Manual de utilizador do Calibre

Versão 6.14.1

Kovid Goyal

março 16, 2023

Conteúdo

1 Interface gráfico de utilizador 3

2 Adicionar o seu sítio de notícias favorito 29

3 Visualizador de livros digitais 49

4 Conversão de livros digitais 55

5 Editar livros digitais 75

6 O servidor de conteúdos Calibre 109

7 Comparar livros digitais 117

8 A editar metadados do livro 121

9 Questões frequentes 125

10 Tutoriais 149

11 O esquema de URL calibre:// 245

12 Personalizar Calibre 249

13 Interface de linha de comandos 295

14 Criar um ambiente de desenvolvimento para o Calibre 337

15 Gestão de direitos digitais (DRM em inglês) 365

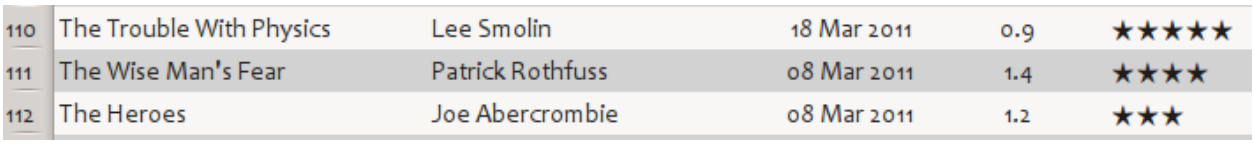
16 Glossário 369

Índice de Módulos do Python 371

Índice 373

O Calibre é um gestor de bibliotecas de livros digitais. Permite visualizar, converter e catalogar livros na maioria dos formatos. Permite ainda comunicar com a maioria dos dispositivos de leitura de livros digitais. Com acesso à Internet pode ainda procurar metadados para os seus livros bem como transferir artigos de jornais e convertê-los em livros digitais para uma leitura mais fácil. O Calibre pode ser executado em diversos sistema operativos como Linux, Windows ou macOS.

Acabou de iniciar o Calibre. O que fazer agora? Antes de poder fazer qualquer coisa com os seus livros digitais, o Calibre terá de os ficar a conhecer. Arraste e solte alguns ficheiros de livro para dentro do Calibre, ou clique no botão «Adicionar livros» e procure os livros com os quais você deseja trabalhar. Após adicionar livros estes irão aparecer na janela principal, que será mais ou menos como esta:



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	★★★

Agora que já admirou a lista de livros que acabou de adicionar aos seus favoritos, provavelmente irá querer ler um deles. Para o fazer, terá de converter o livro para um formato compatível com o seu leitor. Quando executar o Calibre pela primeira vez, o *Assistente de boas-vindas* será iniciado e irá configurar o Calibre para o seu dispositivo de leitura. A conversão é muito fácil. Basta selecionar o livro que deseja converter e clicar no botão «Converter livros». Poderá ignorar todas as opções, por agora, e clicar «OK». O pequeno ícone no canto inferior direito irá começar a girar. Quando terminar de girar, o seu livro convertido estará pronto. Clique no botão «Visualizar» para ler o livro.

Se pretender ler o livro no seu dispositivo, ligue-o ao computador, espere até o Calibre o detetar (10-20 segundos) e depois clique no botão «Enviar para o dispositivo». Assim que o ícone volte a parar de girar, desligue o seu dispositivo e está pronto a ler o livro! Se não tiver convertido o livro no passo anterior, o Calibre vai automaticamente convertê-lo para o formato que o seu dispositivo de leitura compreenda.

Para poder começar a dar um uso mais avançado, você deve ler acerca do tema *Interface gráfico de utilizador* (página 3). Para mais potência e versatilidade, visite generated/PT/cli-index. Você vai encontrar a lista de *Questões frequentes* (página 125) bem útil.

Se tiver mais perguntas, quiser discutir sobre o Calibre com outros utilizadores ou pedir ajuda sobre questões específicas, estão disponíveis fóruns e outros recursos de ajuda[1].

Secções

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

CAPÍTULO 1

Interface gráfico de utilizador

A Interface Gráfica do Usuário fornece acesso a todos os recursos de gerenciamento de bibliotecas e de conversão de formatos de e-books. O fluxo de trabalho básico para o uso do calibre é primeiro adicionar livros à biblioteca a partir do seu disco rígido. O calibre tentará automaticamente ler os metadados dos livros e adicioná-los ao seu banco de dados interno. Uma vez no banco de dados, você pode executar várias ações: ref: *ações*, que incluem conversão de um formato para outro, transferência para o dispositivo de leitura, visualização no computador e edição de metadados. O último inclui a modificação da capa, descrição e tags, entre outros detalhes. Observe que o calibre cria cópias dos arquivos adicionados a ele. Seus arquivos originais permanecem intocados.

A interface está dividida em várias secções:

- *Ações* (página 4)
- *Preferências* (página 11)
- *Catálogos* (página 11)
- *Pesquisar e ordenar* (página 12)
- *A interface de pesquisa* (página 12)
- *A guardar pesquisas* (página 18)
- *A pesquisar texto integral em todos os livros* (página 18)
- *Bibliotecas virtuais* (página 19)
- *Temporarily marking books* (página 19)
- *Calcular metadados a partir do nome dos ficheiros* (página 19)
- *Detalhes do livro* (página 20)
- *Navegador de conteúdos* (página 22)
- *Grelha de capas* (página 24)
- *Navegador de capas* (página 25)

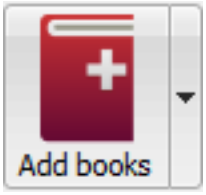
- *Visualização Rápida* (página 25)
- *Processos* (página 26)
- *Atalhos de teclado* (página 26)

1.1 Ações

A barra de ferramentas ações fornece atalhos convenientes para as ações mais usadas. Se você clicar com o botão direito do mouse nos botões, poderá executar variações na ação padrão. Observe que a barra de ferramentas ações terá uma aparência um pouco diferente, dependendo se você possui um leitor de e-book conectado ao seu computador.

- *Adicionar Livros* (página 4)
- *Editar metadados* (página 5)
- *Converter livros* (página 6)
- *Ver* (página 6)
- *Enviar para o dispositivo* (página 6)
- *Obter notícias* (página 7)
- *Biblioteca* (página 8)
- *Dispositivo* (página 8)
- *Exportar para disco* (página 9)
- *Ligar/Partilhar* (página 9)
- *Remover livros* (página 10)

1.1.1 Adicionar Livros



A ação *Add books* possui sete variações acessíveis através do clique com o botão direito do mouse.

1. **Add books from a single folder**: Opens a file chooser dialog and allows you to specify which books in a folder should be added. This action is *context sensitive*, i.e. it depends on which *catalog* (página 11) you have selected. If you have selected the *Library*, books will be added to the library. If you have selected the e-book reader device, the books will be uploaded to the device, and so on.
2. **Adicionar livros de pastas e subpastas**: permite-lhe escolher uma pasta. A pasta e todas as suas subpastas serão analisados recursivamente e qualquer livro eletrónico encontrado será adicionado à biblioteca. Você pode escolher se o Calibre deve adicionar cada ficheiro presente numa pasta como um registo único de livro ou como vários registos de livros. O Calibre assume que cada pasta contém um único livro: todos os ficheiros de ebook numa

mesma pasta são considerados como sendo o mesmo livro mas em diferentes formatos. Esta ação é o inverso da ação *Exportar para disco* (página 9), ou seja, pode usar *Exportar para disco*, eliminar os livros do Calibre e adicioná-los novamente em modo Livro único por pasta, sem perder informações, exceto a data (isto pressupõe que não tenha alterado nenhuma das configurações da ação Exportar para disco).

3. **Adicionar vários livros a partir de arquivo (ZIP/RAR)**: Permite adicionar vários ebooks que estejam armazenados em arquivos ZIP ou RAR selecionados. É um atalho conveniente que evita a necessidade de ter de primeiro descomprimir o arquivo e só depois adicionar os livros através de uma das duas opções acima.
4. **Add empty book (Book Entry with no formats)**: Allows you to create a blank book record. This can be used to then manually fill out the information about a book that you may not have yet in your collection.
5. **Adicionar de ISBN**: Permite que você adicione um ou mais livros através de seus códigos ISBN.
6. **Adicionar arquivos para um registro de livro selecionado**: Permite a você adicionar ou atualizar os arquivos associados a um livro existente em sua biblioteca.
7. **Adicionar um ficheiro vazio aos registos dos livros selecionados**: Permite-lhe adicionar um ficheiro vazio, do formato especificado, aos registos dos livros selecionados

The *Add books* action can read metadata from a wide variety of e-book formats. In addition, it tries to guess metadata from the filename. See the *Calcular metadados a partir do nome dos ficheiros* (página 19) section, to learn how to configure this.

Para adicionar um formato adicional para um livro existente você pode fazer uma das três coisas:

1. Drag and drop the file onto the Book details panel on the right side of the main window
2. Right click the *Add books* button and choose *Add files to selected books*.
3. Click the *Add books* button in the top right area of the *Edit metadata* dialog, accessed by the *Editar metadados* (página 5) action.

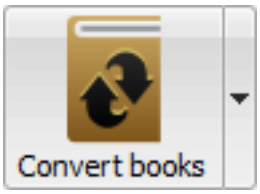
1.1.2 Editar metadados

O :guilabel: ` ação Editar Metadados` tem quatro variações que podem ser acessados por fazer um botão direito do mouse sobre o botão.

1. Permite editar os metadados dos livros, um por um, com a opção de buscar metadados, incluindo capas, da Internet. Também permite adicionar ou remover formatos de e-book específicos de um livro
2. Permite editar campos de metadados comuns para um grande número de livros simultaneamente. Ele opera em todos os livros que você selecionou: ref:`Library view
3. **Baixar metadados e capas**. Baixar metadados e capas (se disponível) para os livros são selecionados na lista de livros.
4. **Registros de livros direta**: Dá-lhe a capacidade de mesclar os metadados e formatos de dois ou mais registros de livros. Você pode optar por excluir ou manter os registros que não foram clicados em primeiro lugar.

For more details, see *A editar metadados do livro* (página 121).

1.1.3 Converter livros



E-books can be converted from a number of formats into whatever format your e-book reader prefers. Many e-books available for purchase will be protected by *Digital Rights Management* (página 365) *(DRM)* technology. calibre will not convert these e-books. It is easy to remove the DRM from many formats, but as this may be illegal, you will have to find tools to liberate your books yourself and then use calibre to convert them.

Para a maioria das pessoas, a conversão deve ser um simples caso de um clique. Se você quer aprender mais sobre o processo de conversão, veja: ref: *conversion*.

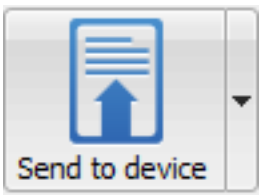
O *Converta ação books* tem três variações , acessado por fazer um botão direito do mouse sobre o botão.

1. Permite especificar opções de conversão para personalizar a conversão de cada livro eletrônico selecionado.
2. Permite especificar opções apenas uma vez para converter vários livros eletrônicos em massa.
3. **Create a catalog of the books in your calibre library**: Allows you to generate a complete listing of the books in your library, including all metadata, in several formats such as XML, CSV, BiBTeX, EPUB and MOBI. The catalog will contain all the books currently showing in the library view. This allows you to use the search features to limit the books to be catalogued. In addition, if you select multiple books using the mouse, only those books will be added to the catalog. If you generate the catalog in an e-book format such as EPUB, MOBI or AZW3, the next time you connect your e-book reader the catalog will be automatically sent to the device. For more information on how catalogs work, read the *Criar catálogos de AZW3 • EPUB • MOBI* (página 237).

1.1.4 Ver

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

1.1.5 Enviar para o dispositivo



A :guilabel: ` Enviar para Dispositivo` ação tem oito variações, acessado por fazer um botão direito do mouse sobre o botão.

1. ** Enviar para a memória principal **: os livros selecionados são transferidos para a memória principal do leitor de e-book.

2. ** Enviar para o cartão (A) **: Os livros selecionados são transferidos para o cartão de armazenamento (A) no leitor de e-book.
3. ** Enviar para o cartão (B) **: Os livros selecionados são transferidos para o cartão de armazenamento (B) no leitor de e-book.
4. **Envie no formato específico para**: Os livros selecionados serem transferidos para o local de armazenamento selecionado no dispositivo no formato que você especificar.
5. **Ejetar dispositivo**: Desconecta o dispositivo do calibre.
6. ** Setar padrão para enviar a ação do dispositivo**: Permite especificar qual das opções de 1 a 5 acima ou 7 abaixo, será a ação padrão quando você clica no botão principal.
7. **Enviar e excluir da biblioteca**: Os livros selecionados são transferidos para o local de armazenamento selecionado no dispositivo e, em seguida, **excluído** a partir da Biblioteca.
8. ** Buscar anotações (experimental) **: transfere anotações que você pode ter feito em um e-book no seu dispositivo para os metadados de comentários do livro na biblioteca de calibre.

Você pode controlar o nome do arquivo e a estrutura de pastas dos arquivos enviados ao dispositivo, configurando um modelo em *Preferences → Import/export → Sending books to devices* Veja também: ref: *templatelangcalibre*.

1.1.6 Obter notícias

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

A ação *Fetch news* usa receitas simples (10-15 linhas de código) para cada site de notícia. Para aprender como criar receitas para suas próprias fontes de notícias, veja *Adicionar o seu sítio de notícias favorito* (página 29).

A ação *Fetch news* tem três variações, acessadas através do clique com o botão direito do mouse sobre o botão.

1. **Schedule news download**: Allows you to schedule the download of your selected news sources from a list of hundreds available. Scheduling can be set individually for each news source you select and the scheduling is flexible allowing you to select specific days of the week or a frequency of days between downloads.
2. **Add a custom news source**: Allows you to create a simple recipe for downloading news from a custom news site that you wish to access. Creating the recipe can be as simple as specifying an RSS news feed URL, or you can be more prescriptive by creating Python-based code for the task. For more information, see *Adicionar o seu sítio de notícias favorito* (página 29).
3. **Fazer download de todas as fontes de notícias agendadas**: Faz o calibre iniciar imediatamente o download de todas as fontes de notícias que você selecionou.

1.1.7 Biblioteca

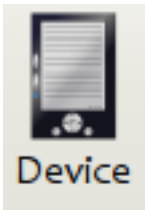
A ação *Library* permite você criar, trocar, renomear e remover uma biblioteca. O calibre permite que você crie quantas bibliotecas desejar. Você pode, por exemplo, criar uma biblioteca de ficção, uma de não ficção, uma de linguagem estrangeira, uma de projetos, ou qualquer outra estrutura que atenda suas necessidades. Bibliotecas são as estruturas organizacionais mais altas dentro do calibre. Cada biblioteca tem seu próprio conjunto de livros, etiquetas, categorias e localização de armazenamento.

1. **Trocar/criar biblioteca…*: Permite você; a) conectar a uma biblioteca do calibre já existente em outra localização, b) criar uma biblioteca vazia em uma nova localização ou, c) mover a biblioteca atual para uma nova localização.
2. **Troca rápida**: permite-lhe alternar entre bibliotecas que tenham sido registadas ou criadas no Calibre.
3. **Renomear biblioteca**: Permite mudar o nome da biblioteca.
4. **Escolher livro aleatoriamente**: Escolhe um livro para si, ao acaso, da biblioteca
5. Permite cancelar o registro de uma biblioteca do calibre.
6. Permite exportar dados de calibre para migração para um novo computador ou importar dados exportados anteriormente.
7. **<library name>**: Actions 7, 8 etc… give you immediate switch access between multiple libraries that you have created or attached to. This list contains only the 5 most frequently used libraries. For the complete list, use the Quick Switch menu.
8. **Manutenção de biblioteca**: Permite você verificar os problemas de consistência de dados da biblioteca atual e restaurar a base de dados da biblioteca atual a partir de backups.

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

Você pode copiar ou mover livros entre diferentes bibliotecas (caso você tenha mais de uma biblioteca configurada) clicando com o botão direito sobre o livro e selecionando a ação *Copy to library*.

1.1.8 Dispositivo



A ação *Device* permite você ver os livros na memória principal ou cartão de memória do seu dispositivo, ou remover o dispositivo (retirá-lo do calibre). Esse ícone aparecerá automaticamente na barra de ferramentas principal do calibre quando você conectar um dispositivo que permita essa operação. Você pode clicar nele para ver os livros do seu dispositivo. Você também pode arrastar e soltar livros da sua biblioteca do calibre para o ícone para transferí-los para

o seu dispositivo. Inversamente, você pode arrastar e soltar livros de seus dispositivos para o ícone de biblioteca da sua barra de ferramentas para transferir livros de seu dispositivo para a biblioteca do calibre.

1.1.9 Exportar para disco

A ação *Salvar no disco* tem 5 variações, acessadas através do clique com o botão direito do mouse sobre o botão.

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

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

Você pode controlar o nome do arquivo e a estrutura de pastas dos arquivos salvos no disco, configurando um modelo em: guilabel: *Preferências-> Importar / Exportar-> Salvar livros no disco* Also see *A linguagem de modelos do Calibre* (página 157).

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

Para 1. e 2., todos os formatos disponíveis, assim como seus metadatos, são armazenados no disco para cada livro selecionado. Metadados são armazenados em um arquivo OPF. Livros salvos podem ser re-importados para a biblioteca sem nenhuma perda de informação através da ação *Adicionar livros* (página 4).

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

1.1.10 Ligar/Partilhar

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

A ação: guilabel: *Connect / share* possui quatro variações, acessadas com o botão direito do mouse no botão.

1. **Conectar à pasta**: Permite você conectar à qualquer pasta do seu computador como se fosse um dispositivo e usar todas as funcionalidades que o calibre tem para dispositivos com essa pasta. Útil se seu dispositivo não é suportado pelo calibre mas está disponível como um disco USB.

2. **Start Content server**: Starts calibre's built-in web server. When started, your calibre library will be accessible via a web browser from the Internet (if you choose). You can configure how the web server is accessed by setting preferences at *Preferences → Sharing → Sharing over the net*
3. **Configuração de email baseado na partilha de livros**: Permite o compartilhamento de livros e feeds de notícias por e-mail. Depois de configurar os endereços de email para essa opção, calibre irá enviar atualizações de notícias e atualizações de livros para os endereços informados. Você pode configurar como calibre envia e-mail definindo as preferências em: guilabel:*Preferências-> Compartilhamento-> livros Compartilhamento por email*. Uma vez que você configurar um ou mais endereços de e-mail, este item de menu será substituído por entradas de menu para enviar livros para os endereços de email configuradas.

1.1.11 Remover livros

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

1. ** Remover livros selecionados **: Permite que você **permanentemente** remova todos os livros que são selecionados na lista de livros.
2. **Remover ficheiros de um formato específico dos livros selecionados…**: Permite-lhe eliminar **permanentemente** ficheiros de ebook, com o formato especificado, dos livros selecionados na lista de livros
3. **Remove all formats from selected books, except…**: Allows you to **permanently** remove e-book files of any format except a specified format from books that are selected in the book list.
4. ** Remover todos os formatos dos livros selecionados **: permite ** permanentemente ** remover todos os arquivos de livros eletrônicos dos livros selecionados na lista de livros. Somente os metadados permanecerão.
5. **Remover capas dos livros selecionados**: Permite você remover **permanentemente** arquivos de capas de livros selecionados na lista de livros.
6. ** Remover livros correspondentes do dispositivo **: permite remover arquivos de livros eletrônicos de um dispositivo conectado que corresponda aos livros selecionados na lista de livros.

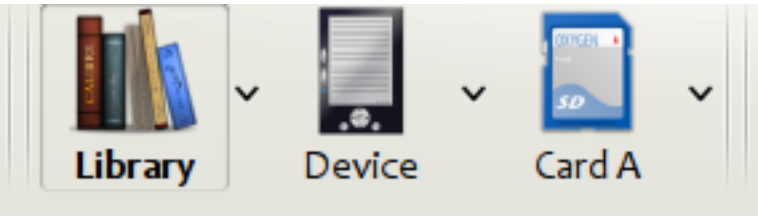
Nota: Note that when you use *Remove books* to delete books from your calibre library, the book record is permanently deleted, but the files are placed into the *Recycle Bin/Trash*. This allows you to recover the files if you change your mind.

1.2 Preferências

A ação *Preferences* permite você mudar a forma que o calibre funciona. Existem quatro variações, acessados clicando com o botão direito do mouse no botão.

1. **Preferências**: Permite você mudar a forma que o calibre funciona. Clicando no botão também executa essa ação.
2. ** Executar assistente de boas-vindas **: Permite iniciar o: guilabel: *Assistente de boas-vindas*, que apareceu na primeira vez em que você iniciou o calibre.
3. **Obtenha plugins para melhorar o calibre**: Abre uma nova janela que mostra plugins para o calibre. Esses plugins são desenvolvidos por terceiros para estender a funcionalidade do calibre.
4. **Reiniciar modo de depuração**: Permite ativar um modo de depuração que pode ajudar os desenvolvedores do calibre na resolução de problemas que você encontra com o programa. Para a maioria dos usuários, este deve permanecer desativado a menos que instruído por um desenvolvedor para habilitá-lo.

1.3 Catálogos



Um *catálogo* é uma coleção de livros. O Calibre aceita dois tipos de catálogos:

1. **Biblioteca**: É uma coleção de livros armazenada na biblioteca do Calibre no seu computador.
2. **Device**: This is a collection of books stored in your e-book reader. It will be available when you connect the reader to your computer.

