Вернуть обложку или None. По умолчанию возвращает обложку в виде байтовой строки.

ВНИМАНИЕ: использование as_path скопирует обложку во временный файл и вернет путь к временному файлу. Вы должны удалить временный файл, когда закончите работать с ним.

Параметры

- **as_file** – Если True, вернуть изображение как объект открытого файла (SpooledTemporaryFile)
- **as_image** – Если True вернуть изображение как объект QImage
- **as_pixmap** – Если True вернуть изображение как объект QPixmap
- **as_path** – Если True, вернуть изображение как путь, указывающий на временный файл

data_for_find_identical_books()

Вернуть данные, которые можно использовать для реализации *find_identical_books()* (страница 368) в рабочем процессе без доступа к базе данных. См. реализацию в db.utils.

data_for_has_book()

Возвращает данные, подходящие для использования в *has_book()* (страница 370). Это можно использовать для реализации *has_book()* (страница 370) в рабочем процессе без доступа к базе данных.

delete_annotations(*annot_ids*)

Удалить аннотации с указанными идентификаторами.

delete_custom_book_data(*name*, *book_ids=()*)

Удалить данные для имени. По умолчанию удаляются все данные. Если вы хотите удалить данные только для некоторых идентификаторов книг, передайте список идентификаторов книг.

delete_trash_entry(*book_id*, *category*)

Удалить запись из корзины. Здесь категория „b“ для книг и „f“ для форматов.

embed_metadata(*book_ids*, *only_fmts=None*, *report_error=None*, *report_progress=None*)

Обновить метаданные во всех форматах указанных book_ids до текущих метаданных в базе данных.

expire_old_trash()

Записи из корзины слишком старые

export_note(*field*, *item_id*) → str

Экспортировать примечание в виде одного HTML-документа с встроенными изображениями в виде URL-адресов данных

fast_field_for(*field_obj*, *book_id*, *default_value=None*)

То же, что field_for, только исключает дополнительный поиск для получения объекта поля

field_for(*name*, *book_id*, *default_value=None*)

Вернуть значение поля name для книги, идентифицированной с помощью book_id. Если такой книги не существует, или она не имеет определенного значения для поля `name` или такое поле не существует, то возвращается `default_value`.

`default_value` не используется для заголовка, title_sort, авторов, author_sort и series_index. Это потому, что они всегда имеют значения в базе данных. `default_value` используется для всех настраиваемых столбцов.

Возвращаемое значение для полей is_multiple всегда является кортежем, даже если значения не найдены (другими словами, default_value игнорируется). Исключение составляют идентификаторы, для которых возвращаемое значение всегда dict. Возвращаемые кортежи всегда находятся в ссылочном порядке (в порядке их создания).

field_ids_for(*name*, *book_id*)

Вернуть идентификаторы (в виде кортежа) для значений, которые поле name имеет в книге, идентифицированной с помощью `book_id`. Если значений нет, или такой книги, или такого поля нет, возвращается пустой кортеж.

field_supports_notes(*field=None*) → bool

Вернуть True, если указанное поле поддерживает примечания. Если поле равно None, вернуть неизменяемое множество всех полей, поддерживающих примечания.

find_identical_books(*mi*, *search_restriction=''*, *book_ids=None*)

Находит книги, у которых есть надмножество авторов в mi и одинаковое название (нечеткое совпадение названия). См. Также *data_for_find_identical_books()* (страница 367).

format(*book_id*, *fmt*, *as_file=False*, *as_path=False*, *preserve_filename=False*)

Вернуть формат электронной книги в виде байтовой строки или *None*, если формат не существует или у нас нет разрешения на запись в файл электронной книги.

Параметры

- **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()* (страница 366) instead for performance.
- **as_path** – Копирует файл формата во временный файл и возвращает путь к временному файлу.
- **preserve_filename** – Если True и возвращает путь, имя файла такое же, как и в библиотеке. Обратите внимание, что использование этого означает, что повторные вызовы дают один и тот же временный файл (который каждый раз создается заново)

format_abspath(*book_id*, *fmt*)

Вернуть абсолютный путь к файлу электронной книги формата *format*. Вам почти никогда не следует использовать это, так как это нарушает обещание потоковой безопасности этого API. Вместо этого используйте *copy_format_to()* (страница 366).

В настоящее время используется только в calibredb list, viewer, edit book, compare_format to original format, open with, массовое редактирование метаданных и каталоги (через get_data_as_dict ()).

Я не верю, что кто-либо, кроме средства просмотра, открывающего и редактирующего книгу, выполняет какие-либо операции ввода-вывода записи файлов с результатами этого вызова.

format_hash(*book_id*, *fmt*)

Вернуть хэш указанного формата для указанной книги. Тип хэша зависит от серверной части, но обычно это SHA-256.

format_metadata(*book_id*, *fmt*, *allow_cache=True*, *update_db=False*)

Вернуть путь, размер и время для указанного формата для указанной книги. Вы не должны использовать path, если в этом нет крайней необходимости, поскольку прямой доступ к нему нарушает гарантии потоковой безопасности этого API. Вместо этого используйте метод *copy_format_to()* (страница 366).

Параметры

- **allow_cache** – Если используются кэшированные значения `True`, в противном случае выполняется медленный доступ к файловой системе. Значения кеша могли быть устаревшими, если доступ к файловой системе выполнялся вне этого API.
- **update_db** – Если `True` поле max_size базы данных обновляется для этой книги.

formats(*book_id*, *verify_formats=True*)

Вернуть кортеж всех форматов для указанной книги. Если verify_formats имеет значение True, проверяет, существуют ли файлы на диске.

get_all_items_that_have_notes(*field_name=None*) → set[int] | dict[str, set[int]]

Вернуть все item_ids для элементов, у которых есть примечания в указанном поле или во всех полях, если field_name равно None

get_all_link_maps_for_book(*book_id*)

Возвращает все ссылки для всех полей, на которые ссылается книга, идентифицированная book_id. Если book_id не существует, метод возвращает {}.

Пример. Предположим, что у автора A есть ссылка X, у автора B есть ссылка Y, у тега S есть ссылка F, а у тега T есть ссылка G. Если у книги 1 есть автор A и тег T, этот метод возвращает {„authors“:{„A“:“X“}, „теги“:{„T“, „G“}}. Если автор книги 2 не является ни A, ни B и не имеет тегов, этот метод возвращает {}.

Параметры

book_id – book id в вопросе.

Результат

{field: {field_value, link_value}, … для всех полей с field_value, имеющим непустое значение ссылки для этой книги

get_categories(*sort='name'*, *book_ids=None*, *already_fixed=None*, *first_letter_sort=False*, *uncollapsed_categories=None*)

Используется внутри для реализации браузера тегов

get_custom_book_data(*name*, *book_ids=()*, *default=None*)

Получите данные для имени. По умолчанию возвращает данные для всех book_ids, передайте список идентификаторов книг, если вам нужны только некоторые данные. Возвращает карту значений book_id. Если определенное значение не может быть декодировано, для него используется значение по умолчанию.

get_id_map(*field*)

Вернуть сопоставление номеров идентификаторов со значениями для указанного поля. Поле должно быть полем «многие-один» или «многие-многие», в противном случае возникает ошибка ValueError.

get_ids_for_custom_book_data(*name*)

Вернуть набор идентификаторов книг, в названии которых есть данные.

get_item_id(*field*, *item_name*, *case_sensitive=False*)

Возвращает идентификатор элемента для item_name или None, если не найден. Эта функция работает очень медленно, если при поиске нескольких имен используется либо get_item_ids(), либо get_item_name_map(). Аналогично, поиск с учетом регистра выполняется быстрее, чем без учета регистра.

get_item_ids(*field*, *item_names*, *case_sensitive=False*)

Возвращает значение dict , сопоставляющее item_name с идентификатором элемента , или None

get_item_name(*field*, *item_id*)

Вернуть имя элемента для элемента, указанного в item_id в указанном поле. См. также *get_id_map()* (страница 369).

get_item_name_map(*field*, *normalize_func=None*)

Вернуть отображение значений элементов в идентификаторы

get_link_map(*for_field*)

Возвращает словарь ссылок для предоставленного поля.

Параметры

for_field – имя поиска поля, для которого требуется карта ссылок

Результат

{field_value:link_value, …} для непустых ссылок

get_metadata(*book_id*, *get_cover=False*, *get_user_categories=True*, *cover_as_data=False*)

Возвращает метаданные для книги, идентифицированной book_id как объект *calibre.ebooks.metadata.book.base.Metadata* (страница 216). Обратите внимание, что список форматов не проверяется. Если get_cover - True, обложка возвращается, либо путь к временному файлу как mi.cover, либо если cover_as_data имеет значение True, то как mi.cover_data.

get_next_series_num_for(*series*, *field='series'*, *current_indices=False*)

Возвращает индекс следующей серии для указанной серии с учетом различных предпочтений, управляющих генерацией номера следующей серии.

Параметры

- **field** – Поле типа серии (по умолчанию - встроенный столбец серии)
- **current_indices** – If True, returns a mapping of book_id to current series_index value instead.

get_notes_resource(*resource_hash*) → dict | None

Вернуть словарь, содержащий данные ресурса и имя или None, если ресурс с указанным хэшем не найден

get_proxy_metadata(*book_id*)

Например *get_metadata()* (страница 370), за исключением того, что он возвращает объект ProxyMetadata, который только считывает значения из базы данных по запросу. Это намного быстрее, чем get_metadata, когда необходимо получить доступ только к небольшому количеству полей из возвращенного объекта метаданных.

get_usage_count_by_id(*field*)

Вернуть отображение идентификатора в счетчик использования для всех значений указанного поля, которое должно быть полем «много-один» или «много-много»

has_book(*mi*)

Верните True, если база данных содержит запись с тем же заголовком, что и переданный объект Metadata. При сравнении регистр не учитывается. См. также *data_for_has_book()* (страница 367).

has_format(*book_id*, *fmt*)

Вернуть True, если формат существует на диске

has_id(*book_id*)

Вернёт True, если указанный book_id существует в БД

import_note(*field*, *item_id*, *path_to_html_file*, *path_is_data=False*)

Импортировать ранее экспортированное примечание или произвольный HTML-файл в качестве примечания для указанного элемента

init()

Инициализировать этот кеш данными из серверной части.

items_with_notes_in_book(*book_id: int*) → dict[str, dict[int, str]]

Вернуть словарь поля для элементов, у которых есть примечания для этого поля для указанной книги

link_for(*field*, *item_id*)

Вернуть ссылку, если она есть, для указанного элемента или None, если ссылка не найдена

list_extra_files(*book_id*, *use_cache=False*, *pattern=''*) → tuple[ExtraFile, ...]

Получить информацию о дополнительных файлах в каталоге книги.

Параметры

- **book_id** – идентификатор книги в базе данных для книги
- **pattern** – паттерн имен файлов для поиска. Пустой паттерн соответствует всем дополнительным файлам. Паттерны должны использовать / в качестве разделителя. Используйте константу DATA_FILE_PATTERN для сопоставления файлов внутри каталога данных.

Результат

Кортеж всех дополнительных файлов, соответствующих указанному шаблону. Каждый элемент кортежа — это ExtraFile(relpath, file_path, stat_result). Где relpath — это относительный путь файла к каталогу книги с использованием / в качестве разделителя. stat_result — это результат вызова os.stat() для файла.

merge_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

Объединяет указанные аннотации с существующими аннотациями для book_id, fm, user_type и user.

merge_extra_files(*dest_id*, *src_ids*, *replace=False*)

Объединить дополнительные файлы из src_ids в dest_id. Конфликтующие файлы автоматически переименовываются, если только replace=True не заменяет их.

move_book_from_trash(*book_id*)

Восстановить книгу из каталога корзины

move_format_from_trash(*book_id*, *fmt*)

Восстановить формат из каталога корзины

multisort(*fields*, *ids_to_sort=None*, *virtual_fields=None*)

Вернуть список отсортированных идентификаторов книг. Если ids_to_sort - None, возвращаются идентификаторы всех книг.

поля должны быть списком из двух кортежей формы (field_name, ascending = True или False). Наиболее значимое поле - первое в кортеже из двух элементов.

notes_data_for(*field*, *item_id*) → str

Вернуть все данные примечаний в виде словаря или None, если примечание не существует

notes_for(*field*, *item_id*) → str

Вернуть документ с примечаниями или пустую строку, если не найдено

notes_resources_used_by(*field*, *item_id*)

Вернуть набор хэшей ресурсов всех ресурсов, используемых примечанием для указанного элемента

pref(*name*, *default=None*, *namespace=None*)

Вернуть значение для указанного предпочтения или значение, указанное как `по умолчанию`, если предпочтение не установлено.

read_backup(*book_id*)

Вернуть резервную копию метаданных OPF для книги в виде байтовой строки или None, если такой резервной копии не существует

remove_books(*book_ids*, *permanent=False*)

Удалить записи книг, указанных в book_ids, из базы данных и удалить их файлы. Если для параметра `permanent` установлено значение False, файлы помещаются в корзину для библиотеки.

remove_extra_files(*book_id: int*, *relpaths: Iterable[str]*, *permanent=False*) → dict[str, Exception | None]

Удалить указанные дополнительные файлы либо в корзину, либо навсегда.

remove_formats(*formats_map*, *db_only=False*)

Remove the specified formats from the specified books.

Параметры

- **formats_map** – Сопоставить book_id со списком форматов, которые нужно удалить из книги.
- **db_only** – Если True, удалить только запись для формата из базы данных, не удалять файл формата из файловой системы.

Результат

Сопоставление идентификатора книги с набором форматов, фактически удаленных из файловой системы для этой книги

remove_items(*field*, *item_ids*, *restrict_to_book_ids=None*)

Удалить все элементы в указанном поле с указанными идентификаторами. Возвращает набор идентификаторов затронутых книг. `restrict_to_book_ids` - необязательный набор идентификаторов книг. Если указано, элементы будут удалены только из этих книг.

rename_extra_files(*book_id*, *map_of_relpath_to_new_relpath*, *replace=False*)

Переименовать дополнительные файлы данных

rename_items(*field*, *item_id_to_new_name_map*, *change_index=True*, *restrict_to_book_ids=None*)

Переименовать элементы из полей «многие-один» или «многие-многие», например тегов или серий.

Параметры

- **change_index** – При переименовании в поле типа серии также изменить значения series_index.
- **restrict_to_book_ids** – Необязательный набор идентификаторов книг, для которых должно быть выполнено переименование, по умолчанию для всех книг.

restore_book(*book_id*, *mi*, *last_modified*, *path*, *formats*, *annotations=()*)

Восстановить запись книги в базе данных для книги, которая уже существует в файловой системе

restore_original_format(*book_id*, *original_fmt*)

Восстановить указанный формат из ранее сохраненного ORIGINAL_FORMAT, если таковой имеется. Вернуть True в случае успеха. ORIGINAL_FORMAT удаляется после успешного восстановления.

property safe_read_lock

Безопасная блокировка чтения - это блокировка, которая ничего не делает, если поток уже имеет блокировку записи, в противном случае он получает блокировку чтения. Это необходимо для предотвращения ошибок DowngradeLockErrors, которые могут возникнуть при обновлении кеша поиска при наличии составных столбцов. Обновление кеша поиска содержит эксклюзивную блокировку, но поиск в составном столбце включает чтение значений полей через ProxyMetadata, который пытается получить общую блокировку. Могут быть и другие сценарии, которые также вызывают это.

Это свойство возвращает новый объект блокировки при каждом доступе. Этот объект блокировки не является рекурсивным (для повышения производительности) и должен использоваться только в операторе with как `with cache.safe_read_lock:`, иначе произойдут плохие вещи.

save_original_format(*book_id*, *fmt*)

Сохранить копию указанного формата как ORIGINAL_FORMAT, перезаписав любой существующий ORIGINAL_FORMAT.

search(*query*, *restriction=''*, *virtual_fields=None*, *book_ids=None*)

Искать в базе данных указанный запрос, возвращая набор совпадающих идентификаторов книг.

Параметры

- **restriction** – Ограничение, добавляемое (AND) к указанному запросу. Обратите внимание, что ограничения кэшируются, поэтому поиск «a» AND «b» будет медленнее, чем «a» с ограничением «b».
- **virtual_fields** – Используется внутри (виртуальные поля, такие как on_device, для поиска).
- **book_ids** – Если не None, то набор идентификаторов книг, по которым будет выполняться поиск книг вместо поиска по всем книгам.

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*)

Возвращает кортеж аннотаций, соответствующий указанному полнотекстовому запросу.

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*)

Искать текст примечаний с использованием индекса FTS. Если запрос пустой, вернуть все примечания.

set_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

Установить все аннотации для указанных book_id, fmt, user_type и user.

set_conversion_options(*options*, *fmt='PIPE'*)

параметры должны быть картой в форме {book_id: conversion_options}

set_cover(*book_id_data_map*)

Установить обложку для этой книги. Данные могут быть QImage, QPixmap, файловым объектом или байтовой строкой. Также может быть None - любая существующая обложка будет удалена

set_field(*name*, *book_id_to_val_map*, *allow_case_change=True*, *do_path_update=True*)

Установить значения поля, указанного в `name`. Возвращает набор всех идентификаторов книг, на которые повлияло изменение.

Параметры

- **book_id_to_val_map** – Сопоставление book_ids значениям, которые следует применить.
- **allow_case_change** – Если True, регистр полей много-один или много-много будет изменен. Например, если у книги есть тег `tag1`, а вы установили тег для другой книги как `Tag1`, тогда обе книги будут иметь тег `Tag1`, если allow_case_change - True, иначе они оба будут иметь тег `tag1`.
- **do_path_update** – Используется внутри, вы никогда не должны его менять.

set_link_map(*field*, *value_to_link_map*, *only_set_if_no_existing_link=False*)

Устанавливает ссылки для значений элемента в поле. Примечание: этот метод не изменяет значения не в value_to_link_map.

Параметры

- **field** – имя поиска
- **value_to_link_map** – dict(field_value:ссылка, …). Обратите внимание, что это значения, а не идентификаторы полей.