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

1.4 Pesquisar e ordenar

A seção Pesquisar & Ordenar permite que você execute várias ações poderosas em suas coleções de livros.

- Você pode ordená-los por título, autor, data, classificação, etc., clicando nos títulos das colunas. Você também pode sub-tipo, ou seja, tipo em várias colunas. Por exemplo, se você clicar na coluna de título e, em seguida, a coluna de autor, o livro será classificada pelo autor e, em seguida, todas as entradas para o mesmo autor será classificado pelo título.
- You can search for a particular book or set of books using the Search bar. More on that below.
- You can quickly and conveniently edit metadata by selecting the entry you want changed in the list and pressing the `E` key.
- You can perform *Ações* (página 4) on sets of books. To select multiple books you can either:
  - Mantenha `Ctrl` pressionado e clique nos livros que deseja selecionar.
  - Mantenha a : kbd : Tecla *Shift* pressionada e clique no início e fim do livro de uma série de livros que você deseja selecionar .
- Você pode configurar quais campos quer mostrar usando a caixa de diálogo :ref:"configuration"
- To perform complex multiple column based sub-sorting add the *Sort by* tool to a toolbar via *Preferences → Toolbars & menus*.

1.5 A interface de pesquisa

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

```
Asimov Foundation format:lrf
```

Isso irá corresponder a todos os livros em sua biblioteca que têm «Asimov» e «Fundação» em seus metadados e estão disponíveis no formato LRF. Mais alguns exemplos

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

Search kinds

There are four search kinds: *contains*, *equality*, *regular expression* (see regular expressions[2]), and *character variant*. You choose the search kind with a prefix character.

"Contains" searches

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

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

then these searches find:

- `title:"er"` matches both ("e" matches both "é" and "e").
- `title:"g "` matches both because spaces are ignored.
- `title:"g,"` matches both because the comma is ignored.
- `title:"gb"` matches both because ", " is ignored in book 1 and spaces are ignored in book 2.
- `title:"g b"` matches both because comma and space are ignored.
- `title:"db"` matches #1 because the space in "and Bad" is ignored.
- `title:","` matches both (it actually matches all books) because commas are ignored.

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

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

"Equality" searches

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

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

- `tags:"=.History"` : True. `History` is a prefix of the tag.
- `tags:"=.History.Military"` : True. `History.Military` is a prefix of the tag.
- `tags:"=.History.Military.WWII"` : True. `History.Military.WWII` is a prefix of the tag, albeit an improper one.
- `tags:"=.Military"` : False. `Military` is not a prefix of the tag.

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

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

"Regular expression" searches

Regular expression searches are indicated by prefixing the search string with a tilde (~). Any Python-compatible regular expression[3] can be used. Backslashes used to escape special characters in regular expressions must be doubled because single backslashes will be removed during query parsing. For example, to match a literal parenthesis you must enter `\\(` or alternatively use *super-quotes* (see below). Regular expression searches are "contains" searches unless the expression is anchored. Character variants are significant: `~e` doesn't match `é`.

"Character variant" searches

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

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

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

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

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

then these character variant searches find:

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

Search Expression Syntax

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

- `not tag:foo` finds all books that don't contain the tag `foo`

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

- `not (author:Asimov or author:Weber)` finds all books not written by either Asimov or Weber.

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

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

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

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

More information

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

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

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

on the left of the search box.

Available fields for searching are: `tag`, `title`, `author`, `publisher`, `series`, `series_index`, `rating`, `cover`, `comments`, `format`, `identifiers`, `date`, `pubdate`, `search`, `size`, `vl` and custom columns. If a device is plugged in, the `ondevice` field becomes available, when searching the calibre library view. To find the search name (actually called the *lookup name*) for a custom column, hover your mouse over the column header in the library view.

Dates

A sintaxe para pesquisar datas é:

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

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

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

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

Searching dates and numeric values with relational comparisons

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

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

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

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

Series indices

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

Saved searches

The special field `search` is used for *saved searches* (página 18). If you save a search with the name «My spouse's books» you can enter `search:"My spouse's books"` in the Search bar to reuse the saved search. More about saving searches below.

Virtual libraries

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

Whether a field has a value

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

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

Yes/no custom columns

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

Identifiers

Identifiers (e.g., ISBN, DOI, LCCN, etc.) use an extended syntax. An identifier has the form `type:value`, as in `isbn:123456789`. The extended syntax permits you to specify independently the type and value to search for. Both the type and the value parts of the query can use any of the *search kinds* (página 13). Examples:

- "identificadores: verdadeiro" encontrarão livros com qualquer identificador.
- "identificadores: falso" encontrarão livros sem identificador.
- `identifiers:123` will search for books with any type having a value containing *123*.
- `identifiers:=123456789` will search for books with any type having a value equal to *123456789*.

- `identifiers:=isbn:` irá encontrar livros que tenham um tipo para ISBN com um valor qualquer.
- `identifiers:=isbn:false` irá encontrar livros que não tenham um tipo para ISBN.
- `identifiers:=isbn:123` irá encontrar livros que tenham um tipo para ISBN com um valor contendo *123*.
- `identifiers:=isbn:=123456789` irá encontrar livros que tenham um tipo para ISBN com um valor igual a *123456789*.
- `identifiers:i:1` will find books with a type containing an *i* having a value containing a *1*.

Categories visible in the Tag browser

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

Search using templates

You can search using a template in *A linguagem de modelos do Calibre* (página 157) instead of a metadata field. To do so you enter a template, a search type, and the value to search for. The syntax is:

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

The `template` is any valid calibre template language template. The `search type` must be one of `t` (text search), `d` (date search), `n` (numeric search), or `b` (set/not set (boolean)). The `value` is whatever you want, and can use the *search kinds* (página 13) described above for the various search types. You must quote the entire search string if there are spaces anywhere in it.

Exemplos:

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

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

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

Advanced search dialog

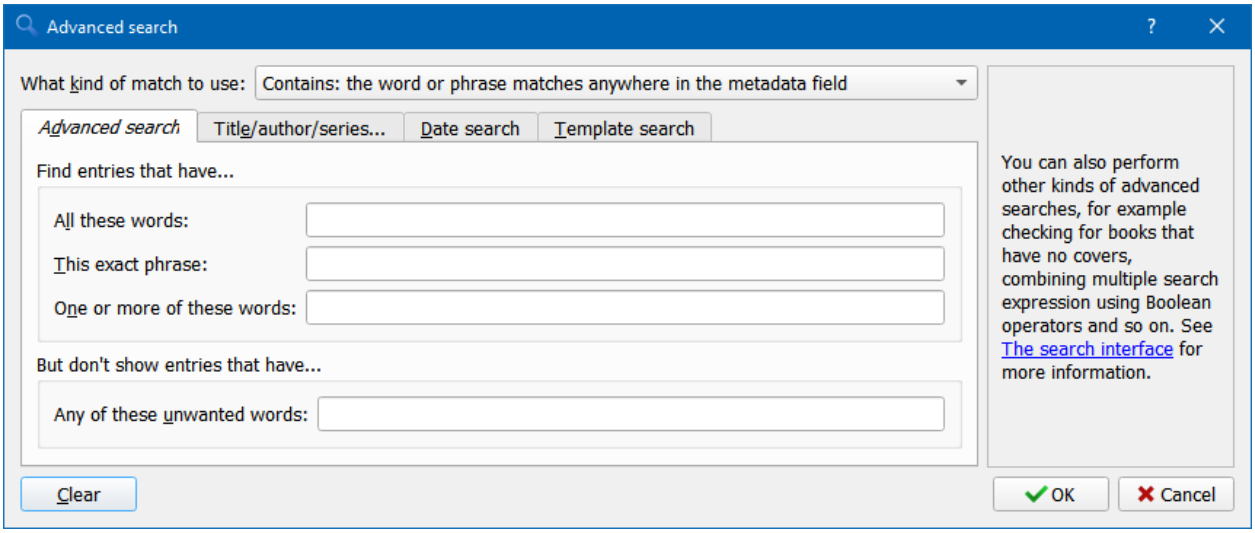


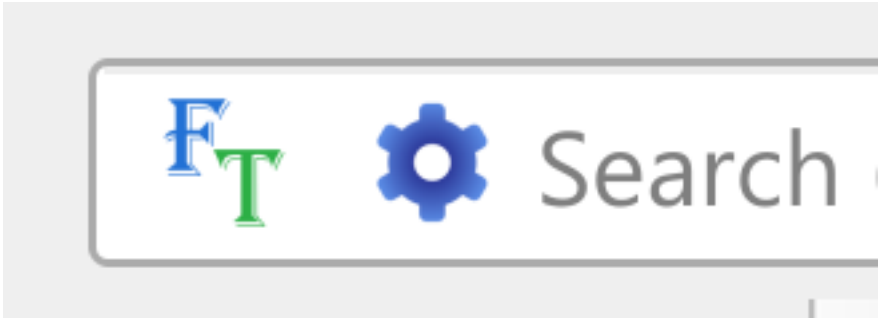
Figura1: *Advanced search dialog*

1.6 A guardar pesquisas

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

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

1.7 A pesquisar texto integral em todos os livros



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

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

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

1.8 Bibliotecas virtuais

A *Virtual library* is a way to pretend that your calibre library has only a few books instead of its full collection. This is an excellent way to partition your large collection of books into smaller, manageable chunks. To learn how to create and use Virtual libraries, see the tutorial: *Bibliotecas virtuais* (página 242).

1.9 Temporarily marking books

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

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

1.10 Calcular metadados a partir do nome dos ficheiros

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

You can also control how metadata is read from the filename using regular expressions (see *Tudo sobre utilização de expressões regulares no Calibre* (página 208)). In the *Adding books* section of the configuration dialog, you can specify a regular expression that calibre will use to try and guess metadata from the names of e-book files that you add to the library. The default regular expression is:

```
title - author
```

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

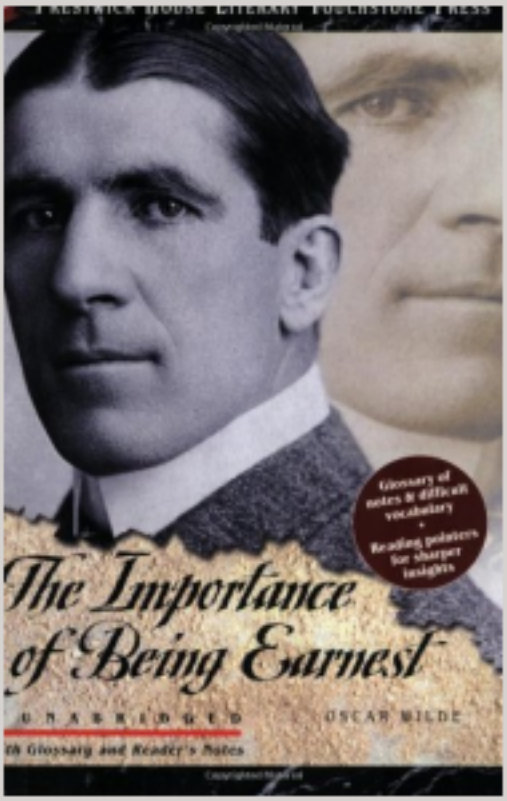
```
Foundation and Earth - Isaac Asimov.txt
```

será interpretada para ter o título: Fundação e Terra e autor: Isaac Asimov

---

Dica: Se o nome não contém o hífen, a expressão regular acima falhará.

1.11 Detalhes do livro



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

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

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

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

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

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

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

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

1.12 Navegador de conteúdos

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

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

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

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

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

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

1. «tudo que corresponde um item na categoria» indicado por um único sinal de mais verde.
2. «tudo que corresponde ao item na categoria ou suas subcategorias» indicados por dois sinais de mais verdes.
3. «tudo que não corresponde ao item na categoria» mostrado por um único sinal de meno vermelho.
4. «tudo que não corresponde ao item na categoria ou suas subcategorias» mostrado por dois sinais de menos vermelhos

It is also possible to create hierarchies inside some of the text categories such as tags, series, and custom columns. These hierarchies show with the small triangle, permitting the sub-items to be hidden. To use hierarchies of items in a category, you must first go to *Preferences → Interface → Look & feel* and enter the category name(s) into the «Categories with hierarchical items» field. Once this is done, items in that category that contain periods will be shown using the small triangle. For example, assume you create a custom column called «Genre» and indicate that it contains hierarchical items. Once done, items such as Mystery.Thriller and Mystery.English will display as Mystery with the small triangle next to it. Clicking on the triangle will show Thriller and English as sub-items. See *Managing subgroups of books, for example «genre»* (página 149) for more information.

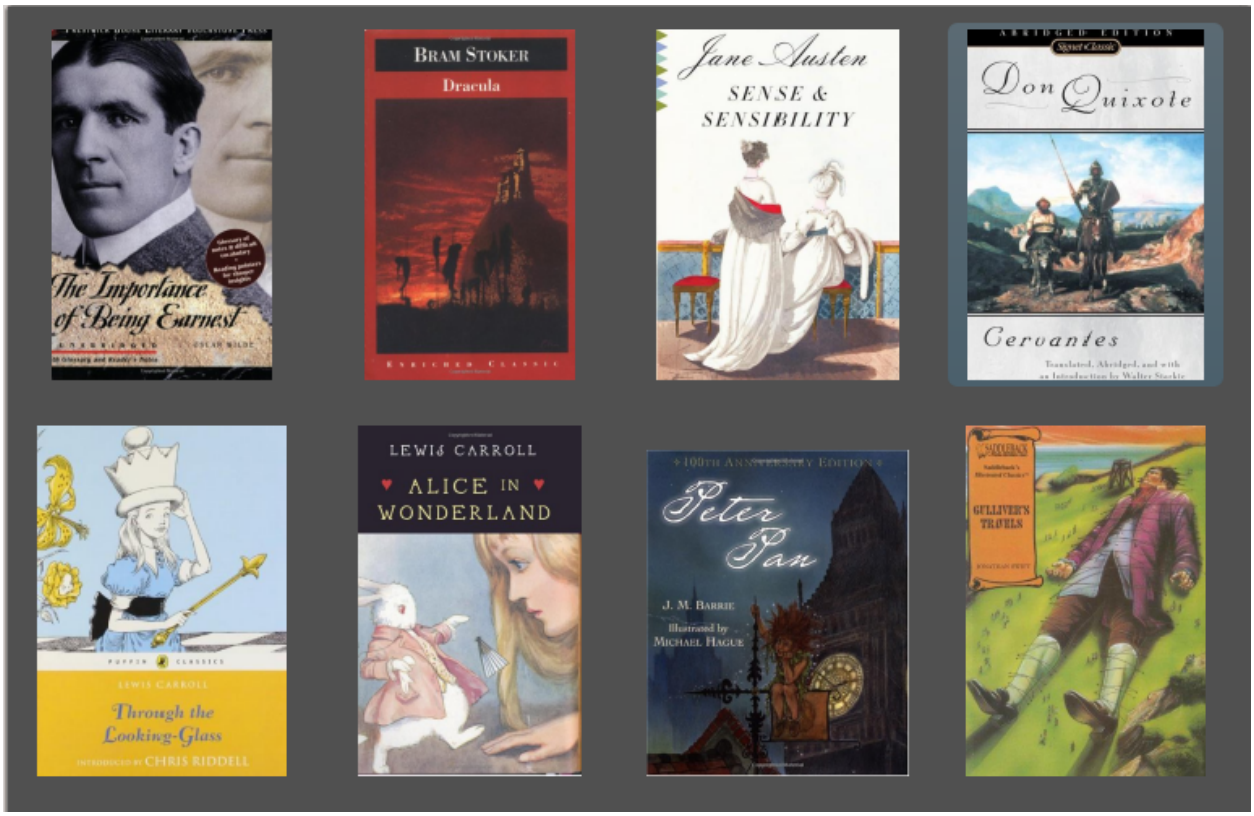
Hierarchical items (items with children) use the same four “click-on” searches as User categories. Items that do not have children use two of the searches: «everything matching» and «everything not matching».

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

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

Pode controlar o modo como os itens são ordenados no navegador de etiquetas através do botão: guilabel: “Configurar` no canto inferior esquerdo do navegador de etiquetas. Pode optar por classificar por nome, classificação média ou popularidade (popularidade é o número de livros com pelo menos um item na sua biblioteca. Por exemplo, a popularidade de Isaac Asimov é o número de livros de Isaac Asimov na sua biblioteca).

1.13 Grelha de capas



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

1.14 Navegador de capas

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

1.15 Visualização Rápida

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

Alguns exemplos de uso da visualização rápida: Ver se outros livros

- have some tag(s) applied to the currently selected book,
- está na mesma série que o livro atual
- Tem os mesmos valores do atual livro em uma coluna personalizada
- foram escritos por um dos mesmos autores do livro atual
- partilhar valores numa coluna personalizada

There are two choices for where the Quickview information appears:

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

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

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

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

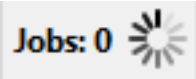
`Shift-` or `Ctrl-` double-click on a book in the Quickview window to open the edit metadata dialog on that book in the calibre window. The edited book will be Quickview'ed when you close the edit metadata dialog.

Você pode ver se uma coluna pode ser vista na Visualização Rápida passando o mouse sobre o cabeçalho da coluna e olhando no seu tooltip. Você pode também descobrir clicando com o botão direito do mouse no cabeçalho da coluna para ver se a opção «Visualização rápida» é mostrado, e selecionando essa opção é equivalente a pressionar "Q" na célula atual.

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

- Respeitar (ou não) a biblioteca virtual atual. Se assinalado, a vista rápida mostra apenas livros na biblioteca virtual atual. Predefinição: Respeitar bibliotecas virtuais
- Change the Quickview window contents when the column is changed on the book list using the cursor keys. Default: don't follow changes made with cursor keys
- Change the column being «quickview'ed» when a cell in the Quickview window is double-clicked. Otherwise the book is changed but the column being examined is not. Default: change the column
- Change the column being «quickview'ed» to the current column when the return key is pressed in the Quickview panel. Otherwise the book is changed but the column being examined is not. Default: change the column
- Choose which columns are shown in the Quickview window/panel.

1.16 Processos



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

1.17 Atalhos de teclado

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

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

Tabela1: Atalhos de teclado para o programa Calibre

Atalho de teclado	Ação
F2 (Enter for macOS)	Editar os metadados do campo atualmente selecionado na lista de livros.
A	Adicionar Livros
Shift+	Adicionar formatos aos livros selecionados
C	Converter livros selecionados
D	Enviar para o dispositivo
Del	Remover livros selecionados
E	Editar metadados dos livros selecionados
G	Obter livros
I	Mostrar detalhes do livro
K	Editar índice
M	Fundir os registos selecionados
Alt+M	Fundir os registos selecionados mas manter os originais
O	Abrir a pasta com o conteúdo
P	Refinar livros
S	Exportar para disco
T	Editar livro
V	Ver
Shift+	Ver o último livro lido
Alt+V/ Cmd+V for macOS	Ver formato específico
Alt+Sh	Alternar lista de processos
Alt+Sh	Alternar visualizador de capas
Alt+Sh	Toggle Book details panel
Alt+Sh	Alternar navegador de conteúdos
Alt+Sh	Alternar grelha de capas
Alt+A	Mostrar livros do mesmo autor que o livro atual
Alt+T	Mostrar livros com as mesmas etiquetas do livro atual
Alt+P	Mostrar livros publicados pela mesma editora do livro atual
Alt+Sh	Mostrar livros na mesma série do livro atual
/, Ctrl+F	Foco na barra de pesquisa
Shift+	Open the Advanced search dialog
Shift+	Toggle the Search bar
Esc	Limpar a pesquisa atual
Shift+	Focar a lista de livros
Ctrl+E	Limpar a biblioteca virtual
Alt+Es	Limpar as restrições adicionais
Ctrl+*	Cria uma biblioteca virtual temporária a partir da pesquisa atual
Ctrl+R	Selecionar a aba seguinte das bibliotecas virtuais
Ctrl+L	Selecionar a aba anterior das bibliotecas virtuais

continues on next page

Tabela 1 – continuação da página anterior

Atalho de teclado	Ação
N ou F3	Ache o próximo livro que corresponde com os parâmetros atuais de pesquisa (funciona somente se o destaque da busca estiver ligado nas preferências de busca)
Shift+ ou Shift+	Ache o livro anterior que corresponde com os parâmetros atuais de pesquisa (funciona somente se o destaque da busca estiver ligado nas preferências de busca)
Ctrl+D	Descarregar metadados e capas
Ctrl+R	Reiniciar Calibre
Ctrl+S	Reiniciar o calibre em modo de depuração
Shift+	Adicionar livros vazios ao Calibre
Ctrl+M	Toggle mark/unmarked status on selected books
Ctrl+/ or Ctrl+A	Open the popup to search the full text of all books in the library
Q	Abra a janela flutuante da Vista rápida para ver livros em séries/etiquetas/etc relacionadas.
Shift+	Centralize o painel de visualização rápida
Shift+	Realiza uma pesquisa no painel de vista rápida
F5	Reaplicar a ordenação atual
Ctrl+Q	Sair do Calibre
X	Toggle auto scroll of the book list
Ctrl+A	Restrict the displayed books to only those books that are in a category currently displayed in the *Tag browser*

CAPÍTULO 2

Adicionar o seu sítio de notícias favorito

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

Para compreender como usar a Framework, seguir os exemplos na ordem listada abaixo:

- *Descarregamento totalmente automático.* (página 30)
  - *O blogue do Calibre* (página 30)
  - *bbc.co.uk* (página 32)
- *Personalizar o processo de obtenção* (página 32)
  - *Utilizar a versão impressa da bbc.co.uk* (página 32)
  - *Substituir estilos de artigo* (página 33)
  - *Fatiar e cortar em secções* (página 34)
  - *Exemplo real* (página 34)
- *Dicas para desenvolver novas receitas* (página 37)
- *Leituras de aprofundamento* (página 38)
- *Documentação da API* (página 38)

2.1 Descarregamento totalmente automático.

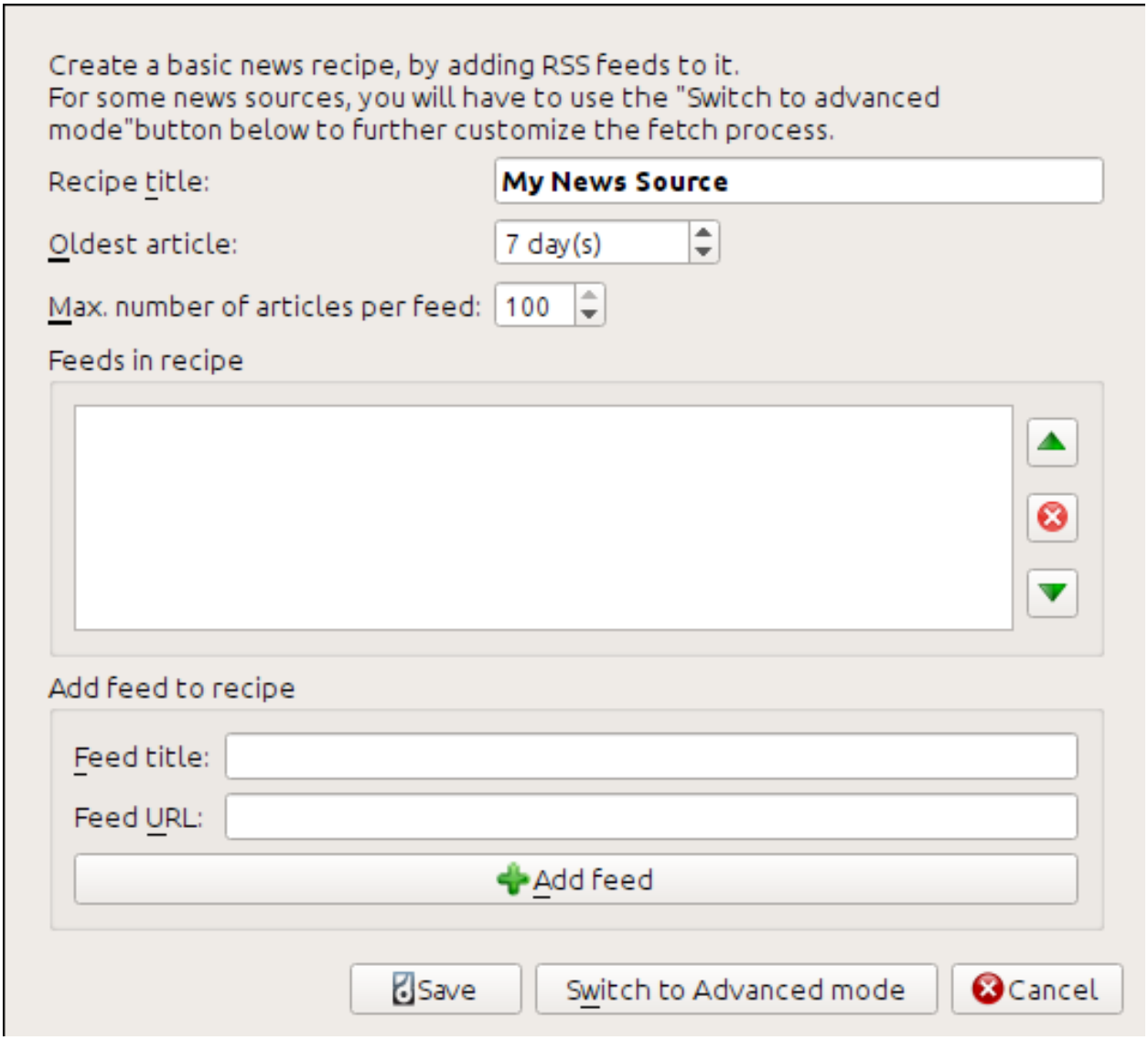
Se a sua fonte de notícias é bastante simples, calibre pode muito bem ser capaz de buscá-la por completo automaticamente, tudo que você precisa fazer é fornecer a URL. calibre reúne todas as informações necessárias para baixar uma fonte de notícias em um *recipe*. Para dizer ao calibre sobre uma fonte de notícias, você tem que criar um *recipe* para ele. Vamos ver alguns exemplos:

2.1.1 O blogue do Calibre

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

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

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



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

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

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

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

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

2.1.2 bbc.co.uk

Vamos tentar os dois sinais seguintes a partir de *The BBC*:

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

Follow the procedure outlined in *O blogue do Calibre* (página 30) above to create a recipe for *The BBC* (using the feeds above). Looking at the downloaded e-book, we see that calibre has done a creditable job of extracting only the content you care about from each article's webpage. However, the extraction process is not perfect. Sometimes it leaves in undesirable content like menus and navigation aids or it removes content that should have been left alone, like article headings. In order, to have perfect content extraction, we will need to customize the fetch process, as described in the next section.

2.2 Personalizar o processo de obtenção

Quando você quer aperfeiçoar o processo de baixar, ou baixar conteúdo de um complexo website particular, você pode aproveitar-se de todo o poder e flexibilidade do *recipe* framework. Afim de fazer isso, no diálogo *Add custom news sources*, simplesmente clique no botão *Switch to Advanced mode*.

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

2.2.1 Utilizar a versão impressa da bbc.co.uk

The first step is to look at the e-book we downloaded previously from *bbc.co.uk* (página 32). At the end of each article, in the e-book is a little blurb telling you where the article was downloaded from. Copy and paste that URL into a browser. Now on the article webpage look for a link that points to the «Printable version». Click it to see the print version of the article. It looks much neater! Now compare the two URLs. For me they were:

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

URL da versão impressa
https://newsvote.bbc.co.uk/mpapps/pagetools/print/news.bbc.co.uk/2/hi/science/nature/7312016.stm

Portanto, parece que para obter a versão impressa, precisamos prefixar cada artigo URL com:

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

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

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

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

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

In the above, `def print_version(self, url)` defines a *method* that is called by calibre for every article. `url` is the URL of the original article. What `print_version` does is take that url and replace it with the new URL that points to the print version of the article. To learn about Python[4] see the tutorial[5].

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

2.2.2 Substituir estilos de artigo

Na seção anterior, vimos que o tamanho da fonte para artigos da versão impressa do *The BBC» era muito pequeno. Na maioria dos sites, incluído *The BBC *, este tamanho da fonte é definido por meio de: term: *folhas de estilo CSS*. Podemos desativar a busca de tais folhas de estilo, adicionando a linha:

```
no_stylesheets = True
```

A receita tem agora esta aparência:

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

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/
```

A nova versão parece muito boa. Se é um perfecionista vai querer ler a próxima secção, que trata especificamente como modificar o conteúdo descarregado.

2.2.3 Fatiar e cortar em secções

calibre contains very powerful and flexible abilities when it comes to manipulating downloaded content. To show off a couple of these, let's look at our old friend the *The BBC* (página 33) recipe again. Looking at the source code (*HTML*) of a couple of articles (print version), we see that they have a footer that contains no useful information, contained in

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

Isto pode ser removido adicionando:

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

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

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

With these additions, our recipe has become «production quality».

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

2.2.4 Exemplo real

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

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

class NYTimes(BasicNewsRecipe):
```

(continues on next page)

(continuação da página anterior)

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

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

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

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

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

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

             elif ''.join(div['class']) in ['story', 'story headline']:
                 a = div.find('a', href=True)
                 if not a:
                     continue
                 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:
```

(continues on next page)

(continuação da página anterior)