Результат

книги изменены путем установки ссылки

set_metadata(*book_id*, *mi*, *ignore_errors=False*, *force_changes=False*, *set_title=True*, *set_authors=True*, *allow_case_change=False*)

Установить метаданные для книги *id* из объекта *Metadata mi*

Установка force_changes = True заставит set_metadata обновлять поля, даже если mi содержит пустые значения. В этом случае *None* отличается от *empty*. Если mi.XXX равно „None“, XXX не заменяется, в противном случае - нет. Теги, идентификаторы и атрибуты обложки - это особые случаи. Теги и идентификаторы не могут быть установлены на „None“, поэтому они всегда будут заменены, если force_changes истинно. Вы должны убедиться, что mi содержит те значения, которые вы хотите, чтобы книга имела. Обложки всегда меняются, если предоставляется новая обложка, но никогда не удаляются. Также обратите внимание, что force_changes не влияет на установку заголовка или авторов.

set_notes_for(*field*, *item_id*, *doc: str*, *searchable_text: str = ''*, *resource_hashes=()*, *remove_unused_resources=False*) → int

Установить документ с примечаниями. Если искомый текст отличается от документа, указать его как searchable_text. Если документ ссылается на ресурсы, их хэши должны присутствовать в resource_hashes. Установить remove_unused_resources в True, чтобы очистить неиспользуемые ресурсы, обратите внимание, что обновление примечания автоматически очищает ресурсы, относящиеся к этому примечанию.

set_pref(*name*, *val*, *namespace=None*)

Установите указанное предпочтение на указанное значение. См. Также: meth:*pref*.

split_if_is_multiple_composite(*f*, *val*)

Если f является составным ключом поиска столбца, а столбец is_multiple, то разделите v на уникальные непустые значения. Сравнение чувствительно к регистру. Порядок не сохраняется. Возвращает list() для совместимости с получателями полей метаданных прокси, например тегами.

tags_older_than(*tag*, *delta=None*, *must_have_tag=None*, *must_have_authors=None*)

Возвратить идентификаторы всех книг с тегом `tag`, которые старше указанного времени. сравнение тегов производится без учета регистра.

Параметры

- **delta** – Объект timedelta или None. Если None, то возвращаются все идентификаторы с тегом.
- **must_have_tag** – Если нет, то список совпадений будет ограничен книгами с этим тегом.
- **must_have_authors** – Список авторов. Если нет, то список совпадений будет ограничен книгами, имеющими этих авторов (без учета регистра).

unretire_note_for(*field*, *item_id*) → int

Восстановить ранее удаленное примечание для указанного элемента. Примечания удаляются, когда элемент удаляется из базы данных.

update_annotations(*annot_id_map*)

Обновить аннотации.

user_categories_for_books(*book_ids*, *proxy_metadata_map=None*)

Вернуть категории пользователей для указанных книг. proxy_metadata_map не является обязательным и полезен для повышения производительности в контекстах, где уже существует объект ProxyMetadata для книг. Это должно быть отображение book_ids на соответствующие им объекты ProxyMetadata.

14.9.2 Документация API инструмента редактирования электронных книг

Инструменты редактирования электронных книг состоят из объекта *calibre.ebooks.oeb.polish.container.Container* (страница 375), который представляет книгу как набор файлов ресурсов HTML +, и различных инструментов, которые можно использовать для выполнения операций с контейнером. Все инструменты представлены в виде функций уровня модуля в различных модулях `calibre.ebooks.oeb.polish.*`. Вы получаете объект-контейнер для книги по следующему пути:

```
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
```

Объект контейнера

class calibre.ebooks.oeb.polish.container.**Container**(*rootpath=None*, *opfpath=None*, *log=<calibre.utils.logging.Log object>*, *clone_data=None*)

Контейнер представляет собой открытую электронную книгу в виде папки, полной файлов и файла OPF. Есть два важных понятия:

- Корневая папка. Это основа электронной книги. Все файлы электронных книг находятся внутри этой папки или в ее подпапках.
- Имена: это пути к файлам книг относительно корневой папки. Они всегда содержат разделители POSIX и не заключаются в кавычки. Их можно рассматривать как канонические идентификаторы файлов в книге. Большинство методов объекта-контейнера работают с именами. Имена всегда в нормальной форме NFC Unicode.
- Клоны: объект-контейнер поддерживает эффективное клонирование на диске, которое используется для реализации контрольных точек в редакторе электронных книг. Чтобы это работало, вы никогда не должны обращаться к файлам в файловой системе напрямую. Вместо этого используйте *raw_data()* (страница 378) или *open()* (страница 378) для чтения/записи файлов компонентов в книге.

При преобразовании между hrefs и именами используйте методы, предоставляемые этим классом, они предполагают, что все hrefs заключены в кавычки.

abspath_to_name(*fullpath*, *root=None*)

Преобразование абсолютного пути в каноническое имя относительно `root`

Параметры

root – Базовая папка. По умолчанию используется корень для этого объекта-контейнера.

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=''*)

Добавить в манифест запись для файла с указанным именем. Возвращает идентификатор манифеста.

add_properties(*name*, **properties*)

Добавить указанные свойства в элемент манифеста, идентифицированный по имени.

apply_unique_properties(*name*, **properties*)

Убедитесь, что указанные свойства установлены только для элемента манифеста, указанного по имени. Вы можете передать None в качестве имени, чтобы удалить свойство из всех элементов.

book_type = 'oeb'

Тип книги epub для файлов EPUB и azw3 для файлов AZW3

commit(*outpath=None*, *keep_parsed=False*)

Зафиксировать все загрязненные проанализированные объекты в файловой системе и записать файл электронной книги по пути outpath.

Параметры

- **output** – Путь для записи сохраненного файла электронной книги. Если None, используется путь к исходному файлу книги.
- **keep_parsed** – Если True, проанализированные представления зафиксированных элементов хранятся в кеше.

commit_item(*name*, *keep_parsed=False*)

Зафиксировать проанализированный объект на диске (он сериализуется и записывается в базовый файл). Если `keep_parsed` - True, проанализированное представление сохраняется в кеше. См. также :*parsed()* (страница 378)

dirty(*name*)

Отметить проанализированный объект, соответствующий имени, как грязный. См. Также :*parsed()* (страница 378).

exists(*name*)

Истинно, только если существует файл/папка, соответствующие каноническому имени. Обратите внимание, что эта функция страдает от ограничений базовой файловой системы ОС, в частности из-за (не) чувствительности. Таким образом, в файловой системе, нечувствительной к регистру, это вернет True, даже если регистр имени отличается от регистра файла базовой файловой системы. См. Также *has_name()* (страница 376)

filesize(*name*)

Вернуть размер в байтах файла, представленного указанным каноническим именем. Автоматически обрабатывает загрязненные проанализированные объекты. См. также :*parsed()* (страница 378)

generate_item(*name*, *id_prefix=None*, *media_type=None*, *unique_href=True*)

Добавить в манифест элемент с href, производным от заданного имени. Автоматически обеспечивает уникальность href и id. Возвращает сгенерированный элемент.

get_file_path_for_processing(*name*, *allow_modification=True*)

Подобно open(), за исключением того, что он возвращает путь к файлу вместо объекта открытого файла.

property guide_type_map

Сопоставление guide типа с каноническим именем

has_name(*name*)

Вернуть True, если существует файл с тем же каноническим именем, что и указанное. В отличие от *exists()* (страница 376) этот метод всегда чувствителен к регистру.

href_to_name(*href*, *base=None*)

Преобразование href (относительно base) в имя. base должно быть именем или None, в этом случае используется self.root.

insert_into_xml(*parent*, *item*, *index=None*)

Вставить элемент в родительский (или добавить, если индекс равен None), исправив отступ. Работает только с самозакрывающимися элементами.

is_dir = False

Если этот контейнер представляет собой распакованную книгу (каталог)

iterlinks(*name*, *get_line_numbers=True*)

Перебрать все ссылки по имени. Если get_line_numbers имеет значение True, выдает результаты формы (ссылка, номер_строки, смещение). Где line_number - это номер line_number, в котором находится ссылка, а смещение - это количество символов от начала строки. Обратите внимание, что смещение может охватывать несколько строк, если не ноль.

make_name_unique(*name*)

Убедится, что *name* уже не существует в этой книге. Если это так, вернуть измененную версию, которая не существует.

manifest_has_name(*name*)

Вернуть True, если в манифесте есть запись, соответствующая имени

property manifest_id_map

Сопоставление идентификатора манифеста с каноническими именами

manifest_items_of_type(*predicate*)

Имена всех элементов манифеста, тип носителя которых соответствует предикату. *predicate* может быть набором, списком, строкой или функцией, принимающей один аргумент, которая будет вызываться с медиа-типом.

manifest_items_with_property(*property_name*)

Все элементы манифеста, имеющие указанное свойство

property manifest_type_map

Сопоставление манифестного медиа-типа со списком канонических имен этого медиа-типа

property mi

Метаданные этой книги как объект метаданных. Обратите внимание, что этот объект создается «на лету» каждый раз, когда это свойство запрашивается, поэтому используйте его с осторожностью.

name_to_abspath(*name*)

Преобразование канонического имени в путь, полностью зависящий от ОС

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

Набор имен, которые нельзя переименовывать. Зависит от формата файла электронной книги.

property names_that_must_not_be_removed

Набор имен, которые нельзя удалять из контейнера. Зависит от формата файла электронной книги.

property names_that_need_not_be_manifested

Набор имен, которые могут отсутствовать в манифесте. Зависит от формата файла электронной книги.

open(*name*, *mode='rb'*)

Открыть файл, указанный по имени, для прямого чтения/записи. Обратите внимание, что это зафиксирует файл, если он загрязнен, и удалит его из кеша синтаксического анализа. Вы должны закончить работу с этим файлом, прежде чем снова обращаться к его проанализированной версии, иначе произойдут неприятности.

property opf

Разобранный файл OPF

opf_get_or_create(*name*)

Удобный метод, позволяющий либо вернуть первый элемент XML с указанным именем, либо создать его в элементе opf:package, а затем вернуть его, если он ещё не существует.

property opf_version

Версия, установленная на <package> элементе OPF

property opf_version_parsed

Версия, заданная в <package> элементе OPF как кортеж целых чисел

opf_xpath(*expr*)

Удобный метод для оценки выражения XPath в файле OPF имеет предопределенные префиксы пространства имен opf: и dc:.

parsed(*name*)

Вернуть проанализированное представление файла, указанного по имени. Для файлов HTML и XML возвращается дерево lxml. Для файлов CSS возвращается таблица стилей css_parser. Обратите внимание, что проанализированные объекты кэшируются для повышения производительности. Если вы вносите какие-либо изменения в анализируемый объект, вы должны вызвать *dirty()* (страница 376), чтобы контейнер знал, что нужно обновить кеш. См. Также *replace()* (страница 379).

raw_data(*name*, *decode=True*, *normalize_to_nfc=True*)

Вернуть необработанные данные, соответствующие файлу, указанному по имени

Параметры

- **decode** – Если True и файл имеет текстовый MIME-тип, декодировать его и вернуть объект Unicode вместо необработанных байтов.
- **normalize_to_nfc** – Если True, возвращаемый объект unicode нормализуется до нормальной формы NFC, как это требуется для форматов файлов EPUB и AZW3.

relpath(*path*, *base=None*)

Преобразование абсолютного пути (с разделителями os) в путь относительно базы (по умолчанию self.root). Относительный путь - *не* имя. Для этого используйте *abspath_to_name()* (страница 375).

remove_from_spine(*spine_items*, *remove_if_no_longer_in_spine=True*)

Удалить указанные пункты (по каноническому имени) из корешка. Если remove_if_no_longer_in_spine имеет значение True, элементы также удаляются из книги, а не только из корешка.

remove_from_xml(*item*)

Удалить элемент из родительского, исправляя отступ (работает только с самозакрывающимися элементами)

remove_item(*name*, *remove_from_guide=True*)

Удалить из этого контейнера элемент, указанный по имени. Это удалит все ссылки на элемент в манифесте, руководстве и корешке OPF, а также из любых внутренних кешей.

rename(*current_name*, *new_name*)

Переименовывает файл с current_name в new_name. Он автоматически переустанавливает все ссылки внутри файла, если папка, в которой находится файл, изменяется. Однако обратите внимание, что ссылки не обновляются в других файлах, которые могут ссылаться на этот файл. Это для производительности, такие обновления нужно делать один раз, массово.

replace(*name*, *obj*)

Заменить проанализированный объект, соответствующий имени, на obj, который должен быть аналогичным объектом, то есть деревом lxml для HTML/XML или таблицей стилей css_parser для файла CSS.

replace_links(*name*, *replace_func*)

Заменить все ссылки в имени с помощью replace_func, которая должна быть вызываемой, которая принимает URL-адрес и возвращает замененный URL-адрес. Он также должен иметь атрибут replace, который имеет значение True, если выполняется какая-либо фактическая замена. Удобные способы создания таких вызываемых объектов - это использование классов `LinkReplacer` и `LinkRebaser`.

serialize_item(*name*)

Преобразовать проанализированный объект (идентифицируемый каноническим именем) в байтовую строку. Смотрите *parsed()* (страница 378).

set_spine(*spine_items*)

Задать для spine значение spine_items, где spine_items - это итерабельность формы (имя, линейная). Вызовет ошибку, если одно из имен отсутствует в манифесте.

property spine_items

Итератор, определяющий путь для каждого элемента корешка книги. См. также :*spine_iter* (страница 379) и *spine_items* (страница 379).

property spine_iter

Итератор, который выдает элемент с именем is_linear для каждого элемента в корешке книги. item - это элемент lxml, name - каноническое имя файла, a is_linear - True, если элемент является линейным. См. также :*spine_names* (страница 379) и *spine_items* (страница 379).

property spine_names

Итератор, дающий имя и is_linear для каждого элемента в корешке книги. См. также :*spine_iter* (страница 379) и *spine_items* (страница 379).

Управление файлами компонентов в контейнере

calibre.ebooks.oeb.polish.replace.**replace_links**(*container*, *link_map*, *frag_map=<function <lambda>>*, *replace_in_opf=False*)

Заменить ссылки на файлы в контейнере. Будет перебирать все файлы в контейнере и изменять в них указанные ссылки.

Параметры

- **link_map** – Сопоставление старого канонического имени с новым каноническим именем. Например: `{'images/old.png': 'images/new.png'}`
- **frag_map** – Вызываемый объект, который принимает два аргумента `(имя, привязка)` и возвращает новую привязку. Это полезно, если вам нужно изменить привязки в файлах HTML. По умолчанию ничего не делает.

- **replace_in_opf** – Если false, ссылки не заменяются в файле OPF.

calibre.ebooks.oeb.polish.replace.**rename_files**(*container*, *file_map*)

Переименовывать файлы в контейнере, автоматически обновляя все ссылки на них.

Параметры

file_map – Сопоставление старого канонического имени с новым каноническим именем, например: `{'text/chapter1.html': 'chapter1.html'}`.

calibre.ebooks.oeb.polish.replace.**get_recommended_folders**(*container*, *names*)

Вернуть папки, рекомендованные для данных имен файлов. Рекомендация основана на том, где в контейнере находится большинство файлов одного типа. Если файлы определенного типа отсутствуют, рекомендуемая папка считается папкой, содержащей файл OPF.

Хорошая печать и автоматическое исправление ошибок синтаксического анализа

calibre.ebooks.oeb.polish.pretty.**fix_html**(*container*, *raw*)

Исправить любые ошибки синтаксического анализа в HTML, представленном в виде строки в необработанном виде. Исправление производится с помощью алгоритма парсинга HTML5.

calibre.ebooks.oeb.polish.pretty.**fix_all_html**(*container*)

Исправить все ошибки синтаксического анализа во всех файлах HTML в контейнере. Исправление производится с помощью алгоритма парсинга HTML5.

calibre.ebooks.oeb.polish.pretty.**pretty_html**(*container*, *name*, *raw*)

Хорошая печать HTML в представленного виде строки в необработанном виде

calibre.ebooks.oeb.polish.pretty.**pretty_css**(*container*, *name*, *raw*)

Хорошая печать CSS, представленной в виде строки в необработанном виде

calibre.ebooks.oeb.polish.pretty.**pretty_xml**(*container*, *name*, *raw*)

Хорошая печать XML, представленного в виде строки в необработанном виде. Если name - это имя OPF, выполняется дополнительная обработка, специфичная для OPF.

calibre.ebooks.oeb.polish.pretty.**pretty_all**(*container*)

Хорошая печать всех файлов HTML/CSS/XML в контейнере

Управление обложками книг

calibre.ebooks.oeb.polish.jacket.**remove_jacket**(*container*)

Удалить существующую обложку, если таковая имеется. Возвращает False, если существующая обложка не найдена.

calibre.ebooks.oeb.polish.jacket.**add_or_replace_jacket**(*container*)

Либо создать новую обложку из метаданных книги, либо заменить существующую обложку. Возвращает True, если существующая обложка была заменена.

Разделение и слияние файлов

calibre.ebooks.oeb.polish.split.**split**(*container*, *name*, *loc_or_xpath*, *before=True*, *totals=None*)

Разделить файл, указанный по имени, в позиции, указанной параметром loc_or_xpath. При разделении автоматически переносятся все ссылки и ссылки на затронутые файлы.

Параметры

- **loc_or_xpath** – Должно быть выражением XPath, например //h:div[@id="split_here"]. Также может быть *loc*, который используется внутри для реализации разделения на панели предварительного просмотра.

- **before** – Если True, разделение происходит до идентифицированного элемента, иначе после него.
- **totals** – Используется внутри

calibre.ebooks.oeb.polish.split.**multisplit**(*container*, *name*, *xpath*, *before=True*)

Разделить указанный файл в нескольких местах (все теги, соответствующие указанному выражению XPath). См. также: *split()* (страница 380). При разделении автоматически переносятся все ссылки и ссылки на затронутые файлы.

Параметры

before – Если True, разбиение происходит до идентифицированного элемента, иначе после него.

calibre.ebooks.oeb.polish.split.**merge**(*container*, *category*, *names*, *master*)

Объединить указанные файлы в один файл, автоматически перенеся все ссылки и ссылки на затронутые файлы. Все файлы должны быть либо HTML, либо CSS.

Параметры

- **category** – Должен быть либо `text` для файлов HTML, либо `styles` для файлов CSS.
- **names** – Список файлов для слияния
- **master** – Какой из объединенных файлов является *главным* файлом, то есть файлом, который останется после объединения.

Управление обложками

calibre.ebooks.oeb.polish.cover.**set_cover**(*container*, *cover_path*, *report=None*, *options=None*)

Установить обложку книги на изображение, на которое указывает cover_path

Параметры

- **cover_path** – Либо абсолютный путь к файлу изображения, либо каноническое имя изображения в книге. При использовании изображения в книге вы также должны установить параметры, см. ниже.
- **report** – Необязательный вызываемый объект, который принимает единственный аргумент. Он будет вызываться с информацией об обрабатываемых задачах
- **options** – None или словарь, который контролирует установку обложки. В словаре могут быть записи: **keep_aspect**: True или False (сохранить соотношение сторон обложек в EPUB) **no_svg**: True или False (использовать обертку обложки SVG на титульной странице EPUB) **existing**: True или False (cover_path относится к существующему изображению в книге)

calibre.ebooks.oeb.polish.cover.**mark_as_cover**(*container*, *name*)

Отметить указанное изображение как изображение обложки.

calibre.ebooks.oeb.polish.cover.**mark_as_titlepage**(*container*, *name*, *move_to_start=True*)

Отметить указанный HTML-файл как титульную страницу EPUB.

Параметры

move_to_start – Если True, HTML-файл перемещается в начало корешка.

Работа с CSS

calibre.ebooks.oeb.polish.fonts.**change_font**(*container*, *old_name*, *new_name=None*)

Изменить семейство шрифтов с old_name на new_name. Изменяет все вхождения семейства шрифтов в таблицах стилей, тегах стилей и атрибутах стиля. Если old_name относится к встроенному шрифту, он удаляется. Вы можете установить для new_name значение None, чтобы удалить семейство шрифтов вместо его изменения.

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*)

Удалить из книги все неиспользуемые правила CSS. Неиспользуемое правило CSS - это правило, не соответствующее фактическому содержанию.

Параметры

- **report** – Необязательный вызываемый объект, который принимает единственный аргумент. Вызывается с информацией о выполняемых операциях.
- **remove_unused_classes** – Если True, атрибуты класса в HTML, которые не соответствуют никаким правилам CSS, также удаляются.
- **merge_rules** – Если True, правила с идентичными селекторами объединяются.
- **merge_rules_with_identical_properties** – Если True, правила с одинаковыми свойствами объединяются.
- **remove_unreferenced_sheets** – Если True, таблицы стилей, на которые не ссылается какой-либо контент, удаляются.

calibre.ebooks.oeb.polish.css.**filter_css**(*container*, *properties*, *names=()*)

Удалить указанные свойства CSS из всех правил CSS в книге.

Параметры

- **properties** – Набор свойств для удаления. Например: `{'font-family', 'color'}`.
- **names** – Файлы, из которых нужно удалить свойства. По умолчанию используются все файлы HTML и CSS в книге.

Работа с оглавлением

calibre.ebooks.oeb.polish.toc.**from_xpaths**(*container*, *xpaths*, *prefer_title=False*)

Создать оглавление из списка выражений XPath. Каждое выражение в списке соответствует уровню создания ToC. Например: `['//h:h1','//h:h2','//h:h3']` сгенерирует трехуровневую таблицу содержания из `<h1>`, `<h2>` и `<h3>` тегов.

calibre.ebooks.oeb.polish.toc.**from_links**(*container*)

Создать оглавление из ссылок в книге.

calibre.ebooks.oeb.polish.toc.**from_files**(*container*)

Создать оглавление из файлов в книге.

calibre.ebooks.oeb.polish.toc.**create_inline_toc**(*container*, *title=None*)

Создать встроенное (HTML) оглавление из существующего оглавления NCX.

Параметры

title – Заголовок этого оглавления.

Средство редактирования книги

class calibre.gui2.tweak_book.plugin.**Tool**

Базовые классы: `object`

Базовый класс для отдельных инструментов в плагине Edit Book. Среди полезных членов:

- `self.plugin`: ссылка на объект *calibre.customize.Plugin* (страница 264), которому принадлежит этот инструмент.
- self. *boss* (страница 383)
- self. *gui* (страница 383)

Методы, которые необходимо переопределить в подклассах:

- *create_action()* (страница 384)
- *register_shortcut()* (страница 383)

name = None

Установить уникальное имя, которое будет использоваться как ключ

allowed_in_toolbar = True

Если True, пользователь может разместить этот инструмент на панели инструментов плагинов.

allowed_in_menu = True

Если True, пользователь может выбрать размещение этого инструмента в меню плагинов.

toolbar_button_popup_mode = 'delayed'

Всплывающий режим для меню (если есть) кнопки панели инструментов. Возможные значения: „delayed“, „instant“, „button“.

property boss

Объект *calibre.gui2.tweak_book.boss.Boss* (страница 384). Используется для управления пользовательским интерфейсом.

property gui

Главное окно пользовательского интерфейса

property current_container

Вернуть текущий объект *calibre.ebooks.oeb.polish.container.Container* (страница 375), представляющий редактируемую книгу.

register_shortcut(*qaction*, *unique_name*, *default_keys=()*, *short_text=None*, *description=None*, ***extra_data*)

Зарегистрируйте сочетание клавиш, которое будет запускать указанное действие `qaction`. Это сочетание клавиш станет автоматически настраиваемым пользователем в разделе «Сочетания клавиш» настроек редактора.

Параметры

- **qaction** – Объект QAction, он будет срабатывать, когда пользователь нажимает настроенную комбинацию клавиш.
- **unique_name** – Уникальное имя для этого ярлыка/действия. Он будет использоваться для внутренних целей, он не должен использоваться другими действиями в этом плагине.
- **default_keys** – Список сочетаний клавиш по умолчанию. Если не указано иное, ярлыки по умолчанию не будут установлены. Если указанные здесь ярлыки конфликтуют со встроенными ярлыками или ярлыками из пользовательской конфигурации/других

плагинов, они будут проигнорированы. В этом случае пользователям придется настраивать ярлыки вручную в настройках. Например: `default_keys=('Ctrl+J', 'F9')`.

- **short_text** – Необязательное краткое описание этого действия. Если не указано иное, будет использован текст из QAction.
- **description** – Необязательное более подробное описание этого действия, оно будет использовано в записи настроек для этого ярлыка.

create_action(*for_toolbar=True*)

Создать QAction, который будет добавлен либо на панель инструментов плагинов, либо в меню плагинов в зависимости от `for_toolbar`. Например:

```
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
```

См. также

Метод *register_shortcut()* (страница 383).

Управление пользовательским интерфейсом редактора

Пользовательский интерфейс редактора электронных книг управляется одним глобальным объектом *Boss*. У этого есть много полезных методов, которые можно использовать в коде плагина для выполнения общих задач.

class calibre.gui2.tweak_book.boss.**Boss**(*parent*, *notify=None*)

add_savepoint(*msg*)

Создать контрольную точку восстановления с именем, указываемым как `msg`

apply_container_update_to_gui(*mark_as_modified=True*)

Обновить все компоненты пользовательского интерфейса, чтобы отразить последние данные в текущем контейнере книги.

Параметры

mark_as_modified – Если True, книга будет помечена как изменённая, поэтому пользователю будет предложено сохранить её при выходе.

close_editor(*name*)

Закрыть редактор, который редактирует файл, указанный в `name`.

commit_all_editors_to_container()

Зафиксировать любые изменения, внесённые пользователем в файлы, открытые в редакторах, в контейнер. Вы должны вызвать этот метод перед выполнением каких-либо действий с текущим контейнером.

property currently_editing

Вернуть имя редактируемого файла или None, если файл не редактируется.

edit_file(*name*, *syntax=None*, *use_template=None*)

Открыть файл, указанный по имени, в редакторе

Параметры

- **syntax** – Медиа-тип файла, например, `text/html`. Если не указано, это определяется по расширению файла.
- **use_template** – Шаблон для инициализации открытого редактора с помощью

open_book(*path=None*, *edit_file=None*, *clear_notify_data=True*, *open_folder=False*, *search_text=None*)

Открыть электронную книгу по пути для редактирования. Будет отображена ошибка, если формат электронной книги не поддерживается или в текущей книге есть несохраненные изменения.

Параметры

edit_file – Имя файла во вновь открытой книге, чтобы начать редактирование. Также может быть список имен.

rewind_savepoint()

Отменить предыдущее создание контрольной точки восстановления, полезно, если вы создаете контрольную точку, а затем прерываете операцию без изменений

save_book()

Сохранить книгу. Сохранение происходит в фоновом режиме

set_modified()

Отметить книгу как изменённую

show_current_diff(*allow_revert=True*, *to_container=None*)

Показать изменения в книге по сравнению с последней контрольной точкой

Параметры

- **allow_revert** – Если True, в диалоговом окне сравнения будет кнопка, позволяющая пользователю отменить все изменения.
- **to_container** – Объект контейнера, с которым сравнивается текущий контейнер. Если None, используется контейнер с ранее установленной контрольной точкой.

show_editor(*name*)

Показать редактор, редактирующий файл, указанный в `name`

sync_preview_to_editor()

Синхронизировать положение панели предварительного просмотра с текущей позицией курсора в текущем редакторе

ГЛАВА 15

Технические средства защиты авторских прав (DRM — digital rights management)

Управление цифровыми правами (DRM) - это общий термин для технологий контроля доступа, который может использоваться производителями оборудования, издателями, владельцами авторских прав и частными лицами, чтобы попытаться наложить ограничения на использование цифрового контента и устройств. Это также, иногда, пренебрежительно описывается как Управление цифровыми ограничениями. Этот термин используется для описания любой технологии, которая запрещает использование (законного или иного) цифрового контента, который не был желателен или предвиден поставщиком контента. Термин обычно не относится к другим формам защиты от копирования, которые можно обойти без изменения файла или устройства, например, серийные номера или файлы ключей. Это может также относиться к ограничениям, связанным с конкретными экземплярами цифровых произведений или устройств. Технологии DRM пытаются контролировать использование цифровых медиа, предотвращая доступ, копирование или преобразование в другие форматы конечными пользователями. Смотрите *wikipedia <https://en.wikipedia.org/wiki/Digital_rights_management> _*.

15.1 Что несёт DRM лично для меня?

Когда вы покупаете электронную книгу с помощью DRM, вы на самом деле не владеете ею, но приобрели разрешение на её использование в соответствии с требованиями продавца. DRM ограничивает то, что вы можете делать с электронными книгами, которые вы «купили». Часто люди, которые покупают книги с DRM, не знают о степени этих ограничений. Эти ограничения не позволяют вам переформатировать электронную книгу по своему вкусу, в том числе вносить стилистические изменения, такие как корректировка размеров шрифта, хотя есть программное обеспечение, которое позволяет вам делать такие вещи для не DRM книг. Люди часто удивляются, что электронная книга, которую они купили в определенном формате, не может быть преобразована в другой формат, если электронная книга имеет DRM. Поэтому, если у вас есть Amazon Kindle и вы купили книгу, проданную Barnes and Nobles, вы должны знать, что если эта электронная книга имеет DRM, вы не сможете прочитать её на своем Kindle. Обратите внимание, что я говорю о книге, которую вы покупаете, не украсть или спиратить, а купить.

15.2 Что DRM даёт авторам?

Издатели электронных книг с DRM утверждают, что DRM - это все ради авторов и для защиты их художественной целостности и предотвращения пиратства. Но DRM НЕ предотвращает пиратство. Люди, которые хотят пиратского контента или используют пиратский контент, все ещё делают это и добиваются успеха. В настоящее время Amazon, Adobe, Barnes и Noble управляют тремя основными схемами DRM для электронных книг, и все три схемы DRM были взломаны. Все, что делает DRM - это неудобство для законных пользователей. Можно утверждать, что это на самом деле вредит авторам, поскольку люди, которые купили бы книгу, решили найти пиратскую версию, поскольку они не готовы мириться с DRM. Те, кто будет пиратствовать в отсутствие DRM, делают это и в его присутствии. Повторюсь, ключевой момент заключается в том, что DRM *не предотвращает пиратство*. Таким образом, DRM не только бессмысленен и вреден для покупателей электронных книг, но и по сути бессмысленная трата денег.

15.3 DRM и свобода

Хотя цифровой контент можно использовать для того, чтобы сделать информацию, а также творческие произведения, легко доступными для всех и расширить возможности человечества, это не в интересах издателей, закрепощающих людей для собственной выгоды.

15.4 Почему calibre не поддерживает DRM?

calibre является программным обеспечением с открытым исходным кодом, в то время как DRM по своей природе закрыт. Если бы calibre поддерживал открытие или просмотр файлов DRM, его можно было бы тривиально изменить, чтобы использовать в качестве инструмента для удаления DRM, а это сейчас незаконно. Программное обеспечение с открытым исходным кодом и DRM - это столкновение принципов. В то время как DRM - это всё, что касается управления пользователем, программное обеспечение с открытым исходным кодом - это расширение возможностей пользователя. Они просто не могут сосуществовать.

15.5 Каково мнение calibre о поставщиках информации?

Мы твердо верим, что авторам и другим контент-провайдерам следует компенсировать их усилия, но DRM - это не тот способ, которым можно это сделать. Мы разрабатываем эту базу данных электронных книг без DRM из различных источников, чтобы помочь вам найти альтернативы без DRM и помочь независимым авторам и издателям электронных книг без DRM обнародовать их содержание. Мы надеемся, что вы найдете это полезным, и просим вас не использовать пиратский контент, доступный вам здесь.

15.6 Как я могу помочь бороться с DRM?

Как человек, который читает и покупает электронные книги, вы можете помочь бороться с DRM. Не покупайте электронные книги с DRM. Отдельные издатели публикуют электронные книги и без DRM. Поищите у них на сайте желаемую электронную книгу. Если вам нравятся книги определённых независимых авторов продающих электронные книги без DRM сделайте им пожертвования (если вы можете себе это позволить). Это деньги будут потрачены не зря, поскольку их электронные книги, как правило, дешевле (но могут быть исключения), приобретаемых у издателей с DRM, и, вероятно, будут работать на всех ваших устройствах в будущем, что позволит вам снова сэкономить на покупке электронных книг. Не препятствуйте своим пиратством издателям и авторам электронных книг без DRM. Контент-провайдеры заслуживают компенсации за свои усилия. Не наказывайте их за попытки сделать доступнее электронные книги без DRM. В конечном итоге это вредит и вам. Если вы купили DRM-отягощённые книги у книготорговцев и не знали, что они содержат DRM, то оставляйте комментарий или отзыв на веб-сайте, информируя будущих покупателей о наличии DRM. Многие книготорговцы не считают важным дать чёткое указание своим покупателям на то, содержит ли электронная книга DRM или нет. *Здесь <https://www.defectivebydesign.org/guide/ebooks>* вы найдете руководство по жизни без DRM.

ГЛАВА 16

Глоссарий

RSS

RSS *(Really Simple Syndication)* - формат веб-фида, используемый для публикации часто обновляемого контента, например новостей, сообщений в блогах и т.д. Это формат, который особенно подходит для чтения компьютерами и поэтому он - предпочтительный способ получения контента из Интернета для электронной книги. В Интернете есть много других форматов фидов, и большинство из них понимается в calibre. В частности, имеется хорошая поддержка формата * ATOM *, который обычно используется для блогов.

рецепт

Рецепт - это набор инструкций, в которых рассказывается о том, как преобразовать онлайн-источник новостей, например журнал или блог, в электронную книгу. Рецепт - это по существу код Python[109]. Таким образом, он способен преобразовывать произвольно сложные источники новостей в электронные книги. На самом простом уровне это всего лишь набор переменных, таких как URL-адреса, которые дают достаточную информацию для выхода в Интернет и загрузки новостей.

HTML

HTML *(язык разметки гипертекста)*, поднабор стандартного обобщенного языка разметки (SGML) для электронной публикации, является стандартом, используемым для Всемирной паутины.

CSS

CSS *(Cascading Style Sheets)* — это язык, описывающий, как должен отображаться документ *HTML* (визуальный стиль).

API

API *(Application Programming Interface)* — это интерфейс исходного кода, предоставляемый библиотекой для поддержки запросов на обслуживание, сделанных с помощью компьютерных программ.

LRF

LRF — формат электронных книг в читалках SONY.

URL

URL *(Uniform Resource Locator)*, например: `http://example.com`

regexp

Регулярные выражения предоставляют ёмкие и гибкие возможности по распознаванию интересующих

[109] https://www.python.org

строк текста, например, конкретных символов, слов или наборов символов. Смотрите *the tutorial* (страница 221) для знакомства с регулярными выражениями.