```
                    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'))
```

We see several new features in this *recipe*. First, we have:

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

This sets the displayed time on the front page of the created e-book to be in the format, `Day, Day_Number Month, Year`. See *timefmt* (página 47).

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

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

These remove everything before the first `<h1>` tag and everything after the first tag whose id is `footer`. See *remove_tags* (página 46), *remove_tags_before* (página 46), *remove_tags_after* (página 46).

A próxima característica interessante é:

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

`needs_subscription = True` tells calibre that this recipe needs a username and password in order to access the content. This causes, calibre to ask for a username and password whenever you try to use this recipe. The code in *calibre.web.feeds.news.BasicNewsRecipe.get_browser()* (página 39) actually does the login into the NYT website. Once logged in, calibre will use the same, logged in, browser instance to fetch all content. See mechanize[6] to understand the code in `get_browser`.

The next new feature is the *calibre.web.feeds.news.BasicNewsRecipe.parse_index()* (página 40) method. Its job is to go to https://www.nytimes.com/pages/todayspaper/index.html and fetch the list of articles that appear in *todays* paper. While more complex than simply using *RSS*, the recipe creates an e-book that corresponds very

[6] https://mechanize.readthedocs.io/en/latest/

closely to the days paper. `parse_index` makes heavy use of BeautifulSoup[7] to parse the daily paper webpage. You can also use other, more modern parsers if you dislike BeautifulSoup. calibre comes with lxml[8] and html5lib[9], which are the recommended parsers. To use them, replace the call to `index_to_soup()` with the following:

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

The final new feature is the *calibre.web.feeds.news.BasicNewsRecipe.preprocess_html()* (página 41) method. It can be used to perform arbitrary transformations on every downloaded HTML page. Here it is used to bypass the ads that the nytimes shows you before each article.

2.3 Dicas para desenvolver novas receitas

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

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

The command **ebook-convert** will download all the webpages and save them to the EPUB file `myrecipe.epub`. The `-vv` option makes ebook-convert spit out a lot of information about what it is doing. The `ebook-convert-recipe-input --test` option makes it download only a couple of articles from at most two feeds. In addition, ebook-convert will put the downloaded HTML into the `debug/input` folder, where `debug` is the folder you specified in the *ebook-convert --debug-pipeline* (página 327) option.

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

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

Se está satisfeito com a sua receita, e se acredita que há procura suficiente para justificar a sua inclusão no conjunto de receitas predefinidas, publique a sua receita no fórum de receitas do Calibre[10] para a partilhar com outros utilizadores do Calibre.

Nota: No macOS, as ferramentas de linha de comando estão dentro do pacote do Calibre. Por exemplo, se instalou o Calibre em `/Applications` as ferramentas de linha de comando estão em `/Applications/calibre.app/Contents/MacOS/`.

Veja também:

***ebook-convert* (página 318)**
The command line interface for all e-book conversion.

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

2.4 Leituras de aprofundamento

Para saber mais sobre a escrita avançada de receitas usando algumas das instalações, disponível em «BasicNewsRecipe» você deve consultar as seguintes fontes:

API documentation **(página 38)**
Documentação da classe «BasicNewsRecipe» e todos os seus métodos e campos importantes.

BasicNewsRecipe[11]
O código fonte do «BasicNewsRecipe»

Built-in recipes[12]
The source code for the built-in recipes that come with calibre

O fórum de receitas do Calibre[13]
Lots of knowledgeable calibre recipe writers hang out here.

2.5 Documentação da API

2.5.1 Documentação da API para receitas

A API para escrever receitas é definida pela `BasicNewsRecipe` (página 38)

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

Classe de base que contém a lógica necessária em todas as receitas. Ao efetuar sobreposições progressivas das funcionalidades desta classe, pode criar, progressivamente, receitas mais personalizadas/poderosas. Para um tutorial introdutório à criação de receitas, veja *Adicionar o seu sítio de notícias favorito* (página 29).

abort_article(*msg=None*)

Invoque este método dentro de qualquer dos métodos de pré-processamento para abortar o descarregamento do artigo atual. É útil para descartar artigos que contêm conteúdo inapropriado, tais como artigos de vídeo puro.

abort_recipe_processing(*msg*)

Provoca, por parte do sistema de descarregamento de receitas, o abortar do descarregamento desta receita, apresentado ao utilizador uma mensagem simples de informação.

add_toc_thumbnail(*article*, *src*)

Invoque-o a partir de populate_article_metadata com o atributo src de uma etiqueta <img> no artigo que seja apropriado para uso como uma miniatura que represente o artigo no índice. Se essa miniatura é de facto utilizada depende do dispositivo usado (atualmente é apenas usada pelos Kindle). Note que a imagem referenciada tem de ser uma que já tenha sido descarregada com sucesso, sendo ignorada em caso contrário.

classmethod adeify_images(*soup*)

Se a sua receita, quando convertida para EPUB, tem problemas com imagens quando visionada no Adobe Digital Editions, chame este método a partir de `postprocess_html()` (página 41).

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

Devolve um conjunto de representações canónicas de `url`. A implementação padrão usa apenas o nome do servidor e o caminho do URL, ignorando tudo o que sejam parâmetros de pesquisa, fragmentos, etc. As representações canónicas têm de ser únicas no conjunto dos URL para esta fonte de notícias. Se o não forem, então as hiperligações internas podem ser incorretamente resolvidas.

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

Parâmetros

is_link – É Verdadeira se o URL deriva de uma ligação interna num ficheiro HTML. Falso se o URL é o URL usado para descarregar um artigo.

cleanup()

Invocado depois de todos os artigos terem sido descarregados. Use-o para efetuar tarefas de limpeza, tais como terminar sessão nos sítios subscritos, etc.

clone_browser(*br*)

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

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

default_cover(*cover_file*)

Cria uma capa genérica para receitas que não tem uma capa

download()

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

extract_readable_article(*html*, *url*)

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

get_article_url(*article*)

Override in a subclass to customize extraction of the *URL* that points to the content for each article. Return the article URL. It is called with *article*, an object representing a parsed article from a feed. See feedparser[14]. 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[15].

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

Return a browser instance used to fetch documents from the web. By default it returns a mechanize[16] browser instance that supports cookies, ignores robots.txt, handles refreshes and has a mozilla firefox user agent.

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

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

get_cover_url()

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

get_extra_css()

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

get_feeds()

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

get_masthead_title()

Override in subclass to use something other than the recipe title

get_masthead_url()

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

get_obfuscated_article(*url*)

If you set *articles_are_obfuscated* this method is called with every article URL. It should return the path to a file on the filesystem that contains the article HTML. That file is processed by the recursive HTML fetching engine, so it can contain links to pages/images on the web.

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

classmethod image_url_processor(*baseurl*, *url*)

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

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

Convenience method that takes an URL to the index page and returns a BeautifulSoup[17] of it.

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

is_link_wanted(*url*, *tag*)

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

Parâmetros

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

parse_feeds()

Create a list of articles from the list of feeds returned by *BasicNewsRecipe.get_feeds()* (página 40). Return a list of `Feed` objects.

parse_index()

This method should be implemented in recipes that parse a website instead of feeds to generate a list of articles. Typical uses are for news sources that have a «Print Edition» webpage that lists all the articles in the current print edition. If this function is implemented, it will be used in preference to *BasicNewsRecipe.parse_feeds()* (página 40).

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

Each list of articles must contain dictionaries of the form:

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

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

If you want to abort processing for some reason and have calibre show the user a simple message instead of an error, call *abort_recipe_processing()* (página 38).

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

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

Parâmetros

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

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

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

Parâmetros

- **oeb** – Um objeto OEBBook
- **opts** – Conversion options

postprocess_html (*soup*, *first_fetch*)

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

Parâmetros

- **soup** – A BeautifulSoup[18] instance containing the downloaded *HTML*.
- **first_fetch** – True if this is the first page of an article.

preprocess_html (*soup*)

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

soup: A BeautifulSoup[19] instance containing the downloaded *HTML*.

preprocess_image (*img_data*, *image_url*)

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

preprocess_raw_html(*raw_html*, *url*)

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

Note that this method acts *before* preprocess_regexps.

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

classmethod print_version(*url*)

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

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

publication_date()

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

skip_ad_pages(*soup*)

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

soup: A BeautifulSoup[20] instance containing the downloaded *HTML*.

sort_index_by(*index*, *weights*)

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

index: A list of titles.

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

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

Convenience method to take a BeautifulSoup[21] `Tag` and extract the text from it recursively, including any CDATA sections and alt tag attributes. Return a possibly empty Unicode string.

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

tag: BeautifulSoup[22] `Tag`

articles_are_obfuscated = False

Set to True and implement *get_obfuscated_article()* (página 40) to handle websites that try to make it difficult to scrape content.

auto_cleanup = False

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

auto_cleanup_keep = None

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

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

center_navbar = True

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

compress_news_images = False

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

compress_news_images_auto_size = 16

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

compress_news_images_max_size = None

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

conversion_options = {}

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

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

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

By default, the cover image returned by get_cover_url() will be used as the cover for the periodical. Overriding this in your recipe instructs calibre to render the downloaded cover into a frame whose width and height are expressed as a percentage of the downloaded cover. cover_margins = (10, 15, "#ffffff") pads the cover with a white margin 10px on the left and right, 15px on the top and bottom. Color names are defined here[23]. Note that for some reason, white does not always work in Windows. Use #ffffff instead

delay = 0

Delay between consecutive downloads in seconds. The argument may be a floating point number to indicate a more precise time.

description = ''

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

encoding = None

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

extra_css = None

Specify any extra *CSS* that should be added to downloaded *HTML* files. It will be inserted into *<style>* tags, just before the closing *</head>* tag thereby overriding all *CSS* except that which is declared using the style attribute on individual *HTML* tags. Note that if you want to programmatically generate the extra_css override the *get_extra_css()* (página 39) method instead. For example:

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

feeds = None

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

filter_regexps = []

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

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

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

Only one of *BasicNewsRecipe.match_regexps* (página 45) or *BasicNewsRecipe.filter_regexps* (página 44) should be defined.

handle_gzip = True

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

ignore_duplicate_articles = None

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

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

Para usar URL em vez disso, definir para:

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

Para fazer coincidir no título ou URL, definir para:

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

keep_only_tags = []

Keep only the specified tags and their children. For the format for specifying a tag see *BasicNewsRecipe.remove_tags* (página 46). If this list is not empty, then the *<body>* tag will be emptied and re-filled with the tags that match the entries in this list. For example:

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

will keep only tags that have an *id* attribute of *«content»* or *«heading»*.

language = 'und'

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

masthead_url = None

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

match_regexps = []

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

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

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

Only one of *BasicNewsRecipe.match_regexps* (página 45) or *BasicNewsRecipe.filter_regexps* (página 44) should be defined.

max_articles_per_feed = 100

Maximum number of articles to download from each feed. This is primarily useful for feeds that don't have article dates. For most feeds, you should use *BasicNewsRecipe.oldest_article* (página 45)

needs_subscription = False

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

no_stylesheets = False

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

oldest_article = 7.0

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

preprocess_regexps = []

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

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

will remove everything from *<!–Article ends here–>* to *</body>*.

publication_type = 'unknown'

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

recipe_disabled = None

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

recursions = 0

Number of levels of links to follow on article webpages

remove_attributes = []

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

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

remove_empty_feeds = False

If True empty feeds are removed from the output. This option has no effect if parse_index is overridden in the sub class. It is meant only for recipes that return a list of feeds using *feeds* or *get_feeds()* (página 40). It is also used if you use the ignore_duplicate_articles option.

remove_javascript = True

Convenient flag to strip all JavaScript tags from the downloaded HTML

remove_tags = []

Lista de etiquetas a serem removidas. As etiquetas especificadas são removidas do HTML descarregado. Uma etiqueta é especificado como um dicionário na forma:

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

All keys are optional. For a full explanation of the search criteria, see Beautiful Soup[24] A common example:

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

Isto removerá todas as etiquetas <*div class=»advert»*> e respetivos dependentes do :term: *HTML* descarregado.

remove_tags_after = None

Remove todos as etiquetas que ocorram depois da etiqueta especificada. Para informações sobre o formato de especificação de uma etiqueta, veja *BasicNewsRecipe.remove_tags* (página 46). Por exemplo:

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

will remove all tags after the first element with *id=»content»*.

remove_tags_before = None

Remove all tags that occur before the specified tag. For the format for specifying a tag see *BasicNewsRecipe.remove_tags* (página 46). For example:

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

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

requires_version = (0, 6, 0)

Minimum calibre version needed to use this recipe

resolve_internal_links = False

If set to True then links in downloaded articles that point to other downloaded articles are changed to point to the downloaded copy of the article rather than its original web URL. If you set this to True, you might also need to implement *canonicalize_internal_url()* (página 38) to work with the URL scheme of your particular website.

reverse_article_order = False

Reverse the order of articles in each feed

scale_news_images = None

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

scale_news_images_to_device = True

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

simultaneous_downloads = 5

Number of simultaneous downloads. Set to 1 if the server is picky. Automatically reduced to 1 if *BasicNewsRecipe.delay* (página 43) > 0

summary_length = 500

Max number of characters in the short description

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

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

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

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

timeout = 120.0

Tempo de espera em segundos para obter ficheiros do servidor

title = 'Fonte de notícias desconhecida'

The title to use for the e-book

use_embedded_content = None

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

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

CAPÍTULO 3

Visualizador de livros digitais

O Calibre inclui um visualizador de ebook incorporado que pode exibir os principais formatos de ebook. O visualizador de livros eletrónicos é altamente personalizável e possui vários recursos avançados.

- *Iniciar o visualizador de livros* (página 50)
- *Navegar por um livro digital* (página 50)
- *Realce de texto* (página 51)
- *Ler em voz alta* (página 51)
- *Searching the text* (página 52)
- *Following links using only the keyboard* (página 52)
- *Personalizar o aspeto visual da sua experiência de leitura* (página 52)
- *Consulta de dicionário* (página 53)
- *Copiar texto e imagens* (página 53)
- *Ampliar em imagens* (página 53)
- *Non re-flowable content* (página 53)
- *Designing your book to work well with the calibre viewer* (página 53)

3.1 Iniciar o visualizador de livros

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

3.2 Navegar por um livro digital

Pode «virar páginas» num livro usando as seguintes opções:

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

You can access the viewer controls by either:

- Clique com o botão direito no texto
- Premir `Esc` ou teclas de `Menu`
- On a touchscreen by tapping the top 1/3rd of the screen

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

3.2.1 Marcadores

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

3.2.2 Índice

Se o livro que estiver a ler possuir um índice, você pode aceder ao índice premindo o botão *Índice* . Isso exibirá uma lista das secções do livro. Poderá então clicar em qualquer uma delas para ir para essa parte do livro.

3.2.3 Navegar por localização

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

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

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

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

3.2.4 Modo de referências

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

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

3.3 Realce de texto

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

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

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

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

3.4 Ler em voz alta

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

Poderá também ler em voz alta passagens destacadas adicionando o botão *Ler em voz alta* à barra de seleção nas preferências de visualização em *Comportamento de seleção*.

Nota: O suporte texto-para-voz em navegadores é muito incompleto e contém muitos erros. A qualidade de funcionamento do *Ler em voz alta* no visualizador embebido no navegador depende da qualidade da implementação subjacente do texto-para-voz nesse navegador. Especificamente, realçar a palavra atual não funciona, e alterar a velocidade ou a voz implica que a leitura será reiniciada a partir do início.

Nota: No Linux, *Ler em voz alta* requer que o Speech Dispatcher[25] esteja instalado e a funcionar.

---

[25] https://freebsoft.org/speechd

Nota: No Windows, as vozes instaladas podem não estar todas visíveis no subsistema SAPI que é usado no texto-para--voz. Nesta ligação encontra instruções para tornar todas as vozes visíveis[26].

3.5 Searching the text

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

1. *Contains* - The simplest default mode. The text entered in the search box is searched for anywhere. All punctuation, accents and spaces are ignored. For example, the search: `Pena` will match all of the following: `penal, pen a, pen.a and Peña`. If you select the *Case sensitive* box then accents, spaces and punctuation are no longer ignored.
2. *Whole words* - Searches for whole words. So for example, the search `pena` will match the word `Peña` but not the word `Penal`. As with *Contains* searches above, accents and punctuation are ignored unless the *Case sensitive* box is checked.
3. *Nearby words* - Searches for whole words that are near each other. So for example, the search `calibre cool` will match places where the words `calibre` and `cool` occur within sixty characters of each other. To change the number of characters add the new number to the end of the list of words. For instance, `calibre cool awesome 120` will match places where the three words occur within 120 characters of each other. Note that punctuation and accents are *not* ignored for these searches.
4. *Regex* - Interprets the search text as a *regular expression*. To learn more about using regular expressions, see *the tutorial* (página 208).

3.6 Following links using only the keyboard

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

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

3.7 Personalizar o aspeto visual da sua experiência de leitura

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

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

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

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

[26] https://www.mobileread.com/forums/showpost.php?p=4084051&postcount=108

More advanced customization can be achieved by the *Styles* settings. Here you can specify a background image to display under the text and also a stylesheet you can set that will be applied to every book. Using it you can do things like change paragraph styles, text justification, etc. For examples of custom stylesheets used by calibre's users, see the forums[27].

3.8 Consulta de dicionário

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

3.9 Copiar texto e imagens

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

3.10 Ampliar em imagens

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

3.11 Non re-flowable content

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

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

3.12 Designing your book to work well with the calibre viewer

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

`body.calibre-viewer-dark-colors`
: Set when using a dark color scheme

`body.calibre-viewer-light-colors`
: Set when using a light color scheme

`body.calibre-viewer-paginated`
: Set when in paged mode

`body.calibre-viewer-scrolling`
: Set when in flow (non-paginated) mode

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

Finally, you can use the calibre color scheme colors via CSS variables[28]. The calibre viewer defines the following variables: `--calibre-viewer-background-color`, `--calibre-viewer-foreground-color` and optionally `--calibre-viewer-link-color` in color themes that define a link color.

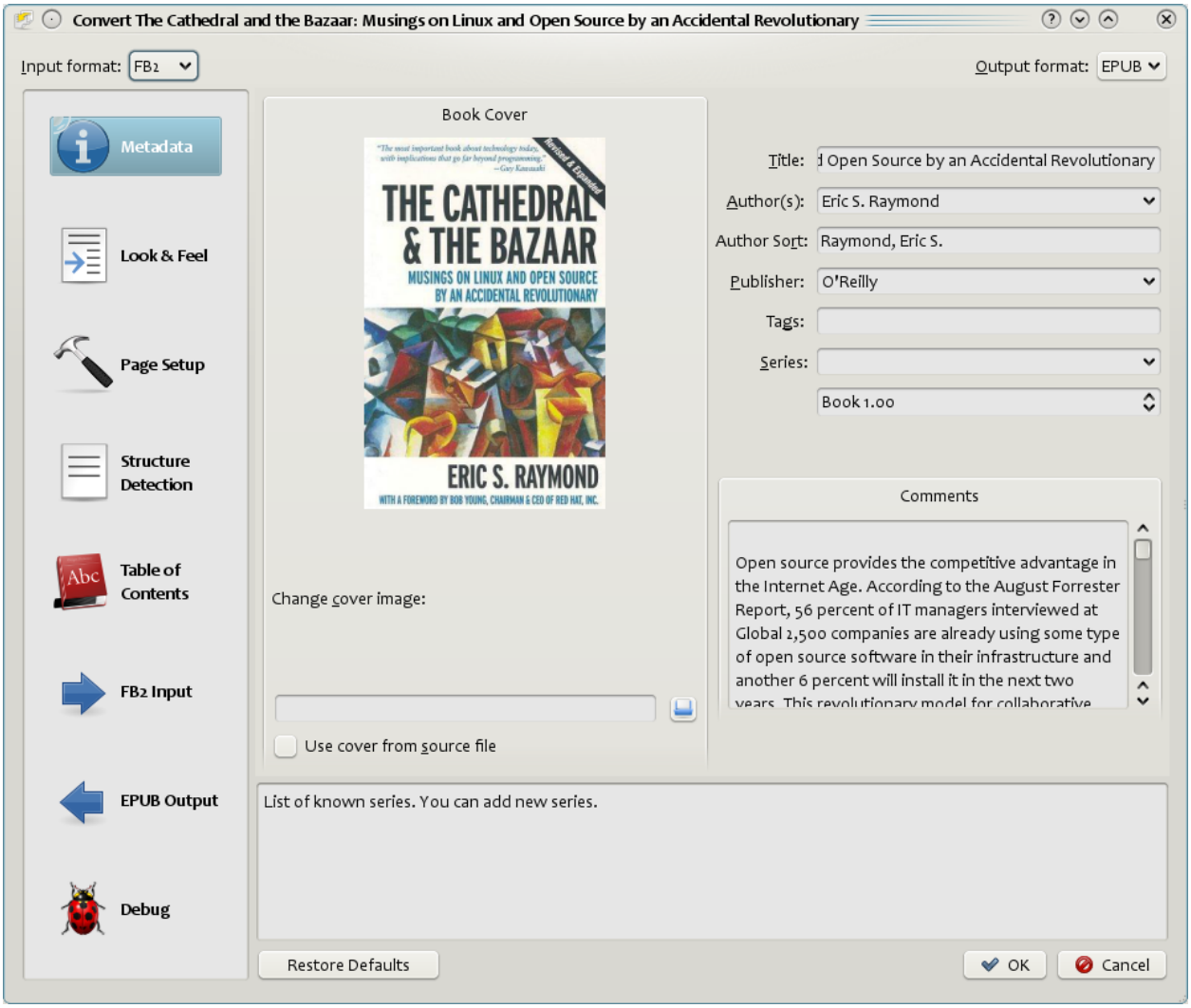
[28] https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties

CAPÍTULO 4

Conversão de livros digitais

O calibre tem um sistema de conversão que é projetado para ser muito fácil de usar. Normalmente, você só adiciona um livro ao calibre, clica em converter e o calibre tentará criar um resultado o mais próximo possível da origem. Contudo, o calibre aceita um número muito grande de formatos de origem e uns não são tão adequados quanto outros para conversão em ebooks. No caso de tais formatos de origem, ou se você quer apenas um controle maior sobre o sistema de conversão, o calibre tem uma gama de opções para ajustar o processo de conversão. Note, contudo, que o sistema de conversão do calibre não é um substituto de um editor de ebooks completo. Para editar ebooks eu recomendo que primeiro se converta para EPUB ou AZW3, usando o calibre e, então, usar a função "Editar Livro" para o ajustá-lo. Você poderá, então, utilizar o ebook já editado e converter o mesmo em outros formatos dentro do calibre.

Esse documento fará referência principalmente às configurações de conversão, da maneira como são encontradas na caixa de diálogo de conversão, mostrada na figura abaixo. Todas essas configurações também estão disponíveis por meio da interface de linha de comando para conversão, esclarecida em generated/pt_BR/ebook-convert. Dentro do calibre, você pode obter ajuda sobre qualquer configuração individual, mantendo seu mouse sobre ela. Uma dica irá aparecer, descrevendo a configuração.

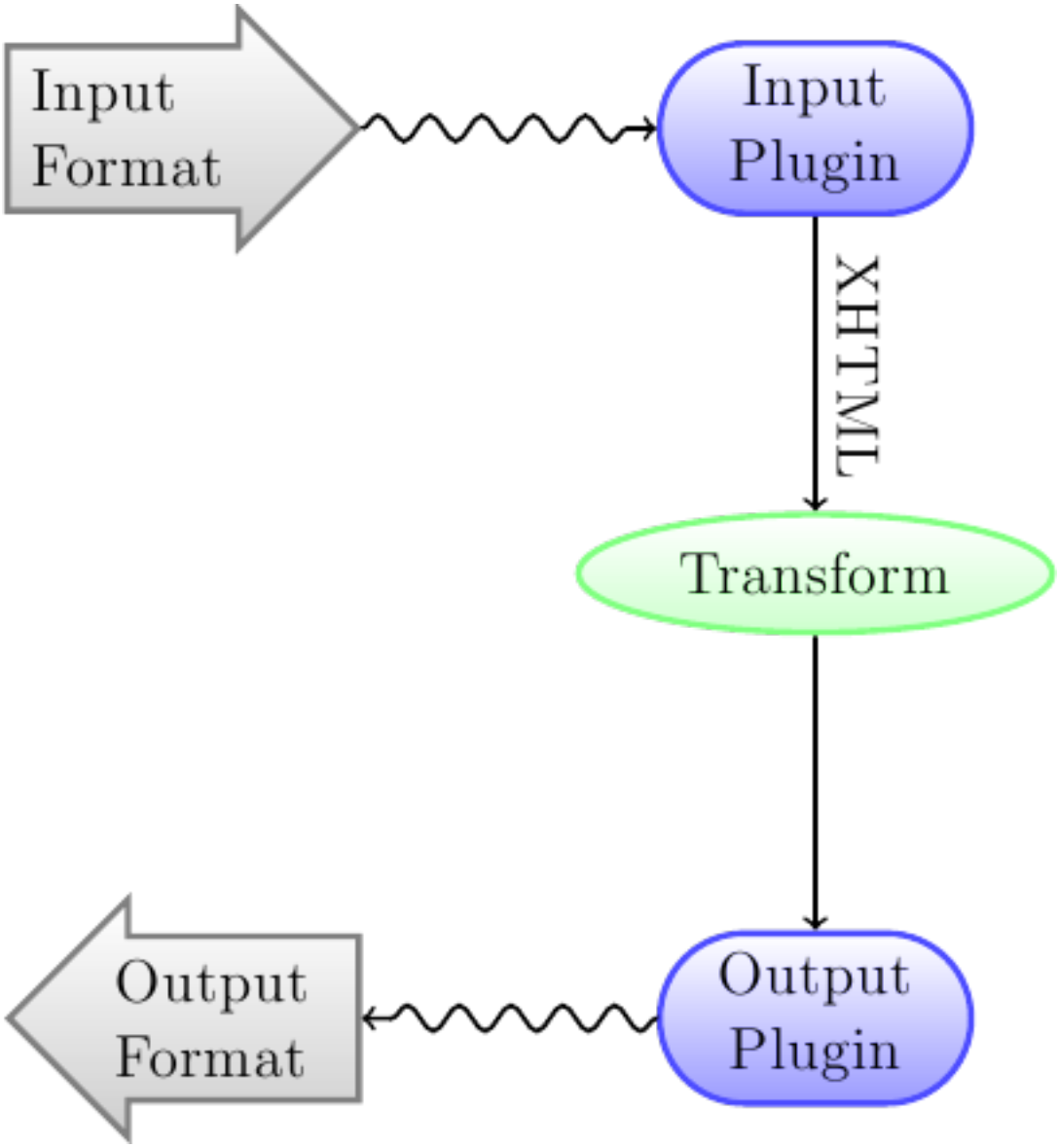


Conteúdos

- *Introdução* (página 57)
- *Aparência* (página 58)
- *Configuração de página* (página 61)
- *Processamento heurístico* (página 61)
- *Pesquisar e substituir* (página 63)
- *Deteção de estrutura* (página 63)
- *Índice* (página 65)
- *Usar imagens como títulos de capítulos ao converter documentos de origem em HTML* (página 66)
- *Usar atributos de etiquetas para fornecer o texto para entradas no índice* (página 67)
- *Modo como as opções são definidas/guardadas para a conversão* (página 67)
- *Dicas específicas do formato* (página 68)

4.1 Introdução

A primeira coisa a entender sobre o sistema de conversão é que ele foi projetado como um fluxo. Pode ser esquematizado da seguinte forma:



Primeiramente, o formato de entrada é convertido em XHTML pelo *plugin de entrada* adequado. Então o HTML é *transformado*. Por fim, o XHTML processado é convertido no formato de saída pelo *plugin de saída* apropriado. Os resultados da conversão podem variar muito, de acordo com o formado de entrada, alguns são mais bem convertidos do que outros. Uma lista dos melhores formatos de origem para conversão está disponível *aqui* (página 126).

As transformações sobre XHTML de saída são onde todo o trabalho acontece. Há diversas transformações, como a inserção dos metadados do livro como página no início do livro, a criação do Sumário com base nos títulos dos capítulos e o ajuste proporcional do tamanho das fontes. Importante lembrar que todas as transformações no XHTML de saída se devem ao *plugin de entrada*, e não ao arquivo de entrada propriamente. Assim, para ser convertido de RTF em EPUB, o arquivo é antes convertido internamente em XHTML, que recebe as várias transformações e só então o *plugin de saída* cria o arquivo EPUB, gerando automaticamente todos os metadados, o Sumário etc.

You can see this process in action by using the debug option . Just specify the path to a folder for the debug output. During conversion, calibre will place the XHTML generated by the various stages of the conversion pipeline in different sub-folders. The four sub-folders are:

Tabela1: Etapas do processo de conversão

Pasta	Descrição
ficheiro de origem	Contém o HTML de saída gerado pelo plugin de entrada. Use-o para depurar o plugin de entrada.
analisado	É o resultada do pré-processamento da saída e conversão em XHTML pelo plugin de entrada. Use-o para depurar a detecção da estrutura.
estrutura	Pós deteção de estrutura, mas antes do achatamento de CSS e conversão de tamanho de fonte. Utilizado para depurar a conversão de tamanhos de fontes e transformação de CSS.
processado	Logo antes de o e-book passar para o plugin de saída. Use-o para depurar o plugin de saída.

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

Aqui são tratadas várias transformações que atuam no XHTML intermediário e como controlá-las. Ao final, são apresentadas dicas específicas para cada tipo de formato de entrada/saída.

4.2 Aparência

Conteúdos

- *Fontes* (página 59)
- *Texto* (página 60)
- *Visual* (página 60)
- *Estilizar* (página 60)
- *Transformar estilos* (página 61)
- *Transformar HTML* (página 61)

Esse conjunto de opções controla vários aspectos da aparência do e-book convertido.

4.2.1 Fontes

Uma das características mais agradáveis da experiência e-reading, é a capacidade de ajustar facilmente o tamanho das fontes para atender as necessidades individuais e condições de iluminação. calibre tem algoritmos sofisticados para garantir que todos os livros que produz tem um tamanho de fonte consistente, não importa o tamanho de fonte especificado no documento de entrada.

O tamanho básico da fonte de um documento é o tamanho mais comum no referido documento, isto é, o tamanho do volume de texto no documento. Quando você especificar :guilabel:"Base font size", calibre automaticamente redimensionará todos os tamanhos de fonte no documento proporcionalmente, de modo que o tamanho de fonte mais comum torna-se o tamanho básico de fonte especificado e os outros tamanhos de fonte são redimensionados adequadamente. Ao escolher um tamanho básico de fonte maior, você pode tornar as fontes do documento maior ou vice-versa Quando você define o tamanho básico da fonte, para melhores resultados, você deve também definir o tamanho da fonte chave.

Via de regra, o calibre escolhe automaticamente um tamanho-padrão de fonte adequado para o perfil de saída que você escolheu (leia *Configuração de página* (página 61)). No entanto, você pode alterá-lo aqui, caso o padrão não esteja adequado.

A opção :guilabel:"Font size key" permite controlar como tamanho não-básico de fonte é redimensionado. O algoritmo de redimensionamento de fonte trabalha utilizando um tamanho de fonte chave, que é simplesmente uma lista de tamanhos de fonte separados por vírgula. O tamanho da fonte chave diz ao calibre quantos «passos» maior ou menor um determinado tamanho de fonte deve ser comparado com o tamanho básico da fonte. A ideia é deve ter um número limitado de tamanhos de fonte no documento. Por exemplo, um tamanho para o texto do corpo, alguns tamanhos de diferentes níveis de título e alguns super/sub scripts e notas de rodapé. O tamanho da fonte chave permite ao calibre compartamentalizar o tamanho das fontes nos documentos de entrada em «caixotes» separados correspondendo aos diferentes tamanhos de fonte lógicos.

Vamos ilustrar com um exemplo. Suponha que o documento que está a ser convertido foi produzido por alguém com uma visão excelente e tem um tamanho básico de fonte de 8pt. Isso significa que a maior parte do texto no documento está dimensionado para 8pts, os títulos ficam num tamanho um pouco maior (digamos 10 e 12pt) e as notas de rodapé um tamanho de fonte pouco menor (em 6pt). Assim, se usmarmos as seguintes configurações:

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

O documento de saída terá um tamanho básico de fonte de 12pt, títulos de 14 e 16pt e notas de rodapé de 8pt. Agora, suponha que queremos que o título de maior tamanho se destaque e as notas de rodapé um pouco maior também. Para conseguir isso, o tamanho de fonte chave deve ser alterado para :

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

O título maior agora será de 18pt, enquanto as notas de rodapé serão de 9pt. Você pode jogar com estas configurações para tentar descobrir qual seria a melhor configuração usando o assistente de redimensionamento, que pode ser acessado clicando no pequeno botão ao lado da definição de :guilabel:"Font size key"

Todo o redimensionamento do tamanho da fonte feita na conversão pode ser desativada aqui, se você quiser preservar o tamanho das fontes do documento de entrada.

Uma configuração relacionada é a *Altura da linha*, que controla o tamanho vertical das linhas de texto. Por padrão (altura da linha 0), não há manipulação da altura das linhas. Se você especificar um valor diferente, a altura vai ser aplicada a todos os locais que não tiverem seu próprio valor. No entanto, trata-se de um recurso arriscado e deve ser usado com cautela. Para ajustar a altura de linha de uma seção do arquivo de entrada, prefira *Extra CSS* (página 60).

Nessa seção, você ainda pode configurar o calibre para embutir as fontes citadas no livro, o que permite que as fontes operem no dispositivo de leitura, mesmo que não estejam disponíveis nele.

4.2.2 Texto

O texto pode ou não ser justificado. Um texto justificado apresenta mais espaço entre as palavras para uniformizar a margem direita. Nem todo mundo gosta de texto justificado. Normalmente, o calibre retém a justificação no documento original. Para alterar o padrão, use a opção *Texto justificado* nessa seção.

Você pode ainda configurar o calibre para *Pontuação inteligente*, que substitui aspas, traços e reticências sem formatação pelas alternativas tipográficas corretas. O algoritmo, no entanto, não é perfeito, e o resultado precisa ser revisto. Também está disponível um algorítimo contrário, a *Pontuação não inteligente*.

Finalmente, existe: guilabel: *Codificação de caracteres de entrada*. Documentos antigos às vezes não especificam a codificação de caracteres. Quando convertido, isso pode resultar em caracteres não ingleses ou caracteres especiais, como aspas inteligentes, sendo corrompidos. O calibre tenta detectar automaticamente a codificação de caracteres do documento de origem, mas nem sempre é bem-sucedido. Você pode forçá-lo a assumir uma codificação de caracteres específica usando essa configuração. *cp1252* é uma codificação comum para documentos produzidos usando o software Windows. Você também deve ler: ref: *char-encoding-faq* para mais informações sobre problemas de codificação.

4.2.3 Visual

Normalmente, os parágrafos em XHTML são renderizados com uma linha em branco entre eles e sem recuo de texto inicial. O calibre tem algumas opções para controlar isso. : guilabel: *Remova o espaçamento entre parágrafos "forçosamente, assegure que todos os parágrafos não tenham espaçamento entre parágrafos. Também define o recuo do texto como 1,5em (pode ser alterado) para marcar o início de cada parágrafo. : guilabel: `Inserir linha em branco* faz o oposto, garantindo que exista exatamente uma linha em branco entre cada par de parágrafos. Ambas as opções são muito abrangentes, removendo o espaçamento ou inserindo-o em * todos * parágrafos (tecnicamente e tags). Isso é para que você possa definir a opção e garantir que ela funcione conforme anunciado, independentemente de como o arquivo de entrada esteja confuso. A única exceção é quando o arquivo de entrada usa quebras de linha rígida para implementar o espaçamento entre parágrafos.

Se você deseja remover o espaçamento entre todos os parágrafos, exceto alguns selecionados, não use essas opções. Em vez disso, adicione o seguinte código CSS a: ref: `CSS extra

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

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

Outra opção útil é: guilabel: *Linearizar tabelas*. Alguns documentos mal projetados usam tabelas para controlar o layout do texto na página. Quando convertidos, esses documentos geralmente possuem texto que sai da página e outros artefatos. Esta opção extrai o conteúdo das tabelas e o apresenta de forma linear. Observe que essa opção lineariza * todas * as tabelas, portanto, use-a somente se tiver certeza de que o documento de entrada não usa tabelas para fins legítimos, como apresentar informações tabulares.

4.2.4 Estilizar

A opção: guilabel: *CSS extra* permite especificar CSS arbitrário que será aplicado a todos os arquivos HTML na entrada. Esse CSS é aplicado com prioridade muito alta e, portanto, deve substituir a maioria dos CSS presentes no próprio ** documento de entrada **. Você pode usar esta configuração para ajustar a apresentação / layout do seu documento. Por exemplo, se você quiser que todos os parágrafos da classe *endnote* estejam alinhados à direita, basta adicionar:

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

ou se deseja alterar a indentação de todos os parágrafos:

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

:guilabel:"Extra CSS" é uma opção muito poderosa, mas você precisa enter como funciona o CSS para usar todo seu potencial. Você pode usar a opção de debug pipeline descrita acima para ver como o CSS está presente em seu documento de entrada.

Uma opção mais simples é usar: guilabel: *Filtrar informações de estilo*. Isso permite remover todas as propriedades CSS dos tipos especificados do documento. Por exemplo, você pode usá-lo para remover todas as cores ou fontes.

4.2.5 Transformar estilos

Esta é a instalação mais poderosa relacionada ao estilo. Você pode usá-lo para definir regras que mudam de estilo com base em várias condições. Por exemplo, você pode usá-lo para alterar todas as cores verdes para azuis ou remover todo o estilo em negrito do texto ou colorir todos os títulos de uma determinada cor, etc.

4.2.6 Transformar HTML

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

4.3 Configuração de página

As opções: guilabel: *Page setup* são para controlar o layout da tela, como margens e tamanhos de tela. Existem opções para configurar as margens da página, que serão usadas pelo plug-in de saída, se o formato de saída selecionado suportar as margens da página. Além disso, você deve escolher um perfil de entrada e um perfil de saída. Ambos os conjuntos de perfis lidam basicamente com como interpretar medições nos documentos de entrada / saída, tamanhos de tela e teclas de redimensionamento de fonte padrão.

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

O perfil de saída também controla o tamanho da tela. Isso fará com que, por exemplo, as imagens sejam redimensionadas automaticamente para caberem na tela em alguns formatos de saída. Portanto, escolha o perfil de um dispositivo com tamanho de tela semelhante ao seu.

4.4 Processamento heurístico

O processamento heurístico fornece uma variedade de funções que podem ser usadas para tentar detectar e corrigir problemas comuns em documentos de entrada mal formatados. Use essas funções se o documento de entrada sofrer uma formatação ruim. Como essas funções dependem de padrões comuns, saiba que, em alguns casos, uma opção pode levar a piores resultados, portanto, use com cuidado. Como exemplo, várias dessas opções removerão todas as entidades sem espaço de quebra ou podem incluir correspondências falsas positivas relacionadas à função.

:guilabe:"Habilitar processamento heurístico"
: Esta opção ativa o estágio calibre's: guilabel: *Heuristic processing* do pipeline de conversão. Isso deve estar ativado para que várias sub-funções sejam aplicadas

:guilabel:"Eliminar quebra de linhas"
A ativação dessa opção fará com que o calibre tente detectar e corrigir quebras de linha rígida existentes em um documento usando pistas de pontuação e comprimento de linha. O calibre primeiro tentará detectar se existem quebras de linha rígida; se elas não parecem existir, o calibre não tentará desembrulhar as linhas. O fator de desembrulhar da linha pode ser reduzido se você quiser "forçar" o calibre a desembrulhar as linhas.

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

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

Esta função não criará um sumário, mas em muitos casos fará com que as configurações padrão de detecção de capítulo do calibre detectem corretamente capítulos e construam um sumário. Ajuste o XPath em Detecção de estrutura se um sumário não for criado automaticamente. Se não houver outros títulos usados no documento, a configuração «// h: h2» em Detecção de estrutura seria a maneira mais fácil de criar um sumário para o documento.

Os cabeçalhos inseridos não são formatados, para aplicar a formatação, use a opção: guilabel: *CSS extra* nas configurações de conversão Aparência. Por exemplo, para centralizar tags de cabeçalho, use o seguinte

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

Renumerar sequências de <h1> ou <h2> etiquetas
Alguns editores formatam os cabeçalhos dos capítulos usando várias ou tags sequencialmente. as configurações de conversão padrão do calibre farão com que esses títulos sejam divididos em duas partes. Esta opção renumerará as tags de cabeçalho para evitar a divisão.

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

:guilabel:"Assegura que as quebras de cena estejam formatadas consistentemente"
Com esta opção, o calibre tentará detectar marcadores comuns de quebra de cena e garantir que eles estejam alinhados ao centro. Os marcadores de quebra de cena "suaves", ou seja, quebras de cena definidas apenas por espaço em branco extra, são estilizados para garantir que eles não sejam exibidos em conjunto com as quebras de página.

:guilabel:"Substituir quebras de cena"
Se esta opção estiver configurada então o Calibre substituirá os marcadores de quebra de cena que encontrar pelo texto de substituição especificado pelo utilizador. Por favor, note que alguns caracteres ornamentais podem não ser suportados em todos os dispositivos de leitura.

Em geral, você deve evitar o uso de tags HTML, o calibre descartará todas as tags e usará a marcação predefinida. tags, ou seja, regras horizontais, e tags são exceções. As regras horizontais podem opcionalmente ser especificadas com estilos; se você optar por adicionar seu próprio estilo, inclua a configuração "width", caso contrário, as informações de estilo serão descartadas. As tags de imagem podem ser usadas, mas o calibre não fornece a capacidade de adicionar a imagem durante a conversão. Isso deve ser feito após o fato, usando o recurso "Editar livro".

Exemplo de etiqueta de imagem (coloca a imagem numa pasta "Images" dentro do EPUB após a conversão):

Exemplo de linha horizontal com estilos:
<hr style=»width:20%;padding-top: 1px;border-top: 2px ridge black;border-bottom: 2px groove black;»/>

Remover hífenes desnecessários
calibre vai analisar todo o conteúdo hifenizado no documento quando essa opção for ativada. O documento propriamente dito é utilizado como um dicionário para análise. Isto permite ao calibre remover com precisão hífens de qualquer palavra no documento em qualquer idioma, juntamente com palavras científicas obscuras e inventadas. A principal desvantagem são palavras que aparecem apenas uma única vez n o documento que não será alterado. A análise acontece em dois passos, o primeiro passo analisa finais de linha. Linhas só são descobertas, se a palavra existir com ou sem um hífen no documento. O segundo passo analisa todas as palavras com hífen em todo o documento, hifens são removidos se a palavra existe em outras partes do documento sem uma combinação.

:guilabel:"Colocar em itálico palavras e padrões comuns"
Quando ativado, o calibre procurará palavras e padrões comuns que denotam itálico e itálico. Exemplos são convenções de texto comuns, como ~ palavras ~ ou frases que geralmente devem estar em itálico, por exemplo frases latinas como "etc." ou "et cetera".

:guilabel:"Substituir a identação de entidade por CSS"
Alguns documentos usam uma convenção de definição de recuos de texto usando entidades de espaço sem quebra. Quando essa opção está ativada, o calibre tentará detectar esse tipo de formatação e convertê-los em um recuo de texto de 3% usando CSS.

4.5 Pesquisar e substituir

Essas opções são úteis principalmente para a conversão de documentos PDF ou conversões de OCR, embora também possam ser usadas para corrigir muitos problemas específicos de documentos. Como exemplo, algumas conversões podem deixar para trás cabeçalhos e rodapés da página no texto. Essas opções usam expressões regulares para tentar detectar cabeçalhos, rodapés ou outro texto arbitrário e removê-los ou substituí-los. Lembre-se de que eles operam no XHTML intermediário produzido pelo pipeline de conversão. Existe um assistente para ajudá-lo a personalizar as expressões regulares do seu documento. Clique na varinha mágica ao lado da caixa de expressão e clique no botão "Teste" depois de compor sua expressão de pesquisa. As partidas bem-sucedidas serão destacadas em amarelo.

A pesquisa funciona usando uma expressão regular do Python. Todo o texto correspondente é simplesmente removido do documento ou substituído usando o padrão de substituição. O padrão de substituição é opcional, se deixado em branco, o texto correspondente ao padrão de pesquisa será excluído do documento. Você pode aprender mais sobre expressões regulares e sua sintaxe em: ref: *regexptutorial*.

4.6 Deteção de estrutura

Detecção de estrutura envolve a tentativa do calibre dar o seu melhor para detectar elementos estruturais no documento de entrada, quando eles não estão devidamente especificados. Por exemplo, capítulos, quebras de página, cabeçalhos, rodapés, etc. Como você pode imaginar, este processo varia muito de livro para livro. Felizmente, o calibre tem opções muito poderosas para controlar isso. Com o poder vem a complexidade, mas se uma vez que você leva um tempo para aprender a complexidade, você bem que vai achar que vale o esforço.

4.6.1 Capítulos e quebras de página

O calibre possui dois conjuntos de opções para: guilabel: *detecção de capítulo* e: guilabel:` inserção de quebras de página`. Às vezes, isso pode ser um pouco confuso, pois, por padrão, o calibre insere quebras de página antes dos capítulos detectados, bem como os locais detectados pela opção de quebras de página. A razão para isso é que geralmente há um local onde as quebras de página devem ser inseridas que não são limites de capítulos. Além disso, os capítulos detectados podem ser opcionalmente inseridos no Sumário gerado automaticamente.

O calibre usa * XPath **, uma linguagem poderosa para permitir ao usuário especificar limites de capítulos / quebras de página. O XPath pode parecer um pouco assustador de se usar no começo, felizmente, existe: ref: *XPath tutorial* no Manual do Usuário. Lembre-se de que a detecção de estrutura opera no XHTML intermediário produzido pelo pipeline de conversão. Use a opção de depuração descrita em: ref: *conversion-Introduction* para descobrir as configurações apropriadas para o seu livro. Há também um botão para um assistente XPath para ajudar na geração de expressões XPath simples.

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

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

Essa expressão é bastante complexa, porque tenta lidar com vários casos comuns simultaneamente. O que isso significa é que o calibre assumirá que os capítulos começam com `` ou `` tags com qualquer uma das palavras *(capítulo, livro, seção ou parte)* nelas ou que tenham o atributo *class =» chapter «*.

Uma opção relacionada é: guilabel: *Marca do capítulo*, que permite controlar o calibre quando detecta um capítulo. Por padrão, ele inserirá uma quebra de página antes do capítulo. Você pode inserir uma linha ordenada em vez de ou além da quebra de página. Você também pode fazer com que não faça nada.

A configuração predefinida para deteção de quebras de página é:

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

o que significa que o calibre irá inserir a quebras de página antes de cada tag *<h1>* e *<h2>* por padrão.

Nota: As expressões predefinidas podem mudar dependendo do formato de entrada de onde esteja a converter.

4.6.2 Diversos

Existem mais algumas opções nesta secção

:guilabel:"Inserir os metadados como uma página no início do livro"
: Uma das grandes vantagens do calibre é que ele permite que você mantenha metadados muito completos sobre todos os seus livros, por exemplo, uma classificação, tags, comentários etc. Essa opção criará uma única página com todos esses metadados e inseri-los em o e-book convertido, normalmente logo após a capa. Pense nisso como uma maneira de criar sua própria capa de livro personalizada.

: guilabel: *Remover primeira imagem*
: Às vezes, o documento de origem que você está convertendo inclui a capa como parte do livro, em vez de como uma capa separada. Se você também especificar uma capa no calibre, o livro convertido terá duas capas. Essa opção simplesmente remove a primeira imagem do documento de origem, garantindo assim que o livro convertido tenha apenas uma capa, a especificada no calibre.

4.7 Índice

Quando o documento de entrada tem um índice em seus metadados, o calibre apenas o usa. No entanto, vários formatos mais antigos não suportam um índice baseado em metadados ou documentos individuais não o possuem. Nesses casos, as opções nesta seção podem ajudá-lo a gerar automaticamente um Sumário no e-book convertido, com base no conteúdo real no documento de entrada.

Nota: O uso dessas opções pode ser um pouco desafiador para ser exatamente correto. Se você preferir criar / editar o Índice manualmente, converta para os formatos EPUB ou AZW3 e marque a caixa de seleção na parte inferior da seção Índice da caixa de diálogo de conversão que diz: guilabel: *Ajuste manualmente o Índice de Conteúdo após a conversão ». Isso iniciará a ferramenta ToC Editor após a conversão. Ele permite que você crie entradas no Sumário simplesmente clicando no local do livro em que deseja que a entrada aponte. Você também pode usar o ToC Editor por si só, sem fazer uma conversão. Vá para: guilabel: `Preferências-> Interface-> Barras de Ferramentas* e adicione o: guilabel:` ToC Editor` à barra de ferramentas principal. Depois, basta selecionar o livro que deseja editar e clicar no botão: guilabel: *ToC Editor*.

A primeira opção é: guilabel: `Força o uso do índice gerado automaticamente “. Ao marcar esta opção, você pode substituir o calibre de qualquer Índice encontrado nos metadados do documento de entrada pelo gerado automaticamente.

A maneira padrão pela qual a criação do Índice gerado automaticamente funciona é que, o calibre tentará primeiro adicionar os capítulos detectados ao índice gerado. Você pode aprender como personalizar a detecção de capítulos na seção: ref: *structure-detection* acima. Se você não deseja incluir capítulos detectados no índice gerado, marque a opção: guilabel: *Não adicione capítulos detectados*.

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

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