Содержание модулей Python

C

calibre.customize, 264
calibre.customize.conversion, 274
calibre.db.cache, 363
calibre.devices.interface, 277
calibre.ebooks.metadata.book.base, 216
calibre.ebooks.metadata.sources.base, 270
calibre.ebooks.oeb.polish.container, 375
calibre.ebooks.oeb.polish.cover, 381
calibre.ebooks.oeb.polish.css, 382
calibre.ebooks.oeb.polish.jacket, 380
calibre.ebooks.oeb.polish.pretty, 380
calibre.ebooks.oeb.polish.replace, 379
calibre.ebooks.oeb.polish.split, 380
calibre.ebooks.oeb.polish.toc, 382
calibre.gui2.tweak_book.boss, 384
calibre.utils.formatter_functions, 191
calibre.web.feeds.news, 41

Алфавитный указатель

Символы

-1
    calibredb-add опция командной строки, 326
-H
    ebook-polish опция командной строки, 347
-I
    calibredb-add опция командной строки, 325
    fetch-ebook-metadata опция командной строки, 350
-S
    calibredb-add опция командной строки, 326
-T
    calibredb-add опция командной строки, 326
-U
    ebook-polish опция командной строки, 348
--access-log
    calibre-server опция командной строки, 317
--add
    calibredb-add опция командной строки, 326
--add-alt-text-to-img
    ebook-convert опция командной строки, 342
--add-plugin
    calibre-customize опция командной строки, 314
--add-simple-plugin
    calibre-debug опция командной строки, 315
--add-soft-hyphens
    ebook-polish опция командной строки, 347
--ajax-timeout
    calibre-server опция командной строки, 317
--all
    calibredb-backup_metadata опция командной строки, 333
    calibredb-export опция командной строки, 328
--allowed-plugin
    fetch-ebook-metadata опция командной строки, 350
--append
    calibredb-set_custom опция командной строки, 332
--as-extra-data-file
    calibredb-add_format опция командной строки, 327
--as-opf
    calibredb-show_metadata опция командной строки, 327
--ascending
    calibredb-list опция командной строки, 324
--asciiize
    ebook-convert опция командной строки, 338
--attachment
    calibre-smtp опция командной строки, 321
--auth-mode
    calibre-server опция командной строки, 317
--author-sort
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 346
--authors
    calibredb-add опция командной строки, 325
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 346
    fetch-ebook-metadata опция командной строки, 350
--auto-reload
    calibre-server опция командной строки, 317
--automerge
    calibredb-add опция командной строки, 325
--ban-after
    calibre-server опция командной строки, 317
--ban-for
    calibre-server опция командной строки, 317

--base-dir
    web2disk опция командной строки, 352
--base-font-size
    ebook-convert опция командной строки, 338
--book-list-mode
    calibre-server опция командной строки, 318
--book-producer
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 346
--build-plugin
    calibre-customize опция командной строки, 314
--cafile
    calibre-smtp опция командной строки, 322
--catalog-title
    calibredb-catalog опция командной строки, 329
--categories
    calibredb-list_categories опция командной строки, 333
--category
    ebook-meta опция командной строки, 346
--change-justification
    ebook-convert опция командной строки, 338
--chapter
    ebook-convert опция командной строки, 342
--chapter-mark
    ebook-convert опция командной строки, 342
--command
    calibre-debug опция командной строки, 315
--comments
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 346
--compress-images
    ebook-polish опция командной строки, 347
--compress-min-size
    calibre-server опция командной строки, 318
--continue
    ebook-viewer опция командной строки, 349
--cover
    calibredb-add опция командной строки, 325
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 346
    ebook-polish опция командной строки, 347
    fetch-ebook-metadata опция командной строки, 350
--cross-reference-authors
    calibredb-catalog опция командной строки, 329
--csv
    calibredb-check_library опция командной строки, 332
    calibredb-list_categories опция командной строки, 333
--custom-list-template
    calibre-server опция командной строки, 318
--customize-plugin
    calibre-customize опция командной строки, 314
--daemonize
    calibre-server опция командной строки, 318
--date
    ebook-meta опция командной строки, 346
--debug-device-driver
    calibre-debug опция командной строки, 315
--debug-pipeline
    calibredb-catalog опция командной строки, 329
    ebook-convert опция командной строки, 345
--default-programs
    calibre-debug опция командной строки, 315
--delay
    web2disk опция командной строки, 352
--detach
    calibre опция командной строки, 314
    ebook-edit опция командной строки, 345
    ebook-viewer опция командной строки, 349
--details
    calibredb-custom_columns опция командной строки, 331
--dialect
    calibredb-list_categories опция командной строки, 333
--diff
    calibre-debug опция командной строки, 315
--disable-allow-socket-preallocation
    calibre-server опция командной строки, 318
--disable-auth
    calibre-server опция командной строки, 318
--disable-dehyphenate
    ebook-convert опция командной строки, 341
--disable-delete-blank-paragraphs
    ebook-convert опция командной строки, 341
--disable-fallback-to-detected-interface
    calibre-server опция командной строки, 318
--disable-fix-indents
    ebook-convert опция командной строки, 341
--disable-font-rescaling
    ebook-convert опция командной строки, 338
--disable-format-scene-breaks
    ebook-convert опция командной строки, 341

--disable-hyphenation
    lrfviewer опция командной строки, 351
--disable-italicize-common-cases
    ebook-convert опция командной строки, 341
--disable-local-write
    calibre-server опция командной строки, 318
--disable-log-not-found
    calibre-server опция командной строки, 318
--disable-markup-chapter-headings
    ebook-convert опция командной строки, 341
--disable-plugin
    calibre-customize опция командной строки, 314
--disable-remove-fake-margins
    ebook-convert опция командной строки, 342
--disable-renumber-headings
    ebook-convert опция командной строки, 341
--disable-unwrap-lines
    ebook-convert опция командной строки, 341
--disable-use-bonjour
    calibre-server опция командной строки, 318
--disable-use-sendfile
    calibre-server опция командной строки, 319
--disallow-rendered-cover
    ebook-meta опция командной строки, 346
--display
    calibredb-add_custom_column опция командной строки, 331
--displayed-fields
    calibre-server опция командной строки, 318
--do-not-match-on-related-words
    calibredb-fts_search опция командной строки, 335
--dont-asciiize
    calibredb-export опция командной строки, 328
--dont-download-stylesheets
    web2disk опция командной строки, 352
--dont-output-resources
    lrf2lrs опция командной строки, 350
--dont-replace
    calibredb-add_format опция командной строки, 327
--dont-save-cover
    calibredb-export опция командной строки, 328
--dont-save-extra-files
    calibredb-export опция командной строки, 328
--dont-split-on-page-breaks
    ebook-convert опция командной строки, 337
--dont-update-metadata
    calibredb-export опция командной строки, 328
--dont-verify-server-certificate
    calibre-smtp опция командной строки, 322
--dont-write-opf
    calibredb-export опция командной строки, 328
--download-external-resources
    ebook-polish опция командной строки, 347
--duplicate-links-in-toc
    ebook-convert опция командной строки, 343
--duplicates
    calibredb-add опция командной строки, 325
--edit-book
    calibre-debug опция командной строки, 315
--embed-all-fonts
    ebook-convert опция командной строки, 338
--embed-font-family
    ebook-convert опция командной строки, 339
--embed-fonts
    ebook-polish опция командной строки, 348
--empty
    calibredb-add опция командной строки, 325
--enable-allow-socket-preallocation
    calibre-server опция командной строки, 318
--enable-auth
    calibre-server опция командной строки, 318
--enable-fallback-to-detected-interface
    calibre-server опция командной строки, 318
--enable-heuristics
    ebook-convert опция командной строки, 341
--enable-local-write
    calibre-server опция командной строки, 318
--enable-log-not-found
    calibre-server опция командной строки, 318
--enable-plugin
    calibre-customize опция командной строки, 314
--enable-use-bonjour
    calibre-server опция командной строки, 318
--enable-use-sendfile
    calibre-server опция командной строки, 319
--encoding
    web2disk опция командной строки, 352

--encryption-method
    calibre-smtp опция командной строки, 322
--epub-flatten
    ebook-convert опция командной строки, 337
--epub-inline-toc
    ebook-convert опция командной строки, 337
--epub-max-image-size
    ebook-convert опция командной строки, 337
--epub-toc-at-end
    ebook-convert опция командной строки, 337
--epub-version
    ebook-convert опция командной строки, 337
--exclude-genre
    calibredb-catalog опция командной строки, 329
--exclusion-rules
    calibredb-catalog опция командной строки, 329
--exec-file
    calibre-debug опция командной строки, 316
--expand-css
    ebook-convert опция командной строки, 339
--explode-book
    calibre-debug опция командной строки, 316
--export-all-calibre-data
    calibre-debug опция командной строки, 316
--extra-css
    ebook-convert опция командной строки, 339
--extract-to
    ebook-convert опция командной строки, 337
--field
    calibredb-set_metadata опция командной строки, 327
--fields
    calibredb-list опция командной строки, 324
--filter-css
    ebook-convert опция командной строки, 339
--filter-regexp
    web2disk опция командной строки, 352
--flow-size
    ebook-convert опция командной строки, 337
--font-size-mapping
    ebook-convert опция командной строки, 339
--for-machine
    calibredb-list опция командной строки, 324
--force
    calibredb-remove_custom_column опция командной строки, 332
--force-reload
    ebook-viewer опция командной строки, 349
--fork
    calibre-smtp опция командной строки, 321
--formats
    calibredb-export опция командной строки, 328
--from-opf
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 346
--full-screen
    ebook-viewer опция командной строки, 349
--fullscreen
    ebook-viewer опция командной строки, 349
--generate-authors
    calibredb-catalog опция командной строки, 329
--generate-descriptions
    calibredb-catalog опция командной строки, 330
--generate-genres
    calibredb-catalog опция командной строки, 330
--generate-recently-added
    calibredb-catalog опция командной строки, 330
--generate-series
    calibredb-catalog опция командной строки, 330
--generate-titles
    calibredb-catalog опция командной строки, 330
--genre-source-field
    calibredb-catalog опция командной строки, 330
--get-cover
    ebook-meta опция командной строки, 346
--gui
    calibre-debug опция командной строки, 316
--gui-debug
    calibre-debug опция командной строки, 316
--header-note-source-field
    calibredb-catalog опция командной строки, 330
--help
    calibre опция командной строки, 314
    calibre-customize опция командной строки, 315
    calibre-debug опция командной строки, 316
    calibre-server опция командной строки, 319
    calibre-smtp опция командной строки, 321
    ebook-convert опция командной строки, 336
    ebook-edit опция командной строки, 345
    ebook-meta опция командной строки, 346
    ebook-polish опция командной строки, 348
    ebook-viewer опция командной строки, 349

fetch-ebook-metadata опция командной строки, 350
lrf2lrs опция командной строки, 350
lrfviewer опция командной строки, 351
lrs2lrf опция командной строки, 351
web2disk опция командной строки, 352
опция командной строки, 323
--html-unwrap-factor
ebook-convert опция командной строки, 341
--identifier
calibredb-add опция командной строки, 325
ebook-meta опция командной строки, 346
fetch-ebook-metadata опция командной строки, 350
--ids
calibredb-catalog опция командной строки, 329
--ignore
calibredb-add опция командной строки, 326
--ignore_extensions
calibredb-check_library опция командной строки, 333
--ignore_names
calibredb-check_library опция командной строки, 333
--ignore-plugins
calibre опция командной строки, 314
--ignored-fields
calibre-server опция командной строки, 319
--implode-book
calibre-debug опция командной строки, 316
--import-calibre-data
calibre-debug опция командной строки, 316
--include-snippets
calibredb-fts_search опция командной строки, 335
--index
ebook-meta опция командной строки, 346
--indexing-speed
calibredb-fts_index опция командной строки, 335
--indexing-threshold
calibredb-fts_search опция командной строки, 335
--input-encoding
ebook-convert опция командной строки, 337
--input-profile
ebook-convert опция командной строки, 336
--insert-blank-line
ebook-convert опция командной строки, 339
--insert-blank-line-size
ebook-convert опция командной строки, 339
--insert-metadata
ebook-convert опция командной строки, 342
--inspect-mobi
calibre-debug опция командной строки, 316
--is-multiple
calibredb-add_custom_column опция командной строки, 331
--isbn
calibredb-add опция командной строки, 325
ebook-convert опция командной строки, 344
ebook-meta опция командной строки, 346
fetch-ebook-metadata опция командной строки, 350
--item_count
calibredb-list_categories опция командной строки, 333
--jacket
ebook-polish опция командной строки, 348
--keep-ligatures
ebook-convert опция командной строки, 339
--kepubify
calibre-debug опция командной строки, 316
--language
ebook-convert опция командной строки, 344
ebook-meta опция командной строки, 346
--languages
calibredb-add опция командной строки, 325
--level1-toc
ebook-convert опция командной строки, 343
--level2-toc
ebook-convert опция командной строки, 343
--level3-toc
ebook-convert опция командной строки, 343
--library-path
опция командной строки, 323
--limit
calibredb-list опция командной строки, 324
calibredb-search опция командной строки, 334
--line-height
ebook-convert опция командной строки, 339
--line-width
calibredb-list опция командной строки, 324
--linearize-tables
ebook-convert опция командной строки, 339
--list-fields
calibredb-set_metadata опция командной строки, 327
--list-plugins
calibre-customize опция командной строки, 315
--list-recipes
ebook-convert опция командной строки, 336

--listen-on
    calibre-server опция командной строки, 319
--localhost
    calibre-smtp опция командной строки, 321
--log
    calibre-server опция командной строки, 319
--lrf-bookid
    ebook-meta опция командной строки, 346
--lrs
    lrs2lrf опция командной строки, 351
--manage-users
    calibre-server опция командной строки, 319
--margin-bottom
    ebook-convert опция командной строки, 339
--margin-left
    ebook-convert опция командной строки, 340
--margin-right
    ebook-convert опция командной строки, 340
--margin-top
    ebook-convert опция командной строки, 340
--match-end-marker
    calibredb-fts_search опция командной строки, 335
--match-regexp
    web2disk опция командной строки, 352
--match-start-marker
    calibredb-fts_search опция командной строки, 335
--max-files
    web2disk опция командной строки, 352
--max-header-line-size
    calibre-server опция командной строки, 319
--max-job-time
    calibre-server опция командной строки, 319
--max-jobs
    calibre-server опция командной строки, 319
--max-log-size
    calibre-server опция командной строки, 319
--max-opds-items
    calibre-server опция командной строки, 319
--max-opds-ungrouped-items
    calibre-server опция командной строки, 319
--max-recursions
    web2disk опция командной строки, 352
--max-request-body-size
    calibre-server опция командной строки, 320
--max-toc-links
    ebook-convert опция командной строки, 343
--merge-comments-rule
    calibredb-catalog опция командной строки, 330
--minimum-line-height
    ebook-convert опция командной строки, 340
--new-instance
    ebook-viewer опция командной строки, 349
--no-chapters-in-toc
    ebook-convert опция командной строки, 343
--no-default-epub-cover
    ebook-convert опция командной строки, 338
--no-svg-cover
    ebook-convert опция командной строки, 338
--no-update-check
    calibre опция командной строки, 314
--num-per-page
    calibre-server опция командной строки, 320
--one-book-per-directory
    calibredb-add опция командной строки, 326
--only-formats
    calibredb-embed_metadata опция командной строки, 334
--open-at
    ebook-viewer опция командной строки, 349
--opf
    ebook-polish опция командной строки, 348
    fetch-ebook-metadata опция командной строки, 350
--outbox
    calibre-smtp опция командной строки, 321
--output
    lrf2lrs опция командной строки, 350
    lrs2lrf опция командной строки, 352
--output-format
    calibredb-fts_search опция командной строки, 335
--output-profile
    calibredb-catalog опция командной строки, 330
    ebook-convert опция командной строки, 336
--page-breaks-before
    ebook-convert опция командной строки, 342
--password
    calibre-smtp опция командной строки, 322
    опция командной строки, 324
--paths
    calibre-debug опция командной строки, 316
--permanent

calibredb-remove опция командной строки, 326
--pidfile
calibre-server опция командной строки, 320
--port
calibre-server опция командной строки, 320
calibre-smtp опция командной строки, 322
--prefer-metadata-cover
ebook-convert опция командной строки, 343
--prefix
calibredb-list опция командной строки, 324
--prefix-rules
calibredb-catalog опция командной строки, 330
--preserve-cover-aspect-ratio
ebook-convert опция командной строки, 338
--preset
calibredb-catalog опция командной строки, 330
--pretty-print
ebook-convert опция командной строки, 338
--profile
lrfviewer опция командной строки, 351
--progress
calibredb-export опция командной строки, 328
--pubdate
ebook-convert опция командной строки, 344
--publisher
ebook-convert опция командной строки, 344
ebook-meta опция командной строки, 346
--raise-window
ebook-viewer опция командной строки, 349
--rating
ebook-convert опция командной строки, 344
ebook-meta опция командной строки, 347
--read-metadata-from-opf
ebook-convert опция командной строки, 344
--really-do-it
calibredb-restore_database опция командной строки, 332
--recurse
calibredb-add опция командной строки, 326
--relay
calibre-smtp опция командной строки, 322
--remove-first-image
ebook-convert опция командной строки, 343
--remove-jacket
ebook-polish опция командной строки, 348
--remove-paragraph-spacing
ebook-convert опция командной строки, 340
--remove-paragraph-spacing-indent-size
ebook-convert опция командной строки, 340
--remove-plugin
calibre-customize опция командной строки, 315
--remove-soft-hyphens
ebook-polish опция командной строки, 348
--remove-unused-css
ebook-polish опция командной строки, 348
--replace-scene-breaks
ebook-convert опция командной строки, 341
--replace-whitespace
calibredb-export опция командной строки, 328
--report
calibredb-check_library опция командной строки, 333
--restrict-to
calibredb-fts_search опция командной строки, 335
--run-plugin
calibre-debug опция командной строки, 316
--run-test
calibre-debug опция командной строки, 316
--run-without-debug
calibre-debug опция командной строки, 317
--search
calibredb-catalog опция командной строки, 329
calibredb-list опция командной строки, 324
--search-replace
ebook-convert опция командной строки, 341
--search-the-net-urls
calibre-server опция командной строки, 320
--select-text
ebook-edit опция командной строки, 345
--separator
calibredb-list опция командной строки, 324
--series
calibredb-add опция командной строки, 325
ebook-convert опция командной строки, 344
ebook-meta опция командной строки, 347
--series-index
calibredb-add опция командной строки, 326
ebook-convert опция командной строки, 344
--shutdown-running-calibre
calibre опция командной строки, 314
calibre-debug опция командной строки, 317
--shutdown-timeout
calibre-server опция командной строки, 320

--single-dir
    calibredb-export опция командной строки, 328
--smarten-punctuation
    ebook-convert опция командной строки, 340
    ebook-polish опция командной строки, 348
--sort-by
    calibredb-list опция командной строки, 325
--sr1-replace
    ebook-convert опция командной строки, 342
--sr1-search
    ebook-convert опция командной строки, 342
--sr2-replace
    ebook-convert опция командной строки, 342
--sr2-search
    ebook-convert опция командной строки, 342
--sr3-replace
    ebook-convert опция командной строки, 342
--sr3-search
    ebook-convert опция командной строки, 342
--ssl-certfile
    calibre-server опция командной строки, 320
--ssl-keyfile
    calibre-server опция командной строки, 320
--start-in-tray
    calibre опция командной строки, 314
--start-reading-at
    ebook-convert опция командной строки, 343
--subject
    calibre-smtp опция командной строки, 321
--subset-embedded-fonts
    ebook-convert опция командной строки, 340
--subset-font
    calibre-debug опция командной строки, 317
--subset-fonts
    ebook-polish опция командной строки, 348
--tags
    calibredb-add опция командной строки, 326
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 347
--template
    calibredb-export опция командной строки, 328
    calibredb-list опция командной строки, 325
--template_file
    calibredb-list опция командной строки, 325
--template_heading
    calibredb-list опция командной строки, 325
--test-build
    calibre-debug опция командной строки, 317
--thumb-width
    calibredb-catalog опция командной строки, 330
--timefmt
    calibredb-export опция командной строки, 328
--timeout
    calibre-server опция командной строки, 320
    calibre-smtp опция командной строки, 321
    fetch-ebook-metadata опция командной строки, 350
    web2disk опция командной строки, 352
    опция командной строки, 324
--timestamp
    ebook-convert опция командной строки, 344
--title
    calibredb-add опция командной строки, 326
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 347
    fetch-ebook-metadata опция командной строки, 350
--title-sort
    ebook-convert опция командной строки, 344
    ebook-meta опция командной строки, 347
--to-dir
    calibredb-export опция командной строки, 328
--to-lowercase
    calibredb-export опция командной строки, 329
--to-opf
    ebook-meta опция командной строки, 347
--toc-filter
    ebook-convert опция командной строки, 343
--toc-threshold
    ebook-convert опция командной строки, 343
--toc-title
    ebook-convert опция командной строки, 338
--transform-css-rules
    ebook-convert опция командной строки, 340
--transform-html-rules
    ebook-convert опция командной строки, 340
--trusted-ips
    calibre-server опция командной строки, 320
--un-kepubify
    calibre-debug опция командной строки, 317
--unsmarten-punctuation
    ebook-convert опция командной строки, 340
--upgrade-book
    ebook-polish опция командной строки, 348

--url-prefix
    calibre-server опция командной строки, 320
--use-auto-toc
    ebook-convert опция командной строки, 343
--use-existing-cover
    calibredb-catalog опция командной строки, 330
--userdb
    calibre-server опция командной строки, 320
--username
    calibre-smtp опция командной строки, 322
    опция командной строки, 324
--vacuum-fts-db
    calibredb-check_library опция командной строки, 333
--verbose
    calibre опция командной строки, 314
    calibre-smtp опция командной строки, 321
    calibredb-catalog опция командной строки, 329
    ebook-convert опция командной строки, 345
    ebook-polish опция командной строки, 348
    fetch-ebook-metadata опция командной строки, 350
    lrf2lrs опция командной строки, 351
    lrfviewer опция командной строки, 351
    lrs2lrf опция командной строки, 352
    web2disk опция командной строки, 353
--version
    calibre опция командной строки, 314
    calibre-customize опция командной строки, 315
    calibre-debug опция командной строки, 317
    calibre-server опция командной строки, 320
    calibre-smtp опция командной строки, 321
    ebook-convert опция командной строки, 337
    ebook-edit опция командной строки, 345
    ebook-meta опция командной строки, 347
    ebook-polish опция командной строки, 348
    ebook-viewer опция командной строки, 349
    fetch-ebook-metadata опция командной строки, 350
    lrf2lrs опция командной строки, 351
    lrfviewer опция командной строки, 351
    lrs2lrf опция командной строки, 352
    web2disk опция командной строки, 353
    опция командной строки, 324
--viewer
    calibre-debug опция командной строки, 317
--visual-debug
    lrfviewer опция командной строки, 351
--wait-for-completion
    calibredb-fts_index опция командной строки, 335
--white-background
    lrfviewer опция командной строки, 351
--width
    calibredb-list_categories опция командной строки, 333
--with-library
    calibre опция командной строки, 314
    опция командной строки, 323
--worker-count
    calibre-server опция командной строки, 320
-a
    calibre-customize опция командной строки, 314
    calibre-smtp опция командной строки, 321
    calibredb-add опция командной строки, 325
    calibredb-set_custom опция командной строки, 332
    ebook-meta опция командной строки, 346
    fetch-ebook-metadata опция командной строки, 350
-b
    calibre-customize опция командной строки, 314
-c
    calibre-debug опция командной строки, 315
    calibredb-add опция командной строки, 325
    calibredb-check_library опция командной строки, 332
    calibredb-list_categories опция командной строки, 333
    ebook-meta опция командной строки, 346
    ebook-polish опция командной строки, 347
    fetch-ebook-metadata опция командной строки, 350
-d
    calibre-debug опция командной строки, 315
    calibredb-add опция командной строки, 325
    calibredb-custom_columns опция командной строки, 331
    ebook-convert опция командной строки, 345
    ebook-meta опция командной строки, 346
    ebook-polish опция командной строки, 347
    fetch-ebook-metadata опция командной строки, 350
    web2disk опция командной строки, 352
-e
    calibre-debug опция командной строки, 316
    calibre-smtp опция командной строки, 322
    calibredb-add опция командной строки, 325

calibredb-check_library опция командной строки, 333
ebook-polish опция командной строки, 348
-f
calibre-debug опция командной строки, 317
calibre-smtp опция командной строки, 321
calibredb-embed_metadata опция командной строки, 334
calibredb-list опция командной строки, 324
calibredb-remove_custom_column опция командной строки, 332
calibredb-set_metadata опция командной строки, 327
ebook-polish опция командной строки, 348
ebook-viewer опция командной строки, 349
-g
calibre-debug опция командной строки, 316
-h
calibre опция командной строки, 314
calibre-customize опция командной строки, 315
calibre-debug опция командной строки, 316
calibre-server опция командной строки, 319
calibre-smtp опция командной строки, 321
ebook-convert опция командной строки, 336
ebook-edit опция командной строки, 345
ebook-meta опция командной строки, 346
ebook-polish опция командной строки, 348
ebook-viewer опция командной строки, 349
fetch-ebook-metadata опция командной строки, 350
lrf2lrs опция командной строки, 350
lrfviewer опция командной строки, 351
lrs2lrf опция командной строки, 351
web2disk опция командной строки, 352
опция командной строки, 323
-i
calibre-debug опция командной строки, 316
calibredb-add опция командной строки, 325
calibredb-catalog опция командной строки, 329
calibredb-list_categories опция командной строки, 333
ebook-meta опция командной строки, 346
ebook-polish опция командной строки, 347
fetch-ebook-metadata опция командной строки, 350
-j
ebook-polish опция командной строки, 348
-k
ebook-meta опция командной строки, 346
-l
calibre-customize опция командной строки, 315
calibre-smtp опция командной строки, 321
calibredb-add опция командной строки, 325
calibredb-search опция командной строки, 334
calibredb-set_metadata опция командной строки, 327
ebook-meta опция командной строки, 346
-m
calibre-debug опция командной строки, 316
calibredb-add опция командной строки, 325
ebook-convert опция командной строки, 344
-n
calibredb-check_library опция командной строки, 333
web2disk опция командной строки, 352
-o
calibre-smtp опция командной строки, 321
ebook-polish опция командной строки, 348
fetch-ebook-metadata опция командной строки, 350
lrf2lrs опция командной строки, 350
lrs2lrf опция командной строки, 352
-p
calibre-smtp опция командной строки, 322
ebook-meta опция командной строки, 346
ebook-polish опция командной строки, 348
fetch-ebook-metadata опция командной строки, 350
-r
calibre-customize опция командной строки, 315
calibre-debug опция командной строки, 316
calibre-smtp опция командной строки, 322
calibredb-add опция командной строки, 326
calibredb-check_library опция командной строки, 333
calibredb-list_categories опция командной строки, 333
calibredb-restore_database опция командной строки, 332
ebook-meta опция командной строки, 347
web2disk опция командной строки, 352
-s
calibre опция командной строки, 314
calibre-debug опция командной строки, 317
calibre-smtp опция командной строки, 321
calibredb-add опция командной строки, 325
calibredb-catalog опция командной строки, 329
calibredb-list опция командной строки, 324
ebook-meta опция командной строки, 347

-t
    calibre-debug опция командной строки, 316
    calibre-smtp опция командной строки, 321
    calibredb-add опция командной строки, 326
    calibredb-list опция командной строки, 325
    ebook-meta опция командной строки, 347
    fetch-ebook-metadata опция командной строки, 350
    web2disk опция командной строки, 352
-u
    calibre-smtp опция командной строки, 322
    ebook-polish опция командной строки, 348
-v
    calibre опция командной строки, 314
    calibre-smtp опция командной строки, 321
    calibredb-catalog опция командной строки, 329
    ebook-convert опция командной строки, 345
    fetch-ebook-metadata опция командной строки, 350
-w
    calibre-debug опция командной строки, 317
    calibredb-list опция командной строки, 324
    calibredb-list_categories опция командной строки, 333
-x
    calibre-debug опция командной строки, 316

A

abort_article() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 42
abort_recipe_processing() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 42
abspath_to_name() (*метод calibre.ebooks.oeb.polish.container.Container*), 375
accept_drag_move_event() (*метод calibre.gui2.actions.InterfaceAction*), 293
accept_enter_event() (*метод calibre.gui2.actions.InterfaceAction*), 293
accepts_drops (*атрибут calibre.gui2.actions.InterfaceAction*), 293
action_add_menu (*атрибут calibre.gui2.actions.InterfaceAction*), 293
action_menu_clone_qaction (*атрибут calibre.gui2.actions.InterfaceAction*), 293
action_shortcut_name (*атрибут calibre.gui2.actions.InterfaceAction*), 293
action_spec (*атрибут calibre.gui2.actions.InterfaceAction*), 292
action_type (*атрибут calibre.gui2.actions.InterfaceAction*), 293
add_annotation_to_library() (*метод calibre.devices.usbms.device.Device*), 289
add_book() (*метод calibre.devices.interface.BookList*), 286
add_books() (*метод calibre.db.cache.Cache*), 364
add_books_to_metadata() (*метод calibre.devices.usbms.driver.USBMS*), 291
add_books_to_metadata() (*метод класса calibre.devices.interface.DevicePlugin*), 282
add_custom_book_data() (*метод calibre.db.cache.Cache*), 365
add_extra_files() (*метод calibre.db.cache.Cache*), 365
add_file() (*метод calibre.ebooks.oeb.polish.container.Container*), 375
add_format() (*метод calibre.db.cache.Cache*), 365
add_listener() (*метод calibre.db.cache.Cache*), 365
add_name_to_manifest() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
add_notes_resource() (*метод calibre.db.cache.Cache*), 365
add_or_replace_jacket() (*в модуле calibre.ebooks.oeb.polish.jacket*), 380
add_properties() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
add_savepoint() (*метод calibre.gui2.tweak_book.boss.Boss*), 384
add_toc_thumbnail() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 42
adeify_images() (*метод класса calibre.web.feeds.news.BasicNewsRecipe*), 41
all_annotation_types() (*метод calibre.db.cache.Cache*), 365
all_annotation_users() (*метод calibre.db.cache.Cache*), 365
all_annotations() (*метод calibre.db.cache.Cache*), 365
all_annotations_for_book() (*метод calibre.db.cache.Cache*), 365
all_book_ids() (*метод calibre.db.cache.Cache*), 365
all_field_for() (*метод calibre.db.cache.Cache*), 365
all_field_ids() (*метод calibre.db.cache.Cache*), 365
all_field_keys() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
all_field_names() (*метод calibre.db.cache.Cache*), 366
all_non_none_fields() (*метод*

calibre.ebooks.metadata.book.base.Metadata), 217
allowed_in_menu (*атрибут calibre.gui2.tweak_book.plugin.Tool*), 383
allowed_in_toolbar (*атрибут calibre.gui2.tweak_book.plugin.Tool*), 383
annotation_count_for_book() (*метод calibre.db.cache.Cache*), 366
annotations_map_for_book() (*метод calibre.db.cache.Cache*), 366
API, **389**
apply_container_update_to_gui() (*метод calibre.gui2.tweak_book.boss.Boss*), 384
apply_unique_properties() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
articles_are_obfuscated (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 46
ASK_TO_ALLOW_CONNECT (*атрибут calibre.devices.interface.DevicePlugin*), 279
author (*атрибут calibre.customize.InterfaceActionBase*), 295
author (*атрибут calibre.customize.MetadataReaderPlugin*), 268
author (*атрибут calibre.customize.MetadataWriterPlugin*), 269
author (*атрибут calibre.customize.Plugin*), 265
author (*атрибут calibre.customize.PreferencesPlugin*), 296
author (*атрибут calibre.devices.usbms.driver.USBMS*), 290
author (*атрибут calibre.ebooks.metadata.sources.base.Source*), 270
author_data() (*метод calibre.db.cache.Cache*), 366
author_sort_from_authors() (*метод calibre.db.cache.Cache*), 366
auto_cleanup (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 46
auto_cleanup_keep (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 47
auto_repeat (*атрибут calibre.gui2.actions.InterfaceAction*), 292
auto_trim_covers (*атрибут calibre.ebooks.metadata.sources.base.Source*), 271

B

BasicNewsRecipe (*класс в calibre.web.feeds.news*), 41
BCD (*атрибут calibre.devices.interface.DevicePlugin*), 277
BCD (*атрибут calibre.devices.usbms.device.Device*), 287
book_class (*атрибут calibre.devices.usbms.driver.USBMS*), 290
book_created (*атрибут calibre.db.cache.Cache.EventType*), 364
book_edited (*атрибут calibre.db.cache.Cache.EventType*), 364
book_type (*атрибут calibre.ebooks.oeb.polish.container.Container*), 376
BookList (*класс в calibre.devices.interface*), 285
booklist_class (*атрибут calibre.devices.usbms.driver.USBMS*), 290
books() (*метод calibre.devices.interface.DevicePlugin*), 281
books() (*метод calibre.devices.usbms.driver.USBMS*), 290
books_for_field() (*метод calibre.db.cache.Cache*), 366
books_in_virtual_library() (*метод calibre.db.cache.Cache*), 366
books_removed (*атрибут calibre.db.cache.Cache.EventType*), 364
Boss (*класс в calibre.gui2.tweak_book.boss*), 384
boss (*свойство calibre.gui2.tweak_book.plugin.Tool*), 383
browser_type (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 47
BuiltinAdd (*класс в calibre.utils.formatter_functions*), 191
BuiltinAnd (*класс в calibre.utils.formatter_functions*), 192
BuiltinAnnotationCount (*класс в calibre.utils.formatter_functions*), 203
BuiltinApproximateFormats (*класс в calibre.utils.formatter_functions*), 203
BuiltinArguments (*класс в calibre.utils.formatter_functions*), 193
BuiltinAssign (*класс в calibre.utils.formatter_functions*), 193
BuiltinAuthorLinks (*класс в calibre.utils.formatter_functions*), 203
BuiltinAuthorSorts (*класс в calibre.utils.formatter_functions*), 204
BuiltinBookCount (*класс в calibre.utils.formatter_functions*), 213
BuiltinBooksize (*класс в calibre.utils.formatter_functions*), 204
BuiltinBookValues (*класс в calibre.utils.formatter_functions*), 214
BuiltinCapitalize (*класс в calibre.utils.formatter_functions*), 208
BuiltinCeiling (*класс в calibre.utils.formatter_functions*), 191
BuiltinCharacter (*класс в*

calibre.utils.formatter_functions), 194
BuiltinCheckYesNo (*класс в calibre.utils.formatter_functions*), 194
BuiltinCmp (*класс в calibre.utils.formatter_functions*), 200
BuiltinConnectedDeviceName (*класс в calibre.utils.formatter_functions*), 204
BuiltinConnectedDeviceUUID (*класс в calibre.utils.formatter_functions*), 204
BuiltinContains (*класс в calibre.utils.formatter_functions*), 194
BuiltinCount (*класс в calibre.utils.formatter_functions*), 196
BuiltinCurrentLibraryName (*класс в calibre.utils.formatter_functions*), 204
BuiltinCurrentLibraryPath (*класс в calibre.utils.formatter_functions*), 204
BuiltinCurrentVirtualLibraryName (*класс в calibre.utils.formatter_functions*), 204
BuiltinDateArithmetic (*класс в calibre.utils.formatter_functions*), 216
BuiltinDaysBetween (*класс в calibre.utils.formatter_functions*), 216
BuiltinDivide (*класс в calibre.utils.formatter_functions*), 191
BuiltinEncodeForURL (*класс в calibre.utils.formatter_functions*), 210
BuiltinEval (*класс в calibre.utils.formatter_functions*), 207
BuiltinExtraFileModtime (*класс в calibre.utils.formatter_functions*), 214
BuiltinExtraFileNames (*класс в calibre.utils.formatter_functions*), 214
BuiltinExtraFileSize (*класс в calibre.utils.formatter_functions*), 214
BuiltinField (*класс в calibre.utils.formatter_functions*), 205
BuiltinFieldExists (*класс в calibre.utils.formatter_functions*), 194
BuiltinFieldListCount (*класс в calibre.utils.formatter_functions*), 197
BuiltinFinishFormatting (*класс в calibre.utils.formatter_functions*), 208
BuiltinFirstMatchingCmp (*класс в calibre.utils.formatter_functions*), 201
BuiltinFirstNonEmpty (*класс в calibre.utils.formatter_functions*), 201
BuiltinFloor (*класс в calibre.utils.formatter_functions*), 191
BuiltinFormatDate (*класс в calibre.utils.formatter_functions*), 209
BuiltinFormatDateField (*класс в calibre.utils.formatter_functions*), 210
BuiltinFormatNumber (*класс в calibre.utils.formatter_functions*), 210
BuiltinFormatsModtimes (*класс в calibre.utils.formatter_functions*), 205
BuiltinFormatsPaths (*класс в calibre.utils.formatter_functions*), 205
BuiltinFormatsSizes (*класс в calibre.utils.formatter_functions*), 205
BuiltinFractionalPart (*класс в calibre.utils.formatter_functions*), 191
BuiltinGetLink (*класс в calibre.utils.formatter_functions*), 215
BuiltinGetNote (*класс в calibre.utils.formatter_functions*), 215
BuiltinGlobals (*класс в calibre.utils.formatter_functions*), 193
BuiltinHasCover (*класс в calibre.utils.formatter_functions*), 205
BuiltinHasExtraFiles (*класс в calibre.utils.formatter_functions*), 215
BuiltinHasNote (*класс в calibre.utils.formatter_functions*), 215
BuiltinHumanReadable (*класс в calibre.utils.formatter_functions*), 210
BuiltinIdentifierInList (*класс в calibre.utils.formatter_functions*), 202
BuiltinIfempty (*класс в calibre.utils.formatter_functions*), 194
BuiltinInList (*класс в calibre.utils.formatter_functions*), 202
BuiltinIsDarkMode (*класс в calibre.utils.formatter_functions*), 193
BuiltinIsMarked (*класс в calibre.utils.formatter_functions*), 206
BuiltinLanguageCodes (*класс в calibre.utils.formatter_functions*), 206
BuiltinLanguageStrings (*класс в calibre.utils.formatter_functions*), 206
BuiltinListCountMatching (*класс в calibre.utils.formatter_functions*), 197
BuiltinListDifference (*класс в calibre.utils.formatter_functions*), 197
BuiltinListEquals (*класс в calibre.utils.formatter_functions*), 197
BuiltinListIntersection (*класс в calibre.utils.formatter_functions*), 197
BuiltinListitem (*класс в calibre.utils.formatter_functions*), 202
BuiltinListJoin (*класс в calibre.utils.formatter_functions*), 198
BuiltinListRe (*класс в calibre.utils.formatter_functions*), 198
BuiltinListReGroup (*класс в calibre.utils.formatter_functions*), 198
BuiltinListRemoveDuplicates (*класс в*