```
Next|Previous
```

As opções: guilabel: *Nível 1,2,3 TOC* permitem criar um sofisticado Índice multinível. São expressões XPath que correspondem a tags no XHTML intermediário produzido pelo pipeline de conversão. Veja o: ref: *conversion-Introduction* para obter acesso a este XHTML. Leia também o: ref: *xpath-tutorial*, para aprender como construir expressões XPath. Ao lado de cada opção há um botão que inicia um assistente para ajudar na criação de expressões XPath básicas. O exemplo simples a seguir ilustra como usar essas opções.

Supostamente você tem um documento de entrada que resulta em um XHTML parecido com isto:

```
<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>
```

(continues on next page)

(continuação da página anterior)

```
        ...
        <h2>Section 2.1</h2>
        ...
    </body>
</html>
```

Depois, definimos as opções como:

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

Isto resultará num índice automático de dois níveis, que será semelhante a:

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

Aviso: Nem todos os formatos de saída suportam um índice multinível. Você deve primeiro tentar com a saída EPUB. Se isso funcionar, tente o seu formato de escolha.

4.8 Usar imagens como títulos de capítulos ao converter documentos de origem em HTML

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

```
<html>
    <body>
        <h2>Chapter 1</h2>
        <p>chapter 1 text...</p>
        <h2 title="Chapter 2"><img src="chapter2.jpg" /></h2>
        <p>chapter 2 text...</p>
    </body>
</html>
```

Defina a configuração: guilabel: *Nível 1 TOC* como` *//h: h2*`. Então, para o capítulo dois, o calibre assumirá o título do valor do atributo `` title`` na tag `` ``, pois a tag não possui texto.

4.9 Usar atributos de etiquetas para fornecer o texto para entradas no índice

Se você possui títulos de capítulo particularmente longos e deseja versões reduzidas no Sumário, pode usar o atributo title para conseguir isso, por exemplo:

```
<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>
```

Defina a configuração: guilabel: *Nível 1 TOC* como` *//h: h2 /@ title*`. Em seguida, o calibre assumirá o título do valor do atributo `` title`` nas tags `` , em vez de usar o texto dentro da tag. Observe o `` / @ title` à direita na expressão XPath, você pode usar este formulário para dizer ao calibre para obter o texto de qualquer atributo que você desejar.

4.10 Modo como as opções são definidas/guardadas para a conversão

Existem dois locais onde as opções de conversão podem ser definidas no calibre. O primeiro está em Preferências-> Conversão. Essas configurações são os padrões para as opções de conversão. Sempre que você tentar converter um novo livro, as configurações definidas aqui serão usadas por padrão.

Você também pode alterar as configurações na caixa de diálogo de conversão para cada conversão de livro. Quando você converte um livro, o calibre lembra as configurações usadas para esse livro, de modo que, se você o converter novamente, as configurações salvas para o livro individual terão precedência sobre os padrões definidos em: guilabel: *Preferências*. Você pode restaurar as configurações individuais aos padrões usando o botão: guilabel: *Restaurar padrões* na caixa de diálogo de conversão de livros individual. Você pode remover as configurações salvas de um grupo de livros selecionando todos os livros e clicando no botão: guilabel: *Editar metadados* para abrir a caixa de diálogo de edição em massa de metadados. Na parte inferior da caixa de diálogo, há uma opção para remover a conversão armazenada. definições.

Quando você converte em massa um conjunto de livros, as configurações são feitas na seguinte ordem (a última vence):

- Das predefinições configuradas em Preferências->Conversão
- Nas configurações de conversão salvas para cada livro que está sendo convertido (se houver). Isso pode ser desativado pela opção no canto superior esquerdo da caixa de diálogo de conversão em massa.
- Das configurações definidas na caixa de diálogo Conversão em lote

Observe que as configurações finais de cada livro em uma conversão em massa serão salvas e reutilizadas se o livro for convertido novamente. Como a prioridade mais alta na conversão em massa é dada às configurações na caixa de diálogo Conversão em massa, elas substituem as configurações específicas de cada livro. Portanto, você deve converter em massa apenas livros que precisam de configurações semelhantes. As exceções são metadados e configurações específicas do formato de entrada. Como a caixa de diálogo de conversão em massa não possui configurações para essas duas categorias, elas serão retiradas das configurações específicas do livro (se houver) ou dos padrões.

Nota: Você pode ver as configurações reais usadas durante qualquer conversão clicando no ícone rotativo no canto inferior direito e clicando duas vezes no trabalho de conversão individual. Isso exibirá um log de conversão que conterá

as configurações reais usadas, na parte superior.

4.11 Dicas específicas do formato

Aqui você encontrará dicas específicas para a conversão de formatos específicos. Opções específicas para um formato específico, se entrada ou saída, estão disponíveis na caixa de diálogo de conversão em sua própria seção, por exemplo, *TXT input* ou` EPUB output`.

4.11.1 Converter documentos de Microsoft Word

O Calibre pode converter automaticamente ficheiros `.docx` criados pelo Microsoft Word 2007 e mais recentes. Basta adicionar o ficheiro ao Calibre e clicar em converter.

Nota: There is a demo .docx file[29] that demonstrates the capabilities of the calibre conversion engine. Just download it and convert it to EPUB or AZW3 to see what calibre can do.

O Calibre irá ger automaticamente um índice com base nos títulos, se você os marcar com os estilos `Título 1`, `Título 2` etc. no Microsoft Word. Abra o ebook de saída no visualizador de ebooks do Calibre e clique no botão: guilabel: *Índice* para visualizar o índice gerado.

Ficheiros .doc antigos

Para arquivos `.doc` mais antigos, é possível salvar o documento como HTML com o *Microsoft Word* e, em seguida, converter o arquivo HTML resultante com o -*Calibre``. Ao salvar como HTML, use a opção «Salvar como página da Web filtrada», pois isso produzirá HTML limpo e com boa conversão. Observe que o Word produz um HTML muito confuso, convertê-lo pode demorar muito tempo, portanto, seja paciente. Se você tiver uma versão mais recente do Word disponível, poderá salvá-la diretamente como `.docx`.

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

4.11.2 Converter documentos de TXT

Os documentos TXT não têm uma maneira bem definida de especificar a formatação, como negrito, itálico, etc., ou a estrutura do documento, como parágrafos, títulos, seções e assim por diante, mas há uma variedade de convenções comumente usadas. Por padrão, o calibre tenta a detecção automática da formatação e marcação corretas com base nessas convenções.

A origem TXT suporta várias opções para diferenciar como os parágrafos são detetados.

: guilabel: ***Estilo de parágrafo: Automático***
Analisa o arquivo de texto e tenta determinar automaticamente como os parágrafos são definidos. Essa opção geralmente funciona bem, se você obtiver resultados indesejáveis, tente uma das opções manuais

: guilabel: ***Estilo de parágrafo: Bloco***
Assume cada linha em branco como um limite de parágrafo:

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

```
This is the first.