calibre.utils.formatter_functions), 198
BuiltinListSort (*класс в calibre.utils.formatter_functions*), 199
BuiltinListSplit (*класс в calibre.utils.formatter_functions*), 199
BuiltinListUnion (*класс в calibre.utils.formatter_functions*), 199
BuiltinLookup (*класс в calibre.utils.formatter_functions*), 201
BuiltinLowercase (*класс в calibre.utils.formatter_functions*), 208
BuiltinMakeUrl (*класс в calibre.utils.formatter_functions*), 211
BuiltinMakeUrlExtended (*класс в calibre.utils.formatter_functions*), 211
BuiltinMod (*класс в calibre.utils.formatter_functions*), 192
BuiltinMultiply (*класс в calibre.utils.formatter_functions*), 192
BuiltinNot (*класс в calibre.utils.formatter_functions*), 192
BuiltinOndevice (*класс в calibre.utils.formatter_functions*), 206
BuiltinOr (*класс в calibre.utils.formatter_functions*), 192
BuiltinPrint (*класс в calibre.utils.formatter_functions*), 193
BuiltinQueryString (*класс в calibre.utils.formatter_functions*), 212
BuiltinRange (*класс в calibre.utils.formatter_functions*), 199
BuiltinRatingToStars (*класс в calibre.utils.formatter_functions*), 210
BuiltinRawField (*класс в calibre.utils.formatter_functions*), 206
BuiltinRawList (*класс в calibre.utils.formatter_functions*), 206
BuiltinRe (*класс в calibre.utils.formatter_functions*), 195
BuiltinReGroup (*класс в calibre.utils.formatter_functions*), 195
BuiltinRound (*класс в calibre.utils.formatter_functions*), 192
BuiltinSelect (*класс в calibre.utils.formatter_functions*), 203
BuiltinSeriesSort (*класс в calibre.utils.formatter_functions*), 207
BuiltinSetGlobals (*класс в calibre.utils.formatter_functions*), 193
BuiltinShorten (*класс в calibre.utils.formatter_functions*), 195
BuiltinStrcat (*класс в calibre.utils.formatter_functions*), 195
BuiltinStrcatMax (*класс в calibre.utils.formatter_functions*), 195
BuiltinStrcmp (*класс в calibre.utils.formatter_functions*), 201
BuiltinStrcmpcase (*класс в calibre.utils.formatter_functions*), 201
BuiltinStrInList (*класс в calibre.utils.formatter_functions*), 203
BuiltinStrlen (*класс в calibre.utils.formatter_functions*), 196
BuiltinSubitems (*класс в calibre.utils.formatter_functions*), 200
BuiltinSublist (*класс в calibre.utils.formatter_functions*), 200
BuiltinSubstr (*класс в calibre.utils.formatter_functions*), 196
BuiltinSubtract (*класс в calibre.utils.formatter_functions*), 192
BuiltinSwapAroundArticles (*класс в calibre.utils.formatter_functions*), 196
BuiltinSwapAroundComma (*класс в calibre.utils.formatter_functions*), 196
BuiltinSwitch (*класс в calibre.utils.formatter_functions*), 201
BuiltinSwitchIf (*класс в calibre.utils.formatter_functions*), 202
BuiltinTemplate (*класс в calibre.utils.formatter_functions*), 207
BuiltinTest (*класс в calibre.utils.formatter_functions*), 196
BuiltinTitlecase (*класс в calibre.utils.formatter_functions*), 208
BuiltinToday (*класс в calibre.utils.formatter_functions*), 216
BuiltinToHex (*класс в calibre.utils.formatter_functions*), 213
BuiltinTransliterate (*класс в calibre.utils.formatter_functions*), 196
BuiltinUppercase (*класс в calibre.utils.formatter_functions*), 208
BuiltinUrlsFromIdentifiers (*класс в calibre.utils.formatter_functions*), 213
BuiltinUserCategories (*класс в calibre.utils.formatter_functions*), 207
BuiltinVirtualLibraries (*класс в calibre.utils.formatter_functions*), 207

C

Cache (*класс в calibre.db.cache*), 364
Cache.EventType (*класс в calibre.db.cache*), 364
cached_cover_url_is_reliable (*атрибут calibre.ebooks.metadata.sources.base.Source*), 271
calibre опция командной строки
    --detach, 314

--help, 314
--ignore-plugins, 314
--no-update-check, 314
--shutdown-running-calibre, 314
--start-in-tray, 314
--verbose, 314
--version, 314
--with-library, 314
-h, 314
-s, 314
-v, 314
calibre.customize
module, 264
calibre.customize.conversion
module, 274
calibre.db.cache
module, 363
calibre.devices.interface
module, 277
calibre.ebooks.metadata.book.base
module, 216
calibre.ebooks.metadata.sources.base
module, 270
calibre.ebooks.oeb.polish.container
module, 375
calibre.ebooks.oeb.polish.cover
module, 381
calibre.ebooks.oeb.polish.css
module, 382
calibre.ebooks.oeb.polish.jacket
module, 380
calibre.ebooks.oeb.polish.pretty
module, 380
calibre.ebooks.oeb.polish.replace
module, 379
calibre.ebooks.oeb.polish.split
module, 380
calibre.ebooks.oeb.polish.toc
module, 382
calibre.gui2.tweak_book.boss
module, 384
calibre.utils.formatter_functions
module, 191
calibre.web.feeds.news
module, 41
calibre-customize опция командной строки
--add-plugin, 314
--build-plugin, 314
--customize-plugin, 314
--disable-plugin, 314
--enable-plugin, 314
--help, 315
--list-plugins, 315
--remove-plugin, 315
--version, 315
-a, 314
-b, 314
-h, 315
-l, 315
-r, 315
calibre-debug опция командной строки
--add-simple-plugin, 315
--command, 315
--debug-device-driver, 315
--default-programs, 315
--diff, 315
--edit-book, 315
--exec-file, 316
--explode-book, 316
--export-all-calibre-data, 316
--gui, 316
--gui-debug, 316
--help, 316
--implode-book, 316
--import-calibre-data, 316
--inspect-mobi, 316
--kepubify, 316
--paths, 316
--run-plugin, 316
--run-test, 316
--run-without-debug, 317
--shutdown-running-calibre, 317
--subset-font, 317
--test-build, 317
--un-kepubify, 317
--version, 317
--viewer, 317
-c, 315
-d, 315
-e, 316
-f, 317
-g, 316
-h, 316
-i, 316
-m, 316
-r, 316
-s, 317
-t, 316
-w, 317
-x, 316
calibre-server опция командной строки
--access-log, 317
--ajax-timeout, 317
--auth-mode, 317
--auto-reload, 317
--ban-after, 317
--ban-for, 317
--book-list-mode, 318

--compress-min-size, 318
--custom-list-template, 318
--daemonize, 318
--disable-allow-socket-preallocation, 318
--disable-auth, 318
--disable-fallback-to-detected-interface, 318
--disable-local-write, 318
--disable-log-not-found, 318
--disable-use-bonjour, 318
--disable-use-sendfile, 319
--displayed-fields, 318
--enable-allow-socket-preallocation, 318
--enable-auth, 318
--enable-fallback-to-detected-interface, 318
--enable-local-write, 318
--enable-log-not-found, 318
--enable-use-bonjour, 318
--enable-use-sendfile, 319
--help, 319
--ignored-fields, 319
--listen-on, 319
--log, 319
--manage-users, 319
--max-header-line-size, 319
--max-job-time, 319
--max-jobs, 319
--max-log-size, 319
--max-opds-items, 319
--max-opds-ungrouped-items, 319
--max-request-body-size, 320
--num-per-page, 320
--pidfile, 320
--port, 320
--search-the-net-urls, 320
--shutdown-timeout, 320
--ssl-certfile, 320
--ssl-keyfile, 320
--timeout, 320
--trusted-ips, 320
--url-prefix, 320
--userdb, 320
--version, 320
--worker-count, 320
-h, 319

calibre-smtp опция командной строки
--attachment, 321
--cafile, 322
--dont-verify-server-certificate, 322
--encryption-method, 322
--fork, 321
--help, 321
--localhost, 321
--outbox, 321
--password, 322
--port, 322
--relay, 322
--subject, 321
--timeout, 321
--username, 322
--verbose, 321
--version, 321
-a, 321
-e, 322
-f, 321
-h, 321
-l, 321
-o, 321
-p, 322
-r, 322
-s, 321
-t, 321
-u, 322
-v, 321

calibredb-add опция командной строки
-1, 326
-I, 325
-S, 326
-T, 326
--add, 326
--authors, 325
--automerge, 325
--cover, 325
--duplicates, 325
--empty, 325
--identifier, 325
--ignore, 326
--isbn, 325
--languages, 325
--one-book-per-directory, 326
--recurse, 326
--series, 325
--series-index, 326
--tags, 326
--title, 326
-a, 325
-c, 325
-d, 325
-e, 325
-i, 325
-l, 325
-m, 325
-r, 326
-s, 325
-t, 326

calibredb-add_custom_column опция командной строки
--display, 331
--is-multiple, 331
calibredb-add_format опция командной строки
--as-extra-data-file, 327
--dont-replace, 327
calibredb-backup_metadata опция командной строки
--all, 333
calibredb-catalog опция командной строки
--catalog-title, 329
--cross-reference-authors, 329
--debug-pipeline, 329
--exclude-genre, 329
--exclusion-rules, 329
--generate-authors, 329
--generate-descriptions, 330
--generate-genres, 330
--generate-recently-added, 330
--generate-series, 330
--generate-titles, 330
--genre-source-field, 330
--header-note-source-field, 330
--ids, 329
--merge-comments-rule, 330
--output-profile, 330
--prefix-rules, 330
--preset, 330
--search, 329
--thumb-width, 330
--use-existing-cover, 330
--verbose, 329
-i, 329
-s, 329
-v, 329
calibredb-check_library опция командной строки
--csv, 332
--ignore_extensions, 333
--ignore_names, 333
--report, 333
--vacuum-fts-db, 333
-c, 332
-e, 333
-n, 333
-r, 333
calibredb-custom_columns опция командной строки
--details, 331
-d, 331
calibredb-embed_metadata опция командной строки
--only-formats, 334
-f, 334
calibredb-export опция командной строки
--all, 328
--dont-asciiize, 328
--dont-save-cover, 328
--dont-save-extra-files, 328
--dont-update-metadata, 328
--dont-write-opf, 328
--formats, 328
--progress, 328
--replace-whitespace, 328
--single-dir, 328
--template, 328
--timefmt, 328
--to-dir, 328
--to-lowercase, 329
calibredb-fts_index опция командной строки
--indexing-speed, 335
--wait-for-completion, 335
calibredb-fts_search опция командной строки
--do-not-match-on-related-words, 335
--include-snippets, 335
--indexing-threshold, 335
--match-end-marker, 335
--match-start-marker, 335
--output-format, 335
--restrict-to, 335
calibredb-list опция командной строки
--ascending, 324
--fields, 324
--for-machine, 324
--limit, 324
--line-width, 324
--prefix, 324
--search, 324
--separator, 324
--sort-by, 325
--template, 325
--template_file, 325
--template_heading, 325
-f, 324
-s, 324
-t, 325
-w, 324
calibredb-list_categories опция командной строки
--categories, 333
--csv, 333
--dialect, 333
--item_count, 333
--width, 333
-c, 333
-i, 333
-r, 333

-w, 333
calibredb-remove опция командной строки
--permanent, 326
calibredb-remove_custom_column опция командной строки
--force, 332
-f, 332
calibredb-restore_database опция командной строки
--really-do-it, 332
-r, 332
calibredb-search опция командной строки
--limit, 334
-l, 334
calibredb-set_custom опция командной строки
--append, 332
-a, 332
calibredb-set_metadata опция командной строки
--field, 327
--list-fields, 327
-f, 327
-l, 327
calibredb-show_metadata опция командной строки
--as-opf, 327
can_be_disabled (*атрибут calibre.customize.conversion.InputFormatPlugin*), 274
can_be_disabled (*атрибут calibre.customize.conversion.OutputFormatPlugin*), 276
can_be_disabled (*атрибут calibre.customize.InterfaceActionBase*), 295
can_be_disabled (*атрибут calibre.customize.Plugin*), 265
can_be_disabled (*атрибут calibre.customize.PreferencesPlugin*), 296
CAN_DO_DEVICE_DB_PLUGBOARD (*атрибут calibre.devices.interface.DevicePlugin*), 278
can_get_multiple_covers (*атрибут calibre.ebooks.metadata.sources.base.Source*), 271
can_handle() (*метод calibre.devices.interface.DevicePlugin*), 280
can_handle_windows() (*метод calibre.devices.interface.DevicePlugin*), 280
can_handle_windows() (*метод calibre.devices.usbms.device.Device*), 288
CAN_SET_METADATA (*атрибут calibre.devices.interface.DevicePlugin*), 278
CAN_SET_METADATA (*атрибут calibre.devices.usbms.driver.USBMS*), 290
canonicalize_internal_url() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 42
capabilities (*атрибут calibre.ebooks.metadata.sources.base.Source*), 270
card_prefix() (*метод calibre.devices.interface.DevicePlugin*), 281
card_prefix() (*метод calibre.devices.usbms.device.Device*), 288
CatalogPlugin (*класс в calibre.customize*), 269
category (*атрибут calibre.customize.PreferencesPlugin*), 296
category_order (*атрибут calibre.customize.PreferencesPlugin*), 296
center_navbar (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 47
change_font() (*в модуле calibre.ebooks.oeb.polish.fonts*), 382
changed_signal (*атрибут calibre.gui2.preferences.ConfigWidgetBase*), 297
changed_signal (*атрибут calibre.gui2.preferences.ConfigWidgetInterface*), 296
clean_downloaded_metadata() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
cleanup() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 42
CLI (*класс в calibre.devices.usbms.cli*), 290
cli_main() (*метод calibre.customize.Plugin*), 266
cli_options (*атрибут calibre.customize.CatalogPlugin*), 269
clone_browser() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 42
close_editor() (*метод calibre.gui2.tweak_book.boss.Boss*), 384
commit() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
commit() (*метод calibre.gui2.preferences.ConfigWidgetBase*), 298
commit() (*метод calibre.gui2.preferences.ConfigWidgetInterface*), 297
commit_all_editors_to_container() (*метод calibre.gui2.tweak_book.boss.Boss*), 384
commit_item() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
common_options (*атрибут calibre.customize.conversion.InputFormatPlugin*), 275
common_options (*атрибут*

calibre.customize.conversion.OutputFormatPlugin), 276
compress_covers() (*метод calibre.db.cache.Cache*), 366
compress_news_images (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 47
compress_news_images_auto_size (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 47
compress_news_images_max_size (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 47
config_help_message (*атрибут calibre.ebooks.metadata.sources.base.Source*), 271
config_widget (*атрибут calibre.customize.PreferencesPlugin*), 296
config_widget() (*метод calibre.customize.Plugin*), 265
config_widget() (*метод calibre.ebooks.metadata.sources.base.Source*), 271
config_widget() (*метод класса calibre.devices.interface.DevicePlugin*), 283
ConfigWidgetBase (*класс в calibre.gui2.preferences*), 297
ConfigWidgetInterface (*класс в calibre.gui2.preferences*), 296
Container (*класс в calibre.ebooks.oeb.polish.container*), 375
contains(), **164**
conversion_options (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 47
convert() (*метод calibre.customize.conversion.InputFormatPlugin*), 275
convert() (*метод calibre.customize.conversion.OutputFormatPlugin*), 276
copy_cover_to() (*метод calibre.db.cache.Cache*), 366
copy_format_to() (*метод calibre.db.cache.Cache*), 366
core_usage (*атрибут calibre.customize.conversion.InputFormatPlugin*), 274
cover() (*метод calibre.db.cache.Cache*), 366
cover_margins (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
create_action() (*метод calibre.gui2.tweak_book.plugin.Tool*), 384
create_inline_toc() (*в модуле calibre.ebooks.oeb.polish.toc*), 382
create_menu_action() (*метод calibre.gui2.actions.InterfaceAction*), 293
create_widget() (*метод calibre.customize.PreferencesPlugin*), 296
CSS, **389**
current_container (*свойство calibre.gui2.tweak_book.plugin.Tool*), 383
currently_editing (*свойство calibre.gui2.tweak_book.boss.Boss*), 385
custom_field_keys() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
customization_help() (*метод calibre.customize.Plugin*), 266
customization_help() (*метод calibre.ebooks.metadata.sources.base.Source*), 271

D

data_for_find_identical_books() (*метод calibre.db.cache.Cache*), 367
data_for_has_book() (*метод calibre.db.cache.Cache*), 367
debug_managed_device_detection() (*метод calibre.devices.interface.DevicePlugin*), 279
deepcopy() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
default_cover() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 43
delay (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
delete_annotations() (*метод calibre.db.cache.Cache*), 367
delete_books() (*метод calibre.devices.interface.DevicePlugin*), 282
delete_books() (*метод calibre.devices.usbms.driver.USBMS*), 291
delete_custom_book_data() (*метод calibre.db.cache.Cache*), 367
delete_trash_entry() (*метод calibre.db.cache.Cache*), 367
description (*атрибут calibre.customize.Plugin*), 264
description (*атрибут calibre.customize.PreferencesPlugin*), 296
description (*атрибут calibre.devices.usbms.driver.USBMS*), 290
description (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
description (*свойство calibre.customize.conversion.OutputFormatPlugin*), 276
detect_managed_devices() (*метод calibre.devices.interface.DevicePlugin*), 279
Device (*класс в calibre.devices.usbms.device*), 287
DevicePlugin (*класс в calibre.devices.interface*), 277

dirty() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
do_user_config() (*метод calibre.customize.Plugin*), 265
dont_add_to (*атрибут calibre.gui2.actions.InterfaceAction*), 293
dont_remove_from (*атрибут calibre.gui2.actions.InterfaceAction*), 293
download() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 43
download_cover() (*метод calibre.ebooks.metadata.sources.base.Source*), 273
drop_event() (*метод calibre.gui2.actions.InterfaceAction*), 293