This is the
second paragraph.
```

: guilabel: ***Estilo de parágrafo: Único***
Assume que cada linha é um parágrafo:

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

: guilabel: ***Estilo do parágrafo: Imprimir***
Supõe que todo parágrafo comece com um recuo (uma guia ou mais de 2 espaços). Os parágrafos terminam quando a próxima linha que começa com um recuo é atingida

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

  This is the
third.
```

Paragraph style: Unformatted
Supõe que o documento não tenha formatação, mas usa quebras de linha rígida. A pontuação e o comprimento médio da linha são usados para tentar recriar parágrafos.

: guilabel: ***Estilo de formatação: Automático***
Tenta detectar o tipo de marcação de formatação que está sendo usada. Se nenhuma marcação for usada, a formatação heurística será aplicada.

: guilabel: ***Estilo de formatação: Heurístico***
Analisa o documento em busca de títulos de capítulos comuns, quebras de cena e palavras em itálico e aplica a marcação HTML apropriada durante a conversão.

: guilabel: ***Estilo de formatação: Markdown***
O calibre também suporta a entrada TXT em execução através de um pré-processador de transformação conhecido como Markdown. O Markdown permite que a formatação básica seja adicionada aos documentos TXT, como negrito, itálico, títulos de seção, tabelas, listas, um Sumário etc. Marcando os títulos dos capítulos com um número inicial e definindo a expressão de detecção XPath do capítulo como «// h: h1 «é a maneira mais fácil de obter um índice adequado gerado a partir de um documento TXT. Você pode aprender mais sobre a sintaxe do Markdown em *daringfireball <https://daringfireball.net/projects/markdown/syntax>* _.

: guilabel: ***Estilo de formatação: Nenhum***
Não aplica qualquer formatação especial ao texto, o documento é convertido para HTML sem alterações adicionais.

4.11.3 Converter documentos de PDF

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

Also, they often have headers and footers as part of the document that will become included with the text. Use the *Search and replace* panel to remove headers and footers to mitigate this issue. If the headers and footers are not removed from the text it can throw off the paragraph unwrapping. To learn how to use the header and footer removal options, read *Tudo sobre utilização de expressões regulares no Calibre* (página 208).

Algumas limitações de entradas em PDF:

- Documentos complexos, com multi-colunas ou os baseados em imagens não são suportados.
- Extração de imagens vetorizadas e tabelas de dentro do documento também não são suportadas.
- Alguns PDFs usam glifos especiais para representar ll ou ff ou fi, etc. A conversão destes pode ou não funcionar, dependendo de como eles são representados internamente no PDF.
- Hiperligações e índices não são suportados
- PDFs that use embedded non-Unicode fonts to represent non-English characters will result in garbled output for those characters
- Alguns PDFs são compostos de fotografias da página com texto OCR por trás deles. Nesses casos, o calibre usa o texto OCR, que pode ser muito diferente do que você vê ao exibir o arquivo PDF
- PDFs usados para exibir texto complexo, como idiomas da direita para a esquerda e tipografia matemática, não serão convertidos corretamente

Reiterar ** PDF é um formato muito, muito ruim ** para ser usado como entrada. Se você precisar absolutamente usar o PDF, esteja preparado para uma saída que varia de decente a inutilizável, dependendo do PDF de entrada.

4.11.4 Coleções de novelas gráficas

Uma coleção de quadrinhos é um arquivo .cbc. Um arquivo .cbc é um arquivo ZIP que contém outros arquivos CBZ / CBR. Além disso, o arquivo .cbc deve conter um arquivo de texto simples chamado comics.txt, codificado em UTF-8. O arquivo comics.txt deve conter uma lista dos arquivos de quadrinhos dentro do arquivo .cbc, no formato filename: title, como mostrado abaixo:

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

O ficheiro .cbc conterá:

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

O calibre converterá automaticamente esse arquivo .cbc em um e-book com um Índice apontando para cada entrada em comics.txt.

4.11.5 Formatação avançada de EPUB - Demo

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

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

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

Observe que, como esse arquivo explora o potencial do EPUB, a maior parte da formatação avançada não funciona em leitores menos capazes que o visualizador EPUB integrado do calibre.

4.11.6 Converter ficheiros ODT

O calibre pode converter diretamente arquivos ODT (OpenDocument Text). Você deve usar estilos para formatar seu documento e minimizar o uso da formatação direta. Ao inserir imagens no documento, você precisa ancorá-las ao parágrafo, as imagens ancoradas em uma página acabam na frente da conversão.

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

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

As propriedades conhecidas do documento (Título, Palavras-chave, Descrição, Criador) são reconhecidas e o calibre usará a primeira imagem (não pequena, e com boa proporção) como imagem de capa.

Há também um modo avançado de conversão de propriedades, que é ativado ao definir a propriedade personalizada `` opf.metadata`` (tipo "Sim ou Não") como Sim no seu documento ODT (Arquivo-> Propriedades-> Propriedades Personalizadas). Se essa propriedade for detectada pelo calibre, as seguintes propriedades personalizadas serão reconhecidas (`` opf.authors`` substitui o criador do documento)

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

Além disso, você pode especificar a imagem a ser usada como capa, nomeando-a como `` opf.cover`` (clique com o botão direito do mouse em Imagem-> Opções-> Nome) na ODT. Se nenhuma imagem com esse nome for encontrada, o método "inteligente" será usado. Como a detecção de capa pode resultar em capas duplas em determinados formatos de saída, o processo removerá o parágrafo (somente se o único conteúdo for a capa!) Do documento. Mas isso funciona apenas com a imagem nomeada!

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

Para desativar a detecção de capa, você pode definir a propriedade personalizada `` of.no cover`` (tipo “Yes or No”) para Yes no modo avançado.

4.11.7 Converter para PDF

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

Cabeçalhos e rodapés

Você pode inserir cabeçalhos e rodapés arbitrários em cada página do PDF especificando modelos de cabeçalho e rodapé. Modelos são apenas trechos de código HTML que são renderizados nos locais de cabeçalho e rodapé. Por exemplo, para exibir os números de página centralizados na parte inferior de cada página, em verde, use o seguinte modelo de rodapé:

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

O calibre substituirá automaticamente: code: *_PAGENUM_* pelo número da página atual. Você pode até colocar conteúdo diferente em páginas pares e ímpares, por exemplo, o seguinte modelo de cabeçalho mostrará o título em páginas ímpares e o autor em páginas pares

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

O calibre substituirá automaticamente: code: *_TITLE_* e: code:` _AUTHOR_` pelo título e autor do documento que está sendo convertido. A configuração: code: *justify-content* to: code:` flex-end` fará com que o texto fique alinhado à direita.

Você também pode exibir o texto nas bordas esquerda e direita e alterar o tamanho da fonte, conforme demonstrado neste modelo de cabeçalho:

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

Isto exibirá o título à esquerda e o autor à direita, num tamanho de fonte menor que o texto principal.

Também pode usar a seção atual nos modelos, tal como mostrado abaixo:

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

: code: *_SECTION_* é substituído por qualquer que seja o nome da seção atual. Esses nomes são retirados do índice de metadados do documento (o resumo em PDF). Se o documento não tiver um índice, ele será substituído pelo texto vazio. Se uma única página PDF tiver várias seções, a primeira seção da página será usada. Da mesma forma, existe uma variável chamada: code: *_TOP_LEVEL_SECTION_* que pode ser usada para obter o nome da seção atual de nível superior.

Você pode até usar JavaScript dentro dos modelos de cabeçalho e rodapé, por exemplo, o modelo a seguir fará com que os números de página sejam iniciados em 4 em vez de 1

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

Além disso, existem mais algumas variáveis que é possível usar nos cabeçalhos e rodapés, documentadas abaixo:

- `_TOTAL_PAGES_` - número total de páginas no arquivo PDF, útil para implementar um contador de progresso, por exemplo.
- `_TOP_LEVEL_SECTION_PAGES_` - número total de páginas na seção de nível superior atual
- `_TOP_LEVEL_SECTION_PAGENUM_` - o número da página atual dentro da seção de nível superior atual

---

Nota: Ao adicionar cabeçalhos e rodapés, defina as margens superior e inferior da página com valores suficientemente grandes, na seção: guilabel: *PDF Output* da caixa de diálogo de conversão.

Índice imprimível

Você também pode inserir um Índice imprimível no final do PDF que lista os números de página para cada seção. Isso é muito útil se você pretende imprimir o PDF em papel. Se você deseja usar o PDF em um dispositivo eletrônico, o Esboço do PDF fornece essa funcionalidade e é gerado por padrão.

É possível personalizar a aparência do Sumário gerado usando a configuração de conversão CSS extra na parte Aparência da caixa de diálogo de conversão. O CSS padrão usado está listado abaixo, basta copiá-lo e fazer as alterações desejadas.

```
.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 }
```

Margens de página personalizadas para arquivos HTML individuais

Se você estiver convertendo um arquivo EPUB ou AZW3 com vários arquivos HTML individuais dentro dele e quiser alterar as margens da página para um arquivo HTML específico, é possível adicionar o seguinte bloco de estilo ao arquivo HTML usando o editor de E-book do calibre:

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

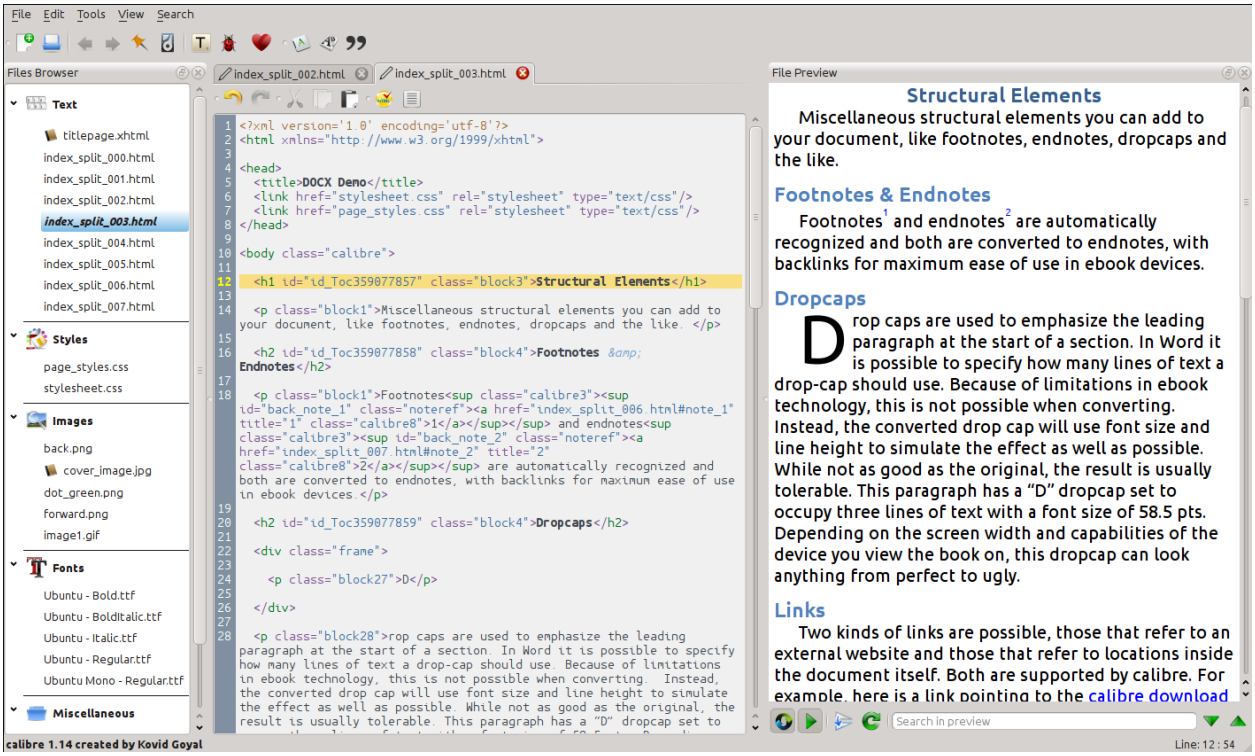
Em seguida, na seção de saída em PDF da caixa de diálogo de conversão, ative a opção para: guilabel: *Use as margens da página do documento que está sendo convertido*. Agora todas as páginas geradas a partir deste arquivo HTML terão margens `` 10pt``.

CAPÍTULO 5

Editar livros digitais

O Calibre tem integrado um editor de ebooks que pode ser usado para editar livros nos formatos EPUB e AZW3 (Kindle). O editor mostra-lhe o HTML e o CSS que são usados internamente dentro dos ficheiros do livro, com uma pré-visualização ao vivo que é atualizada conforme vai fazendo alterações. Também contém várias ferramentas automatizadas para efetuar tarefas comuns de limpeza e reparação.

Pode usar este editor clicando com o botão direito em qualquer livro no Calibre e selecionar *Editar livro*.



Conteúdos

- *Fluxo de trabalho básico* (página 77)
- *O navegador de ficheiros* (página 79)
  - *Renomear ficheiros* (página 80)
  - *Fundir ficheiros* (página 80)
  - *Mudar ordem dos ficheiros de texto* (página 81)
  - *Marcar a capa* (página 81)
  - *Apagar ficheiros* (página 81)
  - *Exportar ficheiros* (página 81)
  - *Adicionar novas imagens/tipos de letra/etc. ou criar novos ficheiros vazios* (página 81)
  - *Substituir ficheiros* (página 82)
  - *Ligar eficientemente folhas de estilo a ficheiros HTML* (página 82)
- *Pesquisar e substituir* (página 82)
  - *Pesquisas guardadas* (página 82)
  - *Modo de função* (página 83)
  - *Pesquisar ignorando etiquetas HTML* (página 83)
- *Ferramentas automatizadas* (página 83)
  - *Editar o índice* (página 83)
  - *A verificar o livro* (página 84)
  - *Adicionar capa* (página 85)
  - *A incorporar os tipos de letra referenciados* (página 85)
  - *A segmentar os tipos de letra incorporados…* (página 85)
  - *Em processamento inteligente da pontuação* (página 86)
  - *Transformar as propriedades CSS* (página 86)
  - *A remover regras CSS não utilizadas* (página 86)
  - *Corrigir HTML* (página 86)
  - *A embelezar ficheiros* (página 86)
  - *A inserir um índice incorporado* (página 87)
  - *Definir semântica* (página 87)
  - *Filtrar estilo de informação* (página 87)
  - *Atualização das estruturas internas do livro* (página 87)
- *Pontos de controlo* (página 87)
- *O painel de pré-visualização em tempo real* (página 89)
  - *Dividir ficheiros HTML* (página 90)
- *Painel CSS ao vivo* (página 91)

- *Ferramentas diversas* (página 92)
  - *A vista de índice* (página 92)
  - *Verificar a ortografia de palavras no livro* (página 92)
  - *Inserir caracteres especiais* (página 94)
  - *Vista de inspeção de código* (página 95)
  - *Verificar ligações externas* (página 95)
  - *Descarregar recursos externos* (página 95)
  - *Organizar ficheiros em pastas por tipo* (página 95)
  - *Importing files in other e-book formats as EPUB* (página 95)
  - *A ferramenta de relatórios* (página 106)
- *Características especiais no editor de código* (página 106)
  - *Realçar sintaxe* (página 106)
  - *Ajuda de conteúdo sensível* (página 107)
  - *Auto-completar* (página 107)
  - *Trechos* (página 107)

5.1 Fluxo de trabalho básico

Nota: Está disponível um vídeo contendo uma visita guiada ao editor de ebooks do Calibre aqui[32].

Quando abre pela primeira vez um livro com a ferramenta de Edição de livro, ser-lhe-á apresentada uma lista de ficheiros à esquerda. Estes são os ficheiros HTML individuais, estilos, imagens, etc. que compõem o conteúdo do livro. Faça simplesmente duplo clique num ficheiro para iniciar a sua edição. Note que se pretende fazer algo mais sofisticado do que algumas pequenas alterações, necessita de estar familiarizado com o Tutorial de HTML[33] e o Tutorial de CSS[34].

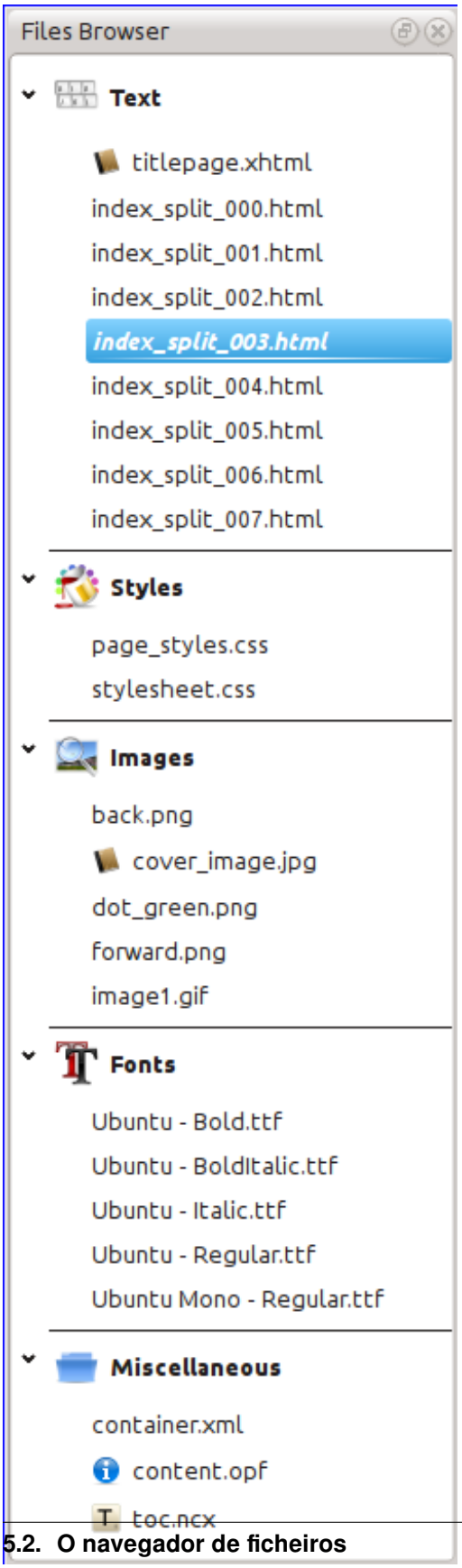
Conforme vai fazendo alterações ao HTML ou ao CSS no editor, as mudanças serão pré-visualizadas, no momento, no painel de pré-visualização à direita. Quando estiver satisfeito com o modo como se apresentam as mudanças que efetuou, clique no botão *Guardar* ou use *Ficheiro → Guardar* para guardar as suas alterações no ebook.

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

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

[32] https://calibre-ebook.com/pt/demo#tutorials
[33] http://html.net/tutorials/html/
[34] http://html.net/tutorials/css/

5.2 O navegador de ficheiros



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.

Ao pairar o rato sobre uma entrada, pode ver o seu tamanho e, também, no fundo do ecrã, o caminho completo para o ficheiro dentro do livro. Note que os ficheiros dentro de ebooks estão compactados, pelo que o tamanho do livro finalizado não é a soma dos tamanhos dos ficheiros individuais.

Muitos ficheiros têm, no livro, significado especial. Estes tipicamente têm um ícone junto aos seus nomes, indicando esse significado especial. Por exemplo, na figura à esquerda, pode ver que os ficheiros *cover_image.jpg* e *titlepage.xhtml* têm um ícone de uma capa junto a eles, que indica que são a capa e a página de título do livro. De igual modo, o ficheiro *content.opf* tem um ícone de metadados junto a ele, indicando que os meta-dados do livro estão presentes e o ficheiro *toc.ncx* tem um ícone T a seguir a ele indicando que é o Índice.

Pode executar muitas ações em ficheiros individuais, clicando sobre eles com o botão direito do rato.

5.2.1 Renomear ficheiros

Pode renomear um ficheiro individual clicando com o botão direito do rato e selecionando *Renomear*. Renomear um ficheiro atualiza automaticamente todos as hiperligações e referências a ele existentes no livro. Assim, tudo o que tem de fazer é fornecer o novo nome, e o Calibre toma conta do resto

Pode também renomear vários ficheiros de uma vez só. Tal é útil se pretender que os ficheiros apresentem um qualquer tipo de padrão simples no nome. Por exemplo, pode querer renomear todos os ficheiros HTML de modo a ter nomes como Capitulo-1.html, Capitulo-2.html e assim sucessivamente. Selecione os ficheiros a que pretende aplicar esta renomeação múltipla mantendo carregada a tecla `Shift` ou `Ctrl` enquanto clica nos ficheiros. De seguida, clique com o botão direito e selecione *Renomear em lote*. Indique um prefixo e o número com que pretende começar a numeração automática, clique OK, e está feito. O diálogo de renomeação em lote permite-lhe ainda renomear ficheiros pela ordem em que aparecem no livro em vez de ser pela ordem em que os escolheu, o que é útil, por exemplo, para renomear todas as figuras pela ordem em que aparecem.