E

ebook-convert опция командной строки
--add-alt-text-to-img, 342
--asciiize, 338
--author-sort, 344
--authors, 344
--base-font-size, 338
--book-producer, 344
--change-justification, 338
--chapter, 342
--chapter-mark, 342
--comments, 344
--cover, 344
--debug-pipeline, 345
--disable-dehyphenate, 341
--disable-delete-blank-paragraphs, 341
--disable-fix-indents, 341
--disable-font-rescaling, 338
--disable-format-scene-breaks, 341
--disable-italicize-common-cases, 341
--disable-markup-chapter-headings, 341
--disable-remove-fake-margins, 342
--disable-renumber-headings, 341
--disable-unwrap-lines, 341
--dont-split-on-page-breaks, 337
--duplicate-links-in-toc, 343
--embed-all-fonts, 338
--embed-font-family, 339
--enable-heuristics, 341
--epub-flatten, 337
--epub-inline-toc, 337
--epub-max-image-size, 337
--epub-toc-at-end, 337
--epub-version, 337
--expand-css, 339
--extra-css, 339
--extract-to, 337
--filter-css, 339
--flow-size, 337
--font-size-mapping, 339
--from-opf, 344
--help, 336
--html-unwrap-factor, 341
--input-encoding, 337
--input-profile, 336
--insert-blank-line, 339
--insert-blank-line-size, 339
--insert-metadata, 342
--isbn, 344
--keep-ligatures, 339
--language, 344
--level1-toc, 343
--level2-toc, 343
--level3-toc, 343
--line-height, 339
--linearize-tables, 339
--list-recipes, 336
--margin-bottom, 339
--margin-left, 340
--margin-right, 340
--margin-top, 340
--max-toc-links, 343
--minimum-line-height, 340
--no-chapters-in-toc, 343
--no-default-epub-cover, 338
--no-svg-cover, 338
--output-profile, 336
--page-breaks-before, 342
--prefer-metadata-cover, 343
--preserve-cover-aspect-ratio, 338
--pretty-print, 338
--pubdate, 344
--publisher, 344
--rating, 344
--read-metadata-from-opf, 344
--remove-first-image, 343
--remove-paragraph-spacing, 340
--remove-paragraph-spacing-indent-size, 340
--replace-scene-breaks, 341
--search-replace, 341
--series, 344
--series-index, 344
--smarten-punctuation, 340
--sr1-replace, 342
--sr1-search, 342
--sr2-replace, 342
--sr2-search, 342
--sr3-replace, 342
--sr3-search, 342
--start-reading-at, 343
--subset-embedded-fonts, 340

--tags, 344
--timestamp, 344
--title, 344
--title-sort, 344
--toc-filter, 343
--toc-threshold, 343
--toc-title, 338
--transform-css-rules, 340
--transform-html-rules, 340
--unsmarten-punctuation, 340
--use-auto-toc, 343
--verbose, 345
--version, 337
-d, 345
-h, 336
-m, 344
-v, 345
ebook-edit опция командной строки
--detach, 345
--help, 345
--select-text, 345
--version, 345
-h, 345
ebook-meta опция командной строки
--author-sort, 346
--authors, 346
--book-producer, 346
--category, 346
--comments, 346
--cover, 346
--date, 346
--disallow-rendered-cover, 346
--from-opf, 346
--get-cover, 346
--help, 346
--identifier, 346
--index, 346
--isbn, 346
--language, 346
--lrf-bookid, 346
--publisher, 346
--rating, 347
--series, 347
--tags, 347
--title, 347
--title-sort, 347
--to-opf, 347
--version, 347
-a, 346
-c, 346
-d, 346
-h, 346
-i, 346
-k, 346
-l, 346
-p, 346
-r, 347
-s, 347
-t, 347
ebook-polish опция командной строки
-H, 347
-U, 348
--add-soft-hyphens, 347
--compress-images, 347
--cover, 347
--download-external-resources, 347
--embed-fonts, 348
--help, 348
--jacket, 348
--opf, 348
--remove-jacket, 348
--remove-soft-hyphens, 348
--remove-unused-css, 348
--smarten-punctuation, 348
--subset-fonts, 348
--upgrade-book, 348
--verbose, 348
--version, 348
-c, 347
-d, 347
-e, 348
-f, 348
-h, 348
-i, 347
-j, 348
-o, 348
-p, 348
-u, 348
ebook-viewer опция командной строки
--continue, 349
--detach, 349
--force-reload, 349
--full-screen, 349
--fullscreen, 349
--help, 349
--new-instance, 349
--open-at, 349
--raise-window, 349
--version, 349
-f, 349
-h, 349
edit_file() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
eject() (*метод calibre.devices.interface.DevicePlugin*), 280
eject() (*метод calibre.devices.usbms.device.Device*), 289

embed_metadata() (*метод calibre.db.cache.Cache*), 367
encoding (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
exists() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
expire_old_trash() (*метод calibre.db.cache.Cache*), 367
export_note() (*метод calibre.db.cache.Cache*), 367
extra_css (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
extract_readable_article() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 43

F

fast_field_for() (*метод calibre.db.cache.Cache*), 367
feeds (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
fetch-ebook-metadata опция командной строки
    -I, 350
    --allowed-plugin, 350
    --authors, 350
    --cover, 350
    --help, 350
    --identifier, 350
    --isbn, 350
    --opf, 350
    --timeout, 350
    --title, 350
    --verbose, 350
    --version, 350
    -a, 350
    -c, 350
    -d, 350
    -h, 350
    -i, 350
    -o, 350
    -p, 350
    -t, 350
    -v, 350
field_for() (*метод calibre.db.cache.Cache*), 367
field_ids_for() (*метод calibre.db.cache.Cache*), 367
field_supports_notes() (*метод calibre.db.cache.Cache*), 368
file_type (*атрибут calibre.customize.conversion.OutputFormatPlugin*), 276
file_types (*атрибут calibre.customize.CatalogPlugin*), 269
file_types (*атрибут calibre.customize.conversion.InputFormatPlugin*), 274
file_types (*атрибут calibre.customize.FileTypePlugin*), 266
file_types (*атрибут calibre.customize.MetadataReaderPlugin*), 268
file_types (*атрибут calibre.customize.MetadataWriterPlugin*), 269
filename_callback() (*метод calibre.devices.usbms.device.Device*), 289
filesize() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
FileTypePlugin (*класс в calibre.customize*), 266
filter_css() (*в модуле calibre.ebooks.oeb.polish.css*), 382
filter_regexps (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
find_identical_books() (*метод calibre.db.cache.Cache*), 368
fix_all_html() (*в модуле calibre.ebooks.oeb.polish.pretty*), 380
fix_html() (*в модуле calibre.ebooks.oeb.polish.pretty*), 380
for_viewer (*атрибут calibre.customize.conversion.InputFormatPlugin*), 275
format() (*метод calibre.db.cache.Cache*), 368
format_abspath() (*метод calibre.db.cache.Cache*), 368
format_added (*атрибут calibre.db.cache.Cache.EventType*), 364
format_field() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
format_hash() (*метод calibre.db.cache.Cache*), 368
format_metadata() (*метод calibre.db.cache.Cache*), 368
FORMATS (*атрибут calibre.devices.interface.DevicePlugin*), 277
FORMATS (*атрибут calibre.devices.usbms.driver.USBMS*), 290
formats() (*метод calibre.db.cache.Cache*), 368
formats_removed (*атрибут calibre.db.cache.Cache.EventType*), 364
free_space() (*метод calibre.devices.interface.DevicePlugin*), 281
free_space() (*метод calibre.devices.usbms.device.Device*), 288
from_files() (*в модуле calibre.ebooks.oeb.polish.toc*), 382
from_links() (*в модуле calibre.ebooks.oeb.polish.toc*), 382
from_xpaths() (*в модуле calibre.ebooks.oeb.polish.toc*),

382

G

generate_item() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
genesis() (*метод calibre.gui2.actions.InterfaceAction*), 294
genesis() (*метод calibre.gui2.preferences.ConfigWidgetInterface*), 297
get_all_items_that_have_notes() (*метод calibre.db.cache.Cache*), 369
get_all_link_maps_for_book() (*метод calibre.db.cache.Cache*), 369
get_all_standard_metadata() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
get_all_user_metadata() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
get_annotations() (*метод calibre.devices.usbms.device.Device*), 289
get_article_url() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 43
get_author_tokens() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
get_book_url() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
get_book_url_name() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
get_book_urls() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
get_browser() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 43
get_cached_cover_url() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
get_categories() (*метод calibre.db.cache.Cache*), 369
get_collections() (*метод calibre.devices.interface.BookList*), 286
get_cover_url() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 43
get_custom_book_data() (*метод calibre.db.cache.Cache*), 369
get_device_information() (*метод calibre.devices.interface.DevicePlugin*), 281
get_device_information() (*метод calibre.devices.usbms.driver.USBMS*), 290
get_device_uid() (*метод calibre.devices.interface.DevicePlugin*), 284
get_driveinfo() (*метод calibre.devices.interface.DevicePlugin*), 281
get_extra_css() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
get_feeds() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
get_file() (*метод calibre.devices.interface.DevicePlugin*), 283
get_file_path_for_processing() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
get_id_map() (*метод calibre.db.cache.Cache*), 369
get_identifiers() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
get_ids_for_custom_book_data() (*метод calibre.db.cache.Cache*), 369
get_images() (*метод calibre.customize.conversion.InputFormatPlugin*), 275
get_item_id() (*метод calibre.db.cache.Cache*), 369
get_item_ids() (*метод calibre.db.cache.Cache*), 369
get_item_name() (*метод calibre.db.cache.Cache*), 369
get_item_name_map() (*метод calibre.db.cache.Cache*), 369
get_link_map() (*метод calibre.db.cache.Cache*), 369
get_masthead_title() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
get_masthead_url() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
get_metadata() (*метод calibre.customize.MetadataReaderPlugin*), 269
get_metadata() (*метод calibre.db.cache.Cache*), 370
get_next_series_num_for() (*метод calibre.db.cache.Cache*), 370
get_notes_resource() (*метод calibre.db.cache.Cache*), 370
get_obfuscated_article() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
get_open_popup_message() (*метод класса calibre.devices.interface.DevicePlugin*), 279
get_option() (*метод calibre.devices.interface.DevicePlugin*), 284
get_proxy_metadata() (*метод calibre.db.cache.Cache*), 370

get_recommended_folders() (*в модуле calibre.ebooks.oeb.polish.replace*), 380
get_standard_metadata() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
get_title_tokens() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
get_url_specific_delay() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
get_usage_count_by_id() (*метод calibre.db.cache.Cache*), 370
get_user_blacklisted_devices() (*метод calibre.devices.interface.DevicePlugin*), 284
get_user_metadata() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
gui (*свойство calibre.gui2.tweak_book.plugin.Tool*), 383
gui_category (*атрибут calibre.customize.PreferencesPlugin*), 296
gui_configuration_widget() (*метод calibre.customize.conversion.InputFormatPlugin*), 276
gui_configuration_widget() (*метод calibre.customize.conversion.OutputFormatPlugin*), 277
gui_layout_complete() (*метод calibre.gui2.actions.InterfaceAction*), 295
gui_name (*атрибут calibre.customize.PreferencesPlugin*), 296
guide_type_map (*свойство calibre.ebooks.oeb.polish.container.Container*), 376

H

handle_gzip (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
has_book() (*метод calibre.db.cache.Cache*), 370
has_format() (*метод calibre.db.cache.Cache*), 370
has_html_comments (*атрибут calibre.ebooks.metadata.sources.base.Source*), 270
has_id() (*метод calibre.db.cache.Cache*), 370
has_name() (*метод calibre.ebooks.oeb.polish.container.Container*), 376
href_to_name() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
HTML, **389**

I

icon (*атрибут calibre.customize.PreferencesPlugin*), 296
icon (*атрибут calibre.devices.interface.DevicePlugin*), 278
id_from_url() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
identify() (*метод calibre.ebooks.metadata.sources.base.Source*), 273
identify_results_keygen() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
ignore_connected_device() (*метод calibre.devices.interface.DevicePlugin*), 284
ignore_duplicate_articles (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 48
ignore_ssl_errors (*атрибут calibre.ebooks.metadata.sources.base.Source*), 270
image_url_processor() (*метод класса calibre.web.feeds.news.BasicNewsRecipe*), 42
import_note() (*метод calibre.db.cache.Cache*), 370
index_to_soup() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
indexing_progress_changed (*атрибут calibre.db.cache.Cache.EventType*), 364
init() (*метод calibre.db.cache.Cache*), 370
initial_tab_changed() (*метод calibre.gui2.preferences.ConfigWidgetInterface*), 297
initialization_complete() (*метод calibre.gui2.actions.InterfaceAction*), 295
initialize() (*метод calibre.customize.CatalogPlugin*), 270
initialize() (*метод calibre.customize.Plugin*), 265
initialize() (*метод calibre.gui2.preferences.ConfigWidgetBase*), 298
initialize() (*метод calibre.gui2.preferences.ConfigWidgetInterface*), 297
InputFormatPlugin (*класс в calibre.customize.conversion*), 274
insert_into_xml() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
installation_type (*атрибут calibre.customize.Plugin*), 265
InterfaceAction (*класс в calibre.gui2.actions*), 292
InterfaceActionBase (*класс в calibre.customize*), 295
InternalMetadataCompareKeyGen (*класс в calibre.ebooks.metadata.sources.base*), 274
is_configured() (*метод*

calibre.ebooks.metadata.sources.base.Source), 271
is_dir (*атрибут calibre.ebooks.oeb.polish.container.Container*), 377
is_dynamically_controllable() (*метод calibre.devices.interface.DevicePlugin*), 284
is_image_collection (*атрибут calibre.customize.conversion.InputFormatPlugin*), 274
is_link_wanted() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 44
is_null() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
is_running() (*метод calibre.devices.interface.DevicePlugin*), 285
is_usb_connected() (*метод calibre.devices.interface.DevicePlugin*), 279
items_removed (*атрибут calibre.db.cache.Cache.EventType*), 364
items_renamed (*атрибут calibre.db.cache.Cache.EventType*), 364
items_with_notes_in_book() (*метод calibre.db.cache.Cache*), 370
iterlinks() (*метод calibre.ebooks.oeb.polish.container.Container*), 377

K

keep_only_tags (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49

L

language (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49
library_about_to_change() (*метод calibre.gui2.actions.InterfaceAction*), 294
library_changed() (*метод calibre.gui2.actions.InterfaceAction*), 295
link_for() (*метод calibre.db.cache.Cache*), 370
links_changed (*атрибут calibre.db.cache.Cache.EventType*), 364
list_extra_files() (*метод calibre.db.cache.Cache*), 371
load_actual_plugin() (*метод calibre.customize.InterfaceActionBase*), 295
load_resources() (*метод calibre.customize.Plugin*), 265
load_resources() (*метод calibre.gui2.actions.InterfaceAction*), 294
location_selected() (*метод calibre.gui2.actions.InterfaceAction*), 294
LRF, **389**
lrf2lrs опция командной строки
--dont-output-resources, 350
--help, 350
--output, 350
--verbose, 351
--version, 351
-h, 350
-o, 350
lrfviewer опция командной строки
--disable-hyphenation, 351
--help, 351
--profile, 351
--verbose, 351
--version, 351
--visual-debug, 351
--white-background, 351
-h, 351
lrs2lrf опция командной строки
--help, 351
--lrs, 351
--output, 352
--verbose, 352
--version, 352
-h, 351
-o, 352

M

make_name_unique() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
MANAGES_DEVICE_PRESENCE (*атрибут calibre.devices.interface.DevicePlugin*), 278
manifest_has_name() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
manifest_id_map (*свойство calibre.ebooks.oeb.polish.container.Container*), 377
manifest_items_of_type() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
manifest_items_with_property() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
manifest_type_map (*свойство calibre.ebooks.oeb.polish.container.Container*), 377
mark_as_cover() (*в модуле calibre.ebooks.oeb.polish.cover*), 381
mark_as_titlepage() (*в модуле calibre.ebooks.oeb.polish.cover*), 381
masthead_url (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49

match_regexps (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49
max_articles_per_feed (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49
MAX_PATH_LEN (*атрибут calibre.devices.usbms.device.Device*), 287
merge() (*в модуле calibre.ebooks.oeb.polish.split*), 381
merge_annotations_for_book() (*метод calibre.db.cache.Cache*), 371
merge_extra_files() (*метод calibre.db.cache.Cache*), 371
Metadata (*класс в calibre.ebooks.metadata.book.base*), 216
metadata_changed (*атрибут calibre.db.cache.Cache.EventType*), 364
metadata_for_field() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
MetadataReaderPlugin (*класс в calibre.customize*), 268
MetadataWriterPlugin (*класс в calibre.customize*), 269
mi (*свойство calibre.ebooks.oeb.polish.container.Container*), 377
minimum_calibre_version (*атрибут calibre.customize.Plugin*), 265
model_metadata() (*метод класса calibre.devices.interface.DevicePlugin*), 279
model_metadata() (*метод класса calibre.devices.usbms.device.Device*), 287
module
    calibre.customize, 264
    calibre.customize.conversion, 274
    calibre.db.cache, 363
    calibre.devices.interface, 277
    calibre.ebooks.metadata.book.base, 216
    calibre.ebooks.metadata.sources.base, 270
    calibre.ebooks.oeb.polish.container, 375
    calibre.ebooks.oeb.polish.cover, 381
    calibre.ebooks.oeb.polish.css, 382
    calibre.ebooks.oeb.polish.jacket, 380
    calibre.ebooks.oeb.polish.pretty, 380
    calibre.ebooks.oeb.polish.replace, 379
    calibre.ebooks.oeb.polish.split, 380
    calibre.ebooks.oeb.polish.toc, 382
    calibre.gui2.tweak_book.boss, 384
    calibre.utils.formatter_functions, 191
    calibre.web.feeds.news, 41
move_book_from_trash() (*метод calibre.db.cache.Cache*), 371
move_format_from_trash() (*метод calibre.db.cache.Cache*), 371
multisort() (*метод calibre.db.cache.Cache*), 371
multisplit() (*в модуле calibre.ebooks.oeb.polish.split*), 381