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

5.2.2 Fundir ficheiros

Às vezes pode pretender combinar dois ficheiros HTML ou dois ficheiros CSS. Pode eventualmente ser útil ter tudo apenas num só ficheiro. No entanto, tenha cuidado, pois colocar uma grande quantidade de conteúdos num só ficheiro terá impacto na performance do sistema ao visualizar o livro num leitor de ebooks.

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

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

5.2.3 Mudar ordem dos ficheiros de texto

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

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

5.2.4 Marcar a capa

Os ebooks tipicamente têm uma imagem de capa. Esta imagem é indicada no *Navegador de ficheiros* pelo ícone de um livro castanho junto ao nome da imagem. Se pretender designar outra imagem como sendo a capa, pode fazê-lo clicando com o botão direito sobre o ficheiro e escolher *Marcar como capa*.

Além disso, os arquivos EPUB têm o conceito de uma *titlepage*. Uma página de capa é um ficheiro HTML que atua como a página de título / capa do livro. Você pode marcar um ficheiro HTML como sendo a página de título ao editar EPUBs clicando com o botão direito do rato. Cuidado para que o ficheiro marcado contenha apenas as informações da capa. Se contiver outro conteúdo, como o primeiro capítulo, esse conteúdo será perdido caso o utilizador converta posteriormente o ficheiro EPUB no Calibre para outro formato. Isto ocorre porque, ao converter, o Calibre pressupõe que a página de capa marcada contenha apenas a capa e nenhum outro conteúdo.

5.2.5 Apagar ficheiros

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

5.2.6 Exportar ficheiros

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

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

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

5.2.7 Adicionar novas imagens/tipos de letra/etc. ou criar novos ficheiros vazios

Pode adicionar uma nova imagem, tipo de letra, folha de estilo, etc. a partir do seu computador para o livro clicando *Ficheiro → Novo ficheiro*. Tal permite importar um ficheiro quando clica no botão *Importar ficheiros de recursos*, ou criar um novo ficheiro HTML ou folha de estilo vazios, bastando para tal simplesmente indicar um nome de ficheiro na caixa para o novo ficheiro.

Pode também importar múltiplos ficheiros de uma só vez para dentro do livro usando Ficheiro->Importar ficheiros para dentro do livro.

5.2.8 Substituir ficheiros

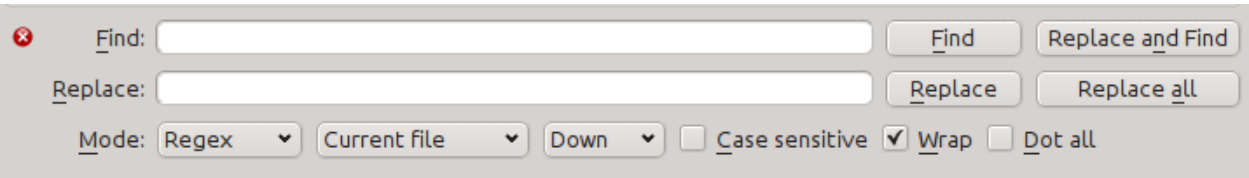
You can easily replace existing files in the book, by right clicking on the file and choosing replace. This will automatically update all links and references, in case the replacement file has a different name than the file being replaced.

5.2.9 Ligar eficientemente folhas de estilo a ficheiros HTML

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

5.3 Pesquisar e substituir

Edit book has a very powerful search and replace interface that allows you to search and replace text in the current file, across all files and even in a marked region of the current file. You can search using a normal search or using regular expressions. To learn how to use regular expressions for advanced searching, see *Tudo sobre utilização de expressões regulares no Calibre* (página 208).



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

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

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

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

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

Nota: Remember, to harness the full power of search and replace, you will need to use regular expressions. See *Tudo sobre utilização de expressões regulares no Calibre* (página 208).

5.3.1 Pesquisas guardadas

Pode guardar expressões de pesquisa/substituição frequentemente utilizadas (incluindo expressões de modo de função) e reutilizá-las várias vezes. Para guardar uma pesquisa basta clicar com o botão direito do rato na caixa Localizar e selecionar *Guardar pesquisa atual*.

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

5.3.2 Modo de função

Function mode allows you to write arbitrarily powerful Python functions that are run on every Find/replace. You can do pretty much any text manipulation you like in function mode. For more information, see *Modo de função para Procurar & Substituir no Editor* (página 95).

5.3.3 Pesquisar ignorando etiquetas HTML

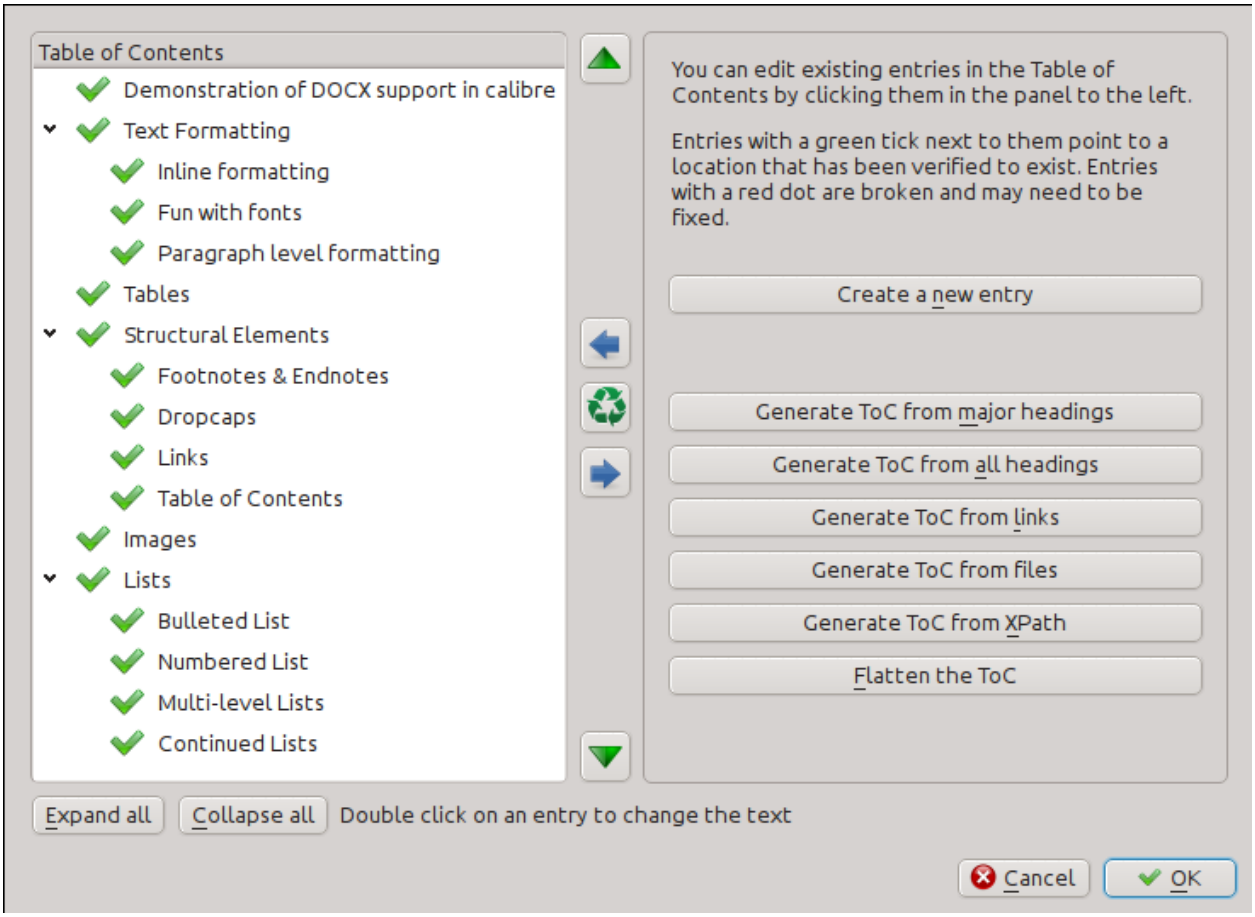
There is also a dedicated tool for searching for text, ignoring any HTML tags in between. For example, if the book has the HTML `Empahisis on a <i>word</i>.` you can search for `on a word` and it will be found even though there is an `<i>` tag in the middle. Use this tool via the *Search → Search ignoring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

5.4 Ferramentas automatizadas

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

5.4.1 Editar o índice

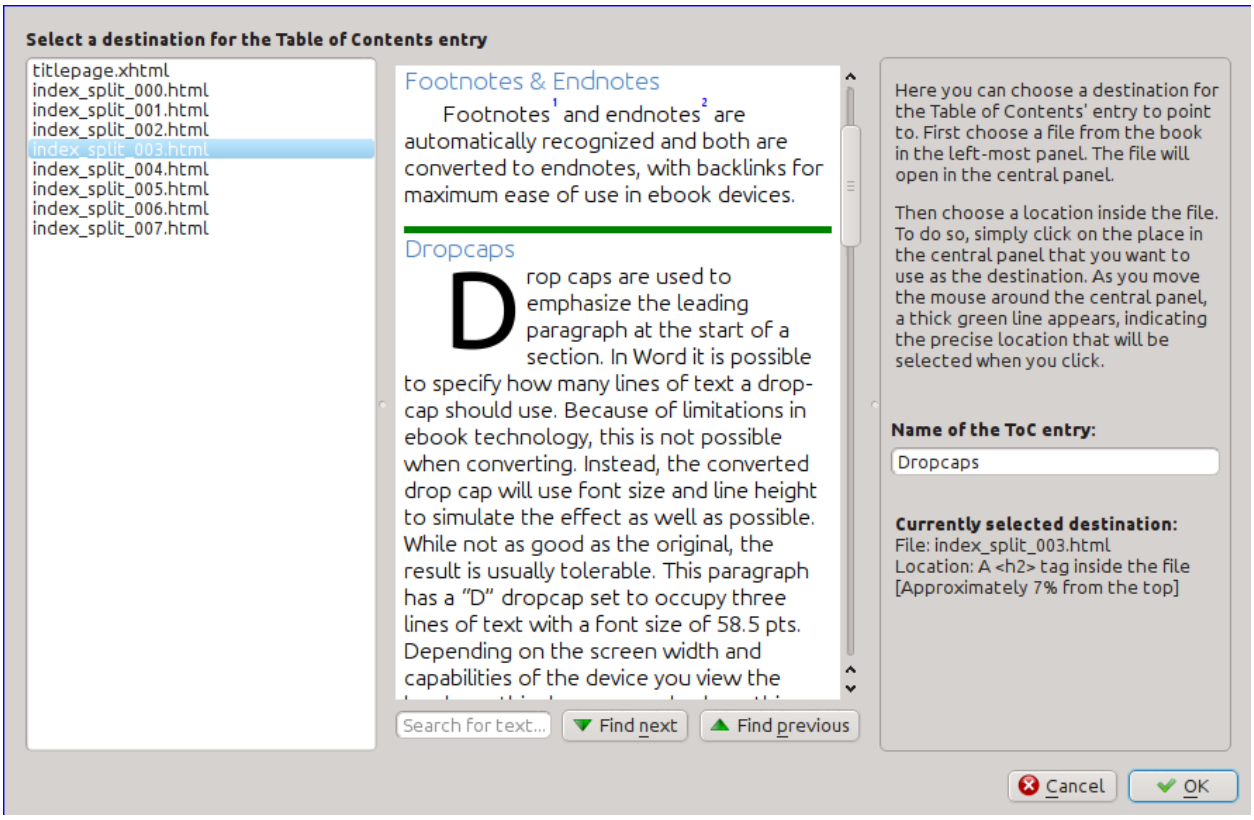
Existe uma ferramenta dedicada para facilitar a edição do índice. Invoque-a com *Ferramentas → Índice → Editar índice*.



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

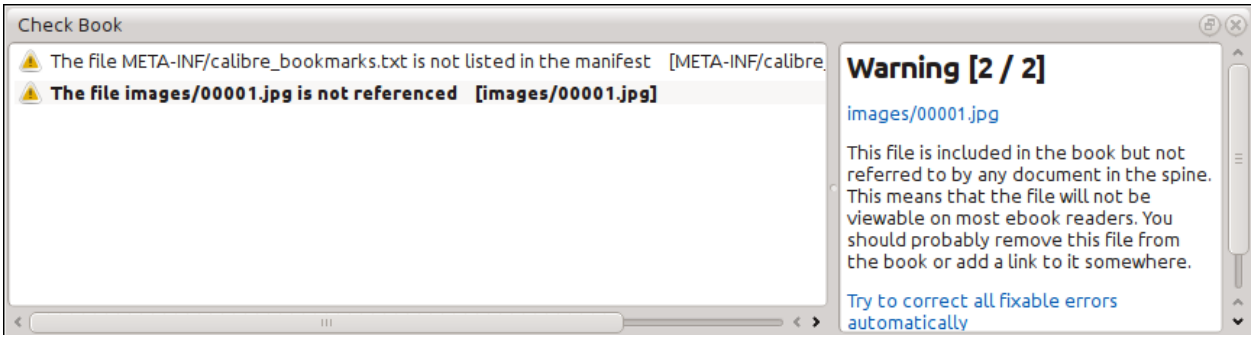
For books that do not have a pre-existing Table of Contents, the tool gives you various options to auto-generate a Table of Contents from the text. You can generate from the headings in the document, from links, from individual files and so on.

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



5.4.2 A verificar o livro

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



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

Algumas das verificações executadas:

- Malformed HTML markup. Any HTML markup that does not parse as well-formed XML is reported. Correcting it will ensure that your markup works as intended in all contexts. calibre can also auto-fix these errors, but auto-fixing can sometimes have unexpected effects, so use with care. As always, a checkpoint is created before auto-fixing so you can easily revert all changes. Auto-fixing works by parsing the markup using the HTML5 algorithm, which is highly fault tolerant and then converting to well formed XML.
- Estilos CSS malformados ou desconhecidos. Qualquer CSS que não seja válido ou que tenha propriedades não definidas na norma CSS 2.1 (e mais alguns do CSS 3) são relatados. O CSS é verificado em todas as folhas de estilo, atributos de estilo em linha e <style> etiquetas em ficheiros HTML.
- Hiperligações quebradas. São relatadas ligações que apontem para ficheiros dentro do livro e que estejam em falta.
- Ficheiros não referenciados. São relatados ficheiros no livro que não são referenciados por qualquer outro ficheiro ou que não constem da espinha.
- Vários problemas comuns em ficheiros OPF tais como lombada ou itens de manifesto duplicados, problemas com idrefs ou metaetiquetas de capa, secções obrigatórias em falta e assim por diante.
- Verificações diversas de compatibilidade para problemas conhecidos que podem provocar o mau funcionamento do livro em dispositivos de leitura.

5.4.3 Adicionar capa

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

5.4.4 A incorporar os tipos de letra referenciados

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

5.4.5 A segmentar os tipos de letra incorporados...

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

5.4.6 Em processamento inteligente da pontuação

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

5.4.7 Transformar as propriedades CSS

Cria regras que transformam o estilo do livro. Por exemplo, criar uma regra para converter todo o texto vermelho para verde ou duplicar o tamanho do tipo de letra em todo o livro ou ainda colocar em itálico o texto de uma certa família de tipos de letra, etc.

Criar regras é simples, pois seguem um formato de linguagem natural que se parece com:

- Se a propriedade *cor* for *vermelho alterar* para *verde*
- Se a propriedade *font-size* for *qualquer valor multiplicar* o valor por *2*

Acedido via *Ferramentas → Transformar estilos*.

5.4.8 A remover regras CSS não utilizadas

Remove todas as regras CSS não usadas das folhas de estilo e das marcas <style>. Alguns livros criados a partir de modelos de produção podem conter um grande número de regras CSS extra que não se aplicam a qualquer conteúdo existente. Estas regras extra podem atrasar leitores que necessitem de as processar a todas. É acedido via *Ferramentas → Remover CSS não usado*.

5.4.9 Corrigir HTML

Esta ferramenta converte HTML que não pode ser analisado como XML em XML bem formado. É muito comum nos e-books existir XML mal-formado, pelo que esta ferramenta simplesmente automatiza o processo de reparação desse HTML. A ferramenta funciona analisando o HTML usando o algoritmo HTML5 (o algoritmo usado em todos os browsers modernos) e depois convertendo o resultado para XML. Esteja ciente de que a reparação automática pode, por vezes, ter resultados contraintuitivos. Se preferir, pode usar a ferramenta Verificação de livro, discutida acima, para encontrar e corrigir manualmente problemas no HTML. Aceder através de *Ferramentas → Reparar HTML*.

5.4.10 A embelezar ficheiros

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

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

---

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

5.4.11 A inserir um índice incorporado

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

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

5.4.12 Definir semântica

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

5.4.13 Filtrar estilo de informação

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

5.4.14 Atualização das estruturas internas do livro

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

5.5 Pontos de controlo

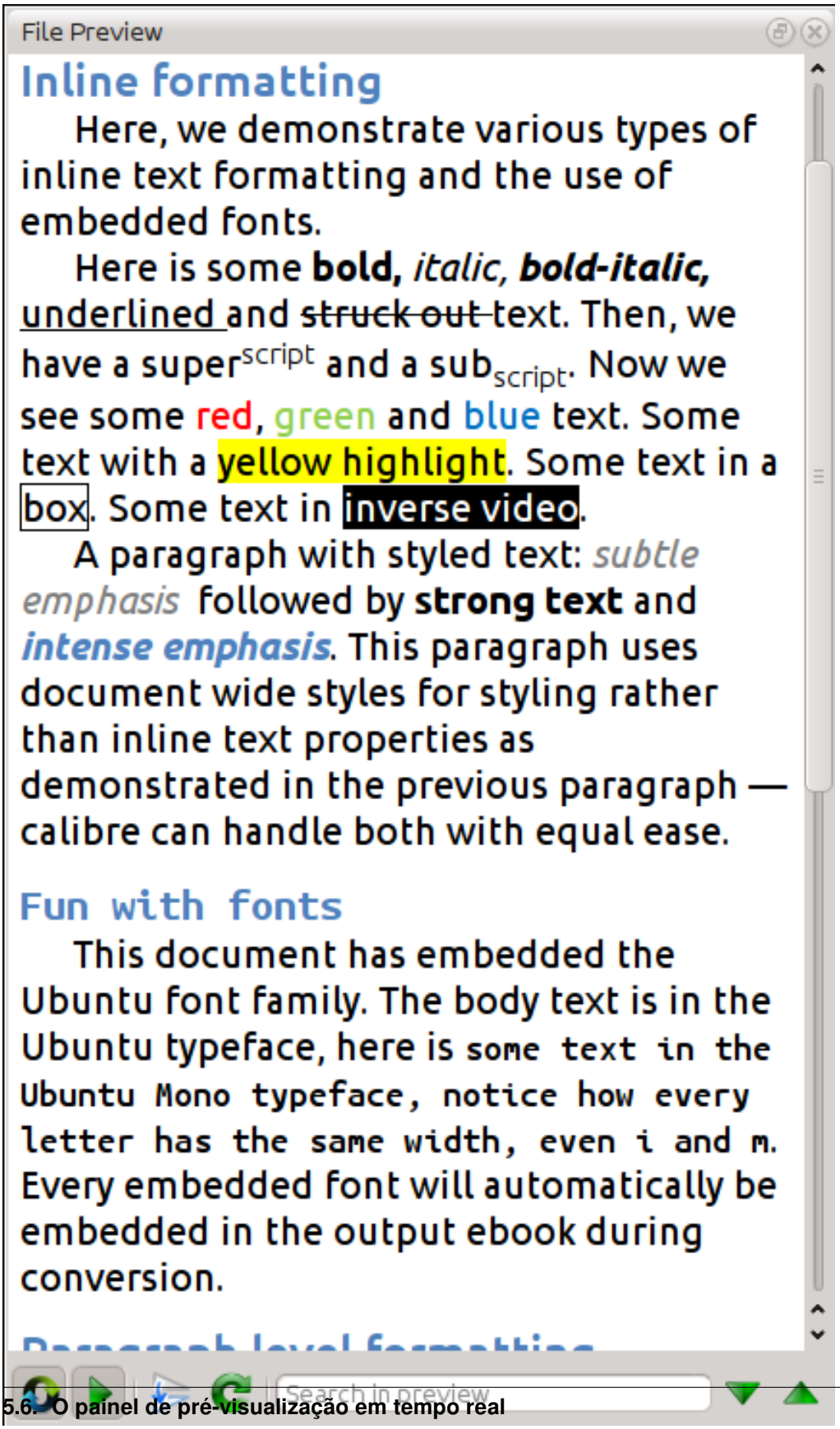
Checkpoints are a way to mark the current state of the book as «special». You can then go on to do whatever changes you want to the book and if you don't like the results, return to the checkpointed state. Checkpoints are automatically created every time you run any of the automated tools described in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

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

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

You can see a list of available checkpoints via *View → Checkpoints*. You can compare the current state of the book to a specified checkpoint using the *Comparar livros digitais* (página 117) tool – by selecting the checkpoint of interest and clicking the *Compare* button. The *Revert to* button restores the book to the selected checkpoint, undoing all changes since that checkpoint was created.