N

name (*атрибут calibre.customize.Plugin*), 264
name (*атрибут calibre.gui2.actions.InterfaceAction*), 292
name (*атрибут calibre.gui2.tweak_book.plugin.Tool*), 383
name(), **164**
name_order (*атрибут calibre.customize.PreferencesPlugin*), 296
name_to_abspath() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
name_to_href() (*метод calibre.ebooks.oeb.polish.container.Container*), 377
names_that_must_not_be_changed (*свойство calibre.ebooks.oeb.polish.container.Container*), 377
names_that_must_not_be_removed (*свойство calibre.ebooks.oeb.polish.container.Container*), 377
names_that_need_not_be_manifested (*свойство calibre.ebooks.oeb.polish.container.Container*), 378
needs_subscription (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49
NEWS_IN_FOLDER (*атрибут calibre.devices.usbms.device.Device*), 287
no_stylesheets (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49
normalize_path() (*метод класса calibre.devices.usbms.driver.USBMS*), 292
notes_changed (*атрибут calibre.db.cache.Cache.EventType*), 364
notes_data_for() (*метод calibre.db.cache.Cache*), 371
notes_for() (*метод calibre.db.cache.Cache*), 371
notes_resources_used_by() (*метод calibre.db.cache.Cache*), 371
NUKE_COMMENTS (*атрибут calibre.devices.interface.DevicePlugin*), 278

O

oldest_article (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 49
on_import (*атрибут calibre.customize.FileTypePlugin*), 266

on_postconvert (*атрибут calibre.customize.FileTypePlugin*), 267
on_postdelete (*атрибут calibre.customize.FileTypePlugin*), 267
on_postimport (*атрибут calibre.customize.FileTypePlugin*), 267
on_postprocess (*атрибут calibre.customize.FileTypePlugin*), 267
on_preprocess (*атрибут calibre.customize.FileTypePlugin*), 267
open() (*метод calibre.devices.interface.DevicePlugin*), 280
open() (*метод calibre.devices.usbms.device.Device*), 289
open() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
open_book() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
OPEN_FEEDBACK_MESSAGE (*атрибут calibre.devices.interface.DevicePlugin*), 278
opf (*свойство calibre.ebooks.oeb.polish.container.Container*), 378
opf_get_or_create() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
opf_version (*свойство calibre.ebooks.oeb.polish.container.Container*), 378
opf_version_parsed (*свойство calibre.ebooks.oeb.polish.container.Container*), 378
opf_xpath() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
options (*атрибут calibre.customize.conversion.InputFormatPlugin*), 275
options (*атрибут calibre.customize.conversion.OutputFormatPlugin*), 276
options (*атрибут calibre.ebooks.metadata.sources.base.Source*), 271
OSX_MAIN_MEM_VOL_PAT (*атрибут calibre.devices.usbms.device.Device*), 287
output_encoding (*атрибут calibre.customize.conversion.InputFormatPlugin*), 275
OutputFormatPlugin (*класс в calibre.customize.conversion*), 276

P

parse_feeds() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 45
parse_index() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 45
parsed() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
path_sep (*атрибут calibre.devices.interface.DevicePlugin*), 278
Plugin (*класс в calibre.customize*), 264
populate_article_metadata() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 45
popup_type (*атрибут calibre.gui2.actions.InterfaceAction*), 292
post_yank_cleanup() (*метод calibre.devices.interface.DevicePlugin*), 281
post_yank_cleanup() (*метод calibre.devices.usbms.device.Device*), 289
postadd() (*метод calibre.customize.FileTypePlugin*), 268
postconvert() (*метод calibre.customize.FileTypePlugin*), 267
postdelete() (*метод calibre.customize.FileTypePlugin*), 268
postimport() (*метод calibre.customize.FileTypePlugin*), 267
postprocess_book() (*метод calibre.customize.conversion.InputFormatPlugin*), 275
postprocess_book() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 45
postprocess_html() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 45
pref() (*метод calibre.db.cache.Cache*), 371
prefer_results_with_isbn (*атрибут calibre.ebooks.metadata.sources.base.Source*), 271
PreferencesPlugin (*класс в calibre.customize*), 295
prepare_addable_books() (*метод calibre.devices.interface.DevicePlugin*), 283
preprocess_html() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 46
preprocess_image() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 46
preprocess_raw_html() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 46
preprocess_regexps (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 50
pretty_all() (*в модуле calibre.ebooks.oeb.polish.pretty*), 380
pretty_css() (*в модуле calibre.ebooks.oeb.polish.pretty*), 380

pretty_html() (*в модуле calibre.ebooks.oeb.polish.pretty*), 380
pretty_xml() (*в модуле calibre.ebooks.oeb.polish.pretty*), 380
print_version() (*метод класса calibre.web.feeds.news.BasicNewsRecipe*), 42
priority (*атрибут calibre.customize.Plugin*), 265
priority (*атрибут calibre.gui2.actions.InterfaceAction*), 292
PRODUCT_ID (*атрибут calibre.devices.interface.DevicePlugin*), 277
PRODUCT_ID (*атрибут calibre.devices.usbms.device.Device*), 287
publication_date() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 46
publication_type (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 50

R

raw_data() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
re:test(), **164**
read_backup() (*метод calibre.db.cache.Cache*), 371
recipe_disabled (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 50
recipe_specific_options (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 50
recommendations (*атрибут calibre.customize.conversion.InputFormatPlugin*), 275
recommendations (*атрибут calibre.customize.conversion.OutputFormatPlugin*), 276
recursions (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 50
refresh_gui() (*метод calibre.gui2.preferences.ConfigWidgetInterface*), 297
regexp, **389**
register() (*метод calibre.gui2.preferences.ConfigWidgetBase*), 298
register_shortcut() (*метод calibre.gui2.tweak_book.plugin.Tool*), 383
relpath() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
remove_attributes (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_book() (*метод calibre.devices.interface.BookList*), 286
remove_books() (*метод calibre.db.cache.Cache*), 371
remove_books_from_metadata() (*метод calibre.devices.usbms.driver.USBMS*), 291
remove_books_from_metadata() (*метод класса calibre.devices.interface.DevicePlugin*), 282
remove_empty_feeds (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_extra_files() (*метод calibre.db.cache.Cache*), 372
remove_formats() (*метод calibre.db.cache.Cache*), 372
remove_from_spine() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
remove_from_xml() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
remove_item() (*метод calibre.ebooks.oeb.polish.container.Container*), 378
remove_items() (*метод calibre.db.cache.Cache*), 372
remove_jacket() (*в модуле calibre.ebooks.oeb.polish.jacket*), 380
remove_javascript (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
remove_stale_user_metadata() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
remove_tags (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
remove_tags_after (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
remove_tags_before (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
remove_unused_css() (*в модуле calibre.ebooks.oeb.polish.css*), 382
rename() (*метод calibre.ebooks.oeb.polish.container.Container*), 379
rename_extra_files() (*метод calibre.db.cache.Cache*), 372
rename_files() (*в модуле calibre.ebooks.oeb.polish.replace*), 380
rename_items() (*метод calibre.db.cache.Cache*), 372
replace() (*метод calibre.ebooks.oeb.polish.container.Container*), 379
replace_links() (*в модуле calibre.ebooks.oeb.polish.replace*), 379

replace_links() (*метод calibre.ebooks.oeb.polish.container.Container*), 379
requires_version (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
reset() (*метод calibre.devices.interface.DevicePlugin*), 279
reset() (*метод calibre.devices.usbms.device.Device*), 287
resolve_internal_links (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
restart_critical (*атрибут calibre.gui2.preferences.ConfigWidgetBase*), 298
restart_critical (*атрибут calibre.gui2.preferences.ConfigWidgetInterface*), 297
restore_book() (*метод calibre.db.cache.Cache*), 372
restore_defaults() (*метод calibre.gui2.preferences.ConfigWidgetBase*), 298
restore_defaults() (*метод calibre.gui2.preferences.ConfigWidgetInterface*), 297
restore_defaults_desc (*атрибут calibre.gui2.preferences.ConfigWidgetInterface*), 297
restore_original_format() (*метод calibre.db.cache.Cache*), 372
reverse_article_order (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
rewind_savepoint() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
RSS, **389**
run() (*метод calibre.customize.CatalogPlugin*), 270
run() (*метод calibre.customize.FileTypePlugin*), 267

S

safe_read_lock (*свойство calibre.db.cache.Cache*), 372
sanitize_callback() (*метод calibre.devices.usbms.device.Device*), 289
sanitize_path_components() (*метод calibre.devices.usbms.device.Device*), 289
save_book() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
save_original_format() (*метод calibre.db.cache.Cache*), 372
save_settings() (*метод calibre.customize.Plugin*), 265
save_settings() (*метод calibre.ebooks.metadata.sources.base.Source*), 271
save_settings() (*метод класса calibre.devices.interface.DevicePlugin*), 283
scale_news_images (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 51
scale_news_images_to_device (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
search() (*метод calibre.db.cache.Cache*), 373
search_annotations() (*метод calibre.db.cache.Cache*), 373
search_notes() (*метод calibre.db.cache.Cache*), 373
serialize_item() (*метод calibre.ebooks.oeb.polish.container.Container*), 379
set_all_user_metadata() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
set_annotations_for_book() (*метод calibre.db.cache.Cache*), 373
set_conversion_options() (*метод calibre.db.cache.Cache*), 373
set_cover() (*в модуле calibre.ebooks.oeb.polish.cover*), 381
set_cover() (*метод calibre.db.cache.Cache*), 373
set_driveinfo_name() (*метод calibre.devices.interface.DevicePlugin*), 283
set_driveinfo_name() (*метод calibre.devices.usbms.driver.USBMS*), 290
set_field() (*метод calibre.db.cache.Cache*), 373
set_identifier() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
set_identifiers() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
set_library_info() (*метод calibre.devices.interface.DevicePlugin*), 284
set_link_map() (*метод calibre.db.cache.Cache*), 373
set_metadata() (*метод calibre.customize.MetadataWriterPlugin*), 269
set_metadata() (*метод calibre.db.cache.Cache*), 374
set_modified() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
set_notes_for() (*метод calibre.db.cache.Cache*), 374
set_option() (*метод calibre.devices.interface.DevicePlugin*), 285
set_plugboards() (*метод calibre.devices.interface.DevicePlugin*), 283
set_pref() (*метод calibre.db.cache.Cache*), 374
set_progress_reporter() (*метод*

calibre.devices.interface.DevicePlugin), 281
set_progress_reporter() (*метод calibre.devices.usbms.device.Device*), 288
set_spine() (*метод calibre.ebooks.oeb.polish.container.Container*), 379
set_user_blacklisted_devices() (*метод calibre.devices.interface.DevicePlugin*), 284
set_user_metadata() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
settings() (*метод класса calibre.devices.interface.DevicePlugin*), 283
show_current_diff() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
show_editor() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
shutdown() (*метод calibre.devices.interface.DevicePlugin*), 283
shutting_down() (*метод calibre.gui2.actions.InterfaceAction*), 295
simultaneous_downloads (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
skip_ad_pages() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 46
SLOW_DRIVEINFO (*атрибут calibre.devices.interface.DevicePlugin*), 278
smart_update() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
sort_index_by() (*метод calibre.web.feeds.news.BasicNewsRecipe*), 46
Source (*класс в calibre.ebooks.metadata.sources.base*), 270
specialize() (*метод calibre.customize.conversion.InputFormatPlugin*), 275
specialize_css_for_output() (*метод calibre.customize.conversion.OutputFormatPlugin*), 277
specialize_global_preferences() (*метод calibre.devices.interface.DevicePlugin*), 284
specialize_options() (*метод calibre.customize.conversion.OutputFormatPlugin*), 277
spine_items (*свойство calibre.ebooks.oeb.polish.container.Container*), 379
spine_iter (*свойство calibre.ebooks.oeb.polish.container.Container*), 379
spine_names (*свойство calibre.ebooks.oeb.polish.container.Container*), 379
split() (*в модуле calibre.ebooks.oeb.polish.split*), 380
split_if_is_multiple_composite() (*метод calibre.db.cache.Cache*), 374
split_jobs() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
standard_field_keys() (*метод calibre.ebooks.metadata.book.base.Metadata*), 217
STANDARD_METADATA_FIELDS (*в модуле calibre.ebooks.metadata.book.base*), 218
start_plugin() (*метод calibre.devices.interface.DevicePlugin*), 284
startup() (*метод calibre.devices.interface.DevicePlugin*), 283
stop_plugin() (*метод calibre.devices.interface.DevicePlugin*), 284
summary_length (*атрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
supported_platforms (*атрибут calibre.customize.conversion.InputFormatPlugin*), 274
supported_platforms (*атрибут calibre.customize.conversion.OutputFormatPlugin*), 276
supported_platforms (*атрибут calibre.customize.InterfaceActionBase*), 295
supported_platforms (*атрибут calibre.customize.MetadataReaderPlugin*), 268
supported_platforms (*атрибут calibre.customize.MetadataWriterPlugin*), 269
supported_platforms (*атрибут calibre.customize.Plugin*), 264
supported_platforms (*атрибут calibre.customize.PreferencesPlugin*), 296
supported_platforms (*атрибут calibre.devices.usbms.driver.USBMS*), 290
supported_platforms (*атрибут calibre.ebooks.metadata.sources.base.Source*), 270
supports_collections() (*метод calibre.devices.interface.BookList*), 286
supports_gzip_transfer_encoding (*атрибут calibre.ebooks.metadata.sources.base.Source*), 270
supports_restoring_to_defaults (*атрибут calibre.gui2.preferences.ConfigWidgetBase*), 297
supports_restoring_to_defaults (*атрибут*

calibre.gui2.preferences.ConfigWidgetInterface), 297
sync_booklists() (*метод calibre.devices.interface.DevicePlugin*), 282
sync_booklists() (*метод calibre.devices.usbms.driver.USBMS*), 291
sync_preview_to_editor() (*метод calibre.gui2.tweak_book.boss.Boss*), 385
synchronize_with_db() (*метод calibre.devices.interface.DevicePlugin*), 285

T

tag_browser_context_action() (*метод calibre.gui2.actions.InterfaceAction*), 295
tag_to_string() (*метод класса calibre.web.feeds.news.BasicNewsRecipe*), 42
tags_older_than() (*метод calibre.db.cache.Cache*), 374
template_css (*амрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
template_to_attribute() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
temporary_file() (*метод calibre.customize.Plugin*), 266
test_fields() (*метод calibre.ebooks.metadata.sources.base.Source*), 272
THUMBNAIL_COMPRESSION_QUALITY (*амрибут calibre.devices.interface.DevicePlugin*), 278
THUMBNAIL_HEIGHT (*амрибут calibre.devices.interface.DevicePlugin*), 278
timefmt (*амрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
timeout (*амрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
title (*амрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
to_html() (*метод calibre.ebooks.metadata.book.base.Metadata*), 218
Tool (*класс в calibre.gui2.tweak_book.plugin*), 383
toolbar_button_popup_mode (*амрибут calibre.gui2.tweak_book.plugin.Tool*), 383
total_space() (*метод calibre.devices.interface.DevicePlugin*), 281
total_space() (*метод calibre.devices.usbms.device.Device*), 288
touched_fields (*амрибут calibre.ebooks.metadata.sources.base.Source*), 270
type (*амрибут calibre.customize.CatalogPlugin*), 269
type (*амрибут calibre.customize.conversion.InputFormatPlugin*), 274
type (*амрибут calibre.customize.conversion.OutputFormatPlugin*), 276
type (*амрибут calibre.customize.FileTypePlugin*), 267
type (*амрибут calibre.customize.InterfaceActionBase*), 295
type (*амрибут calibre.customize.MetadataReaderPlugin*), 268
type (*амрибут calibre.customize.MetadataWriterPlugin*), 269
type (*амрибут calibre.customize.Plugin*), 265
type (*амрибут calibre.customize.PreferencesPlugin*), 296
type (*амрибут calibre.devices.interface.DevicePlugin*), 277
type (*амрибут calibre.ebooks.metadata.sources.base.Source*), 270

U

unretire_note_for() (*метод calibre.db.cache.Cache*), 374
update_annotations() (*метод calibre.db.cache.Cache*), 374
upload_books() (*метод calibre.devices.interface.DevicePlugin*), 282
upload_books() (*метод calibre.devices.usbms.driver.USBMS*), 290
upload_cover() (*метод calibre.devices.usbms.driver.USBMS*), 291
URL, **389**
USBMS (*класс в calibre.devices.usbms.driver*), 290
use_embedded_content (*амрибут calibre.web.feeds.news.BasicNewsRecipe*), 52
user_categories_for_books() (*метод calibre.db.cache.Cache*), 374
user_feedback_after_callback (*амрибут calibre.devices.interface.DevicePlugin*), 279
UserAnnotation (*амрибут calibre.devices.interface.DevicePlugin*), 278

V

VENDOR_ID (*амрибут calibre.devices.interface.DevicePlugin*), 277
VENDOR_ID (*амрибут calibre.devices.usbms.device.Device*), 287
version (*амрибут calibre.customize.MetadataReaderPlugin*), 268
version (*амрибут calibre.customize.MetadataWriterPlugin*), 269
version (*амрибут calibre.customize.Plugin*), 264
VIRTUAL_BOOK_EXTENSION_MESSAGE (*амрибут calibre.devices.interface.DevicePlugin*), 278

VIRTUAL_BOOK_EXTENSIONS (*атрибут calibre.devices.interface.DevicePlugin*), 278

W

WANTS_UPDATED_THUMBNAILS (*атрибут calibre.devices.interface.DevicePlugin*), 278
web2disk опция командной строки
    --base-dir, 352
    --delay, 352
    --dont-download-stylesheets, 352
    --encoding, 352
    --filter-regexp, 352
    --help, 352
    --match-regexp, 352
    --max-files, 352
    --max-recursions, 352
    --timeout, 352
    --verbose, 353
    --version, 353
    -d, 352
    -h, 352
    -n, 352
    -r, 352
    -t, 352
WINDOWS_CARD_A_MEM (*атрибут calibre.devices.usbms.device.Device*), 287
WINDOWS_CARD_B_MEM (*атрибут calibre.devices.usbms.device.Device*), 287
WINDOWS_MAIN_MEM (*атрибут calibre.devices.usbms.device.Device*), 287
windows_sort_drives() (*метод calibre.devices.usbms.device.Device*), 288

опция командной строки
    --help, 323
    --library-path, 323
    --password, 324
    --timeout, 324
    --username, 324
    --version, 324
    --with-library, 323
    -h, 323

рецепт, **389**