5.6 O painel de pré-visualização em tempo real



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

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

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

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

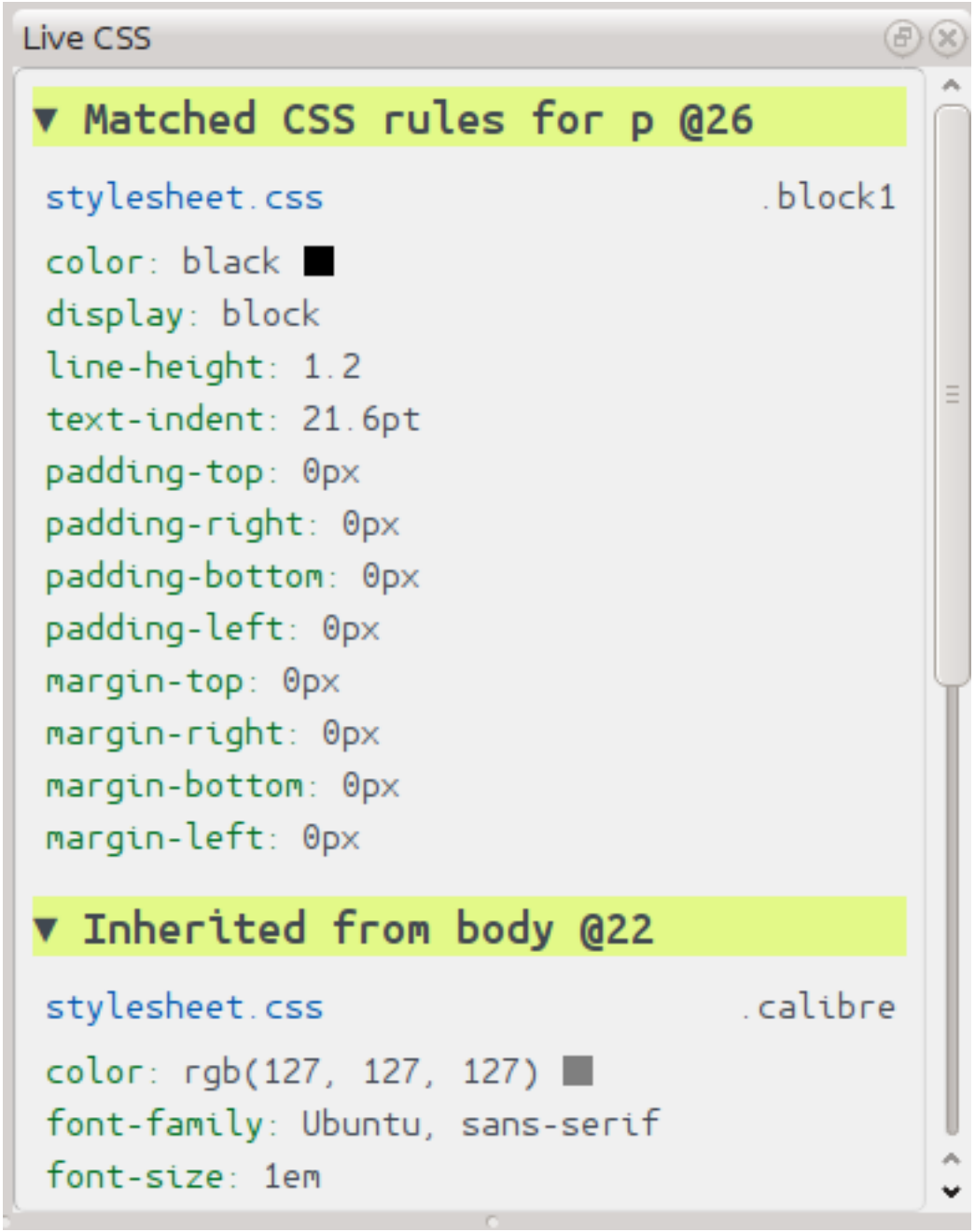
5.6.1 Dividir ficheiros HTML

One, perhaps non-obvious, use of the preview panel is to split long HTML files. While viewing the file you want to split, click the *Split mode* button under the preview panel . Then simply move your mouse to the place where you want to split the file and click. A thick green line will show you exactly where the split will happen as you move your mouse. Once you have found the location you want, simply click and the split will be performed.

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

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

5.7 Painel CSS ao vivo



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

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

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

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

5.8 Ferramentas diversas

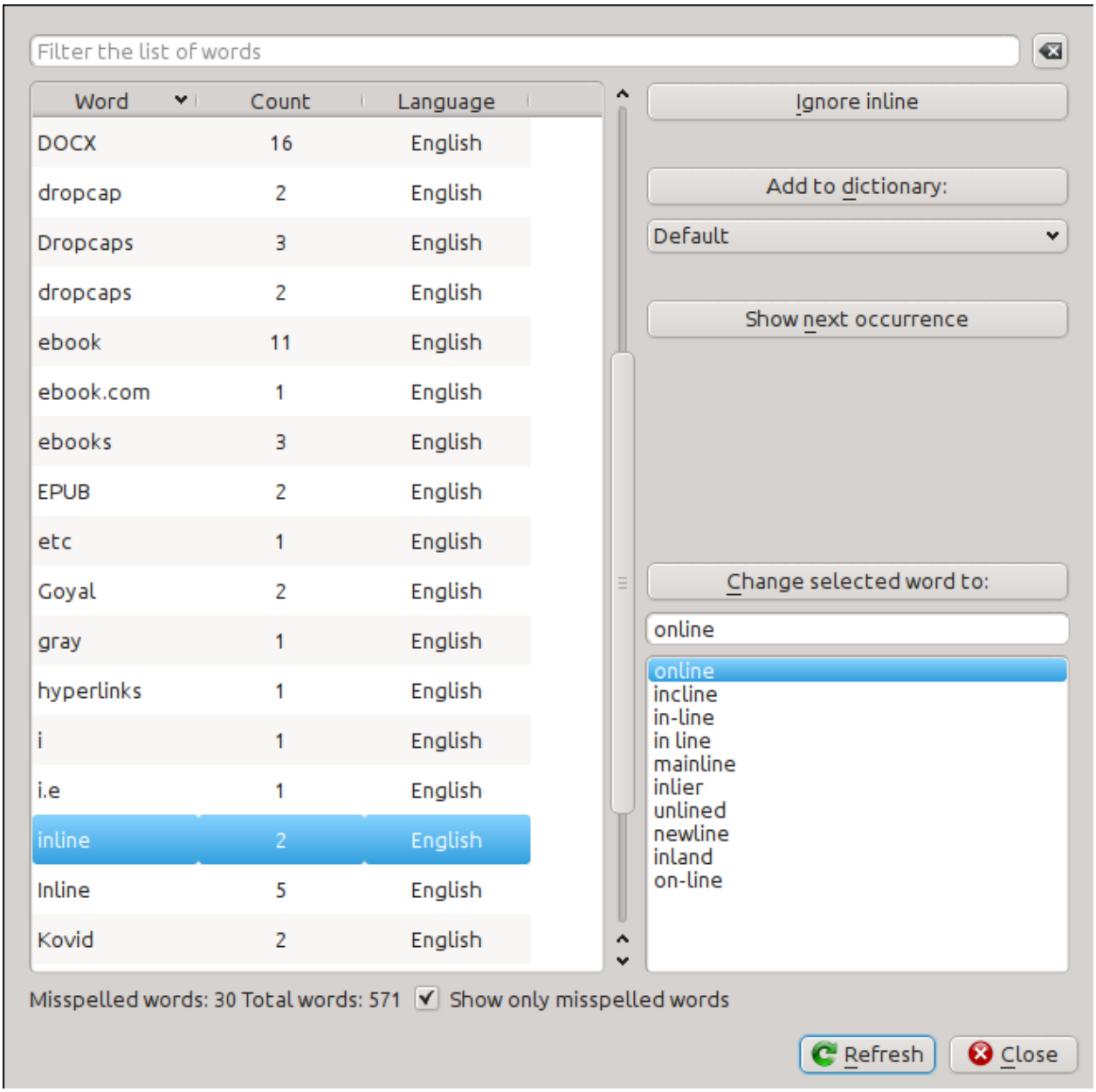
Existem mais algumas ferramentas que podem ser úteis quando editar o livro.

5.8.1 A vista de índice

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

5.8.2 Verificar a ortografia de palavras no livro

Pode correr um corretor ortográfico através de *Ferramentas → Verificar ortografia*.



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

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

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

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

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

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

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

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

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

Adicionar dicionários

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

5.8.3 Inserir caracteres especiais

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

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

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

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

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

5.8.4 Vista de inspeção de código

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

5.8.5 Verificar ligações externas

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

5.8.6 Descarregar recursos externos

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

5.8.7 Organizar ficheiros em pastas por tipo

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

5.8.8 Importing files in other e-book formats as EPUB

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

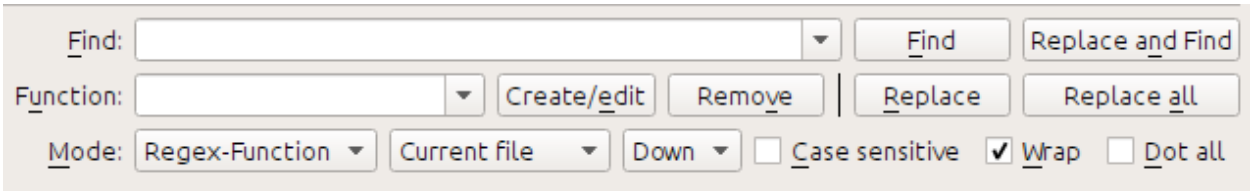
Modo de função para Procurar & Substituir no Editor

The *Search & replace* tool in the editor support a *function mode*. In this mode, you can combine regular expressions (see *Tudo sobre utilização de expressões regulares no Calibre* (página 208)) with arbitrarily powerful Python functions to do all sorts of advanced text processing.

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

Techniques for using function mode and the syntax will be described by means of examples, showing you how to create functions to perform progressively more complex tasks.

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



A reparar automaticamente a maiusculização dos cabeçalhos no documento.

Aqui, potenciamos uma das funções implementadas no editor que modifica automaticamente a maiusculização de todo o texto contido nas etiquetas HTML de cabeçalho para a maiusculização de título:

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

Para a função, basta escolher a função predefinida *Maiusculização de cabeçalhos (ignorar etiquetas)*. Isto irá alterar títulos que sejam parecidos com: `<h1>CABEÇALHO exemplo</h1>` para `<h1>Cabeçalho Exemplo</h1>`. Tal funcionará mesmo que haja outras etiquetas HTML dentro das etiquetas Heading.

A sua primeira função personalizada - hífens inteligentes

O verdadeiro poder do modo de função advém do facto de lhe permitir criar as suas próprias funções para processar texto de modo arbitrário. A ferramenta do editor de Pontuação inteligente deixa inalterados hífens individuais, pelo que pode usar a função this para os substituir por travessões.

Para criar uma nova função, simplesmente clique no botão *Criar/editar* para criar uma nova função e copiar o código Python que se encontra abaixo.

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

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

Use esta função com a expressão regular de pesquisa:

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

E irá substituir todos os hifenes por travessões, mas apenas no texto e não dentro das definições das etiquetas HTML.

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

O poder do modo de função - usando um dicionário ortográfico para corrigir palavras incorretamente hifenizadas.

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

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

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

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

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

Use this function with the same find expression as before, namely:

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

And it will magically fix all mis-hyphenated words in the text of the book. The main trick is to use one of the useful extra arguments to the replace function, `dictionaries`. This refers to the dictionaries the editor itself uses to spell check text in the book. What this function does is look for words separated by a hyphen, remove the hyphen and check if the dictionary recognizes the composite word, if it does, the original words are replaced by the hyphen free composite word.

Note that one limitation of this technique is it will only work for mono-lingual books, because, by default, `dictionaries.recognized()` uses the main language of the book.

Secções autonumeráveis

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

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

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

Usar com a expressão de pesquisa:

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

Posicione o cursor no topo do ficheiro e clique *Substituir todos*.

This function uses another of the useful extra arguments to `replace()`: the `number` argument. When doing a *Replace All* number is automatically incremented for every successive match.

Another new feature is the use of `replace.file_order` – setting that to `'spine'` means that if this search is run on multiple HTML files, the files are processed in the order in which they appear in the book. See *Escolha a ordem dos ficheiros ao executar em múltiplos ficheiros HTML* (página 101) for details.

Criar automaticamente um índice

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

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

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

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

(continues on next page)

(continuação da página anterior)

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

E use-o com a expressão de pesquisa:

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

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

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

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

Isto foi apenas uma demonstração para lhe mostrar o poder do modo de função. Se tivesse realmente a necessidade de gerar um índice a partir dos cabeçalhos do seu livro, conseguiria melhores resultados usando a ferramenta dedicada de índice em *Ferramentas → Índice*.

The API for the function mode

Todas as funções do modo de função têm de ser funções codificadas em Python, ter o nome replace, e a seguinte assinatura:

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

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

The `match` argument

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

The `number` argument

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

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

The `file_name` argument

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

The `metadata` argument

This represents the metadata of the current book, such as title, authors, language, etc. It is an object of class *calibre.ebooks.metadata.book.base.Metadata* (página 204). Useful attributes include, `title`, `authors` (a list of authors) and `language` (the language code).

O argumento `dictionaries`

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

O argumento `data`

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

O argumento `functions`

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

```
def utility():
    # do something

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

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

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

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

```
# Function One
persistent_data = {}
```

(continues on next page)

(continuação da página anterior)

```
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']
    ...
```

Depurar as suas funções

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

Escolha a ordem dos ficheiros ao executar em múltiplos ficheiros HTML

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

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

replace.file_order = 'spine'
```

`file_order` accepts two values, `spine` and `spine-reverse` which cause the search to process multiple files in the order they appear in the book, either forwards or backwards, respectively.

Having your function called an extra time after the last match is found

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

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

replace.call_after_last_match = True
```

Appending the output from the function to marked text

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

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

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

Suppressing the result dialog when performing searches on marked text

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

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

replace.suppress_result_dialog = True
```

Mais exemplos

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

Trechos

O editor Calibre de ebook suporta *trechos*. Um trecho é um pedaço de texto que é reutilizado com frequência ou contém algum texto redundante. O editor permite que você insira um trecho com apenas algumas combinações de teclas. Por exemplo, suponha que você insira frequentemente etiquetas de hiperligação ao editar ficheiros HTML: poderá digitar simplesmente `<a` no editor e premir `Control+J`. O editor irá expandi-lo para:

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

Não apenas isto, a palavra `filename` será selecionada, com o cursor sobre ela, assim você pode facilmente digitar o verdadeiro nome do arquivos, usando o recurso astucioso do editor *Auto-completar* (página 107). Assim que você terminar de digitar o nome do arquivo, pressione `Control+J` novamente e o cursor se moverá para a posição entre as tags `<a>`, assim você poderá facilmente digitar o texto para o link.

O sistema de trechos do editor é muito sofisticado, há uma pequena quantidade de trechos embutidos e você pode criar seus próprios trechos para se adequar ao seu estilo de edição.

A seguinte discussão sobre os trechos embutidos devem ajudar a ilustrar o poder do sistema de trechos.

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

---

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

Trechos integrados

Os trechos embutidos estão descritos logo abaixo. Note que você pode substitui-los criando seus próprios trechos com o mesmo texto de ativação.

Inserindo texto de preenchimento [Lorem]

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

A definição deste trecho é bem simples, o texto de ativação é definido como `Lorem` e o modelo é definido simplesmente como o texto literal a ser inserido. Você pode facilmente personaliza-lo para usar sua forma favorita de texto de preenchimento.

Inserindo uma tag HTML de fechamento automático [<>]

Agora olhe um simples exemplo do poderoso conceito de *placeholders*. Digamos que você queira inserir a tag de fechamento automático `<hr/>`. Apenas digite `<>`, e pressione `Control+J`, o editor expandirá o trecho para:

```
<|/>
```

Aqui, o símbolo `|` representa a atual posição do cursor. Você pode então pressionar `hr` e pressionar `Control+J` para mover o cursor para o final da tag. Este trecho é definido como:

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

Placeholders são simplesmente o símbolo dólar ($) seguido por um número. Quando o trecho é expandido ao pressionar `Control+J` o cursor é posicionado no primeiro placeholder (aquele com o menor número). Quando você pressionar `Control+J` novamente, o cursor move-se para o próximo placeholder (aquel com o próximo número superior).

A inserir uma etiqueta de hiperligação HTML [<a]

Todas as tags HTML de link compartilham uma estrutura comum. Elas tem um atributo `href` e algum texto entre as tags de abertura e fechamento. Um trecho para tornar a digitação delas mais eficiente nos introduzirá a mais características dos placeholders. Para usar esse trecho, simplesmente digite `<a` e pressione `Control+J`. O editor vai expandir isso para:

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

Não só isso, a palavra «filename» vai ser selecionada, com o cursor localizado sobre esta, assim você pode facilmente digitar no nome do arquivo, usando o recurso de estilo do editor *Auto-completar* (página 107). E uma vez que você acabou de digitar o nome do arquivo, pressione `Control+J` novamente e o cursor vai pular para a posição entre as tags `<a>` assim você pode facilmente digitar o texto para o link. Depois de você acabar de digitar o texto, pressione `Control+J` novamente para pular para o ponto depois da tag de fechamento. Esse fragmento é definido como:

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

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

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

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

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

Inserting a HTML image tag [<i]

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

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

Inserir uma etiqueta HTML aleatória [<<]

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

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

Digite o nome da tag, por exemplo: `span` e pressione `Control+J`, isto irá resultar em:

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

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

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

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

A inserir uma etiqueta HTML aleatória com o atributo de classe [<c]

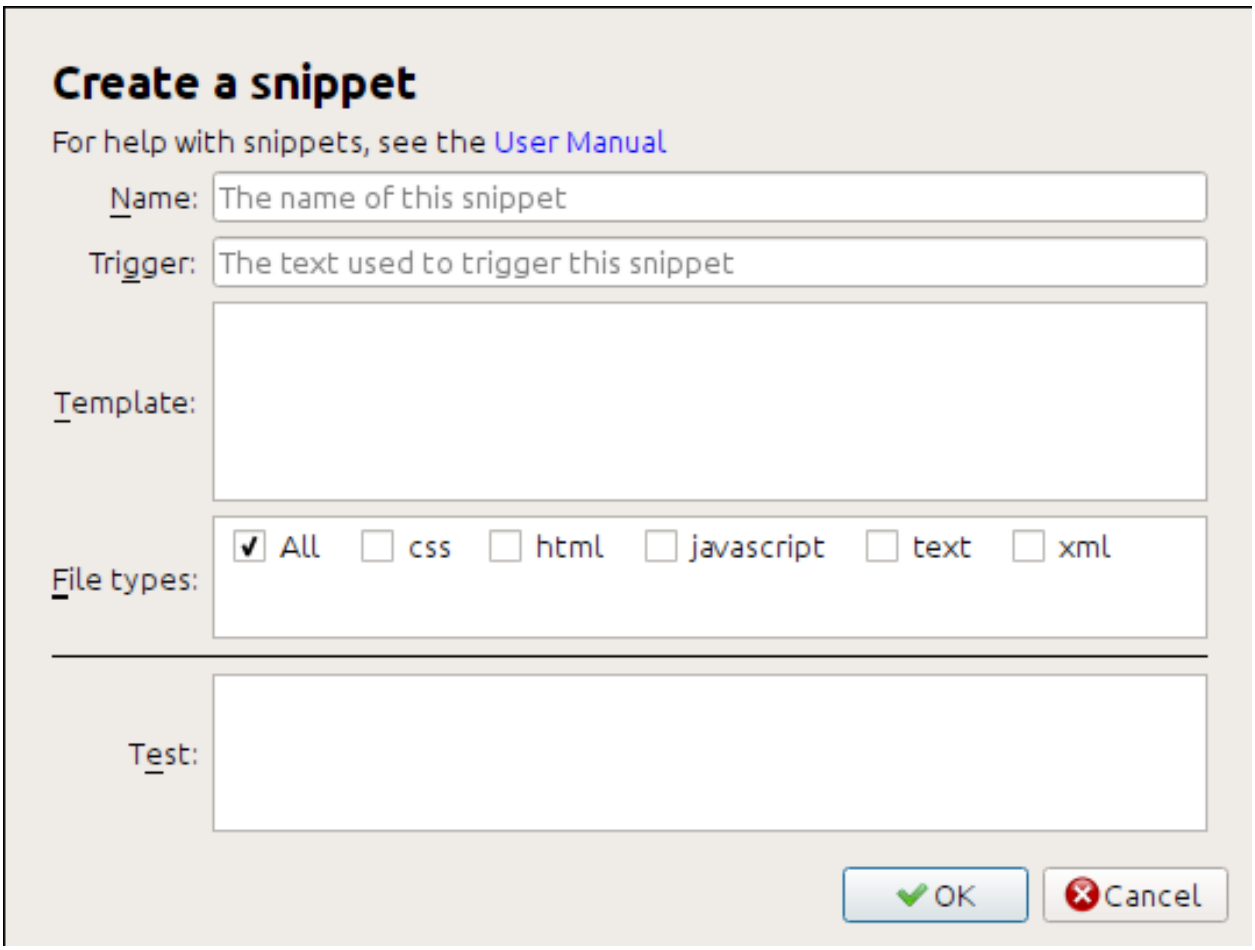
This is very similar to the insert arbitrary tag example above, except that it assumes that you want to specify a class for the tag:

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

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

Criar os seus próprios trechos

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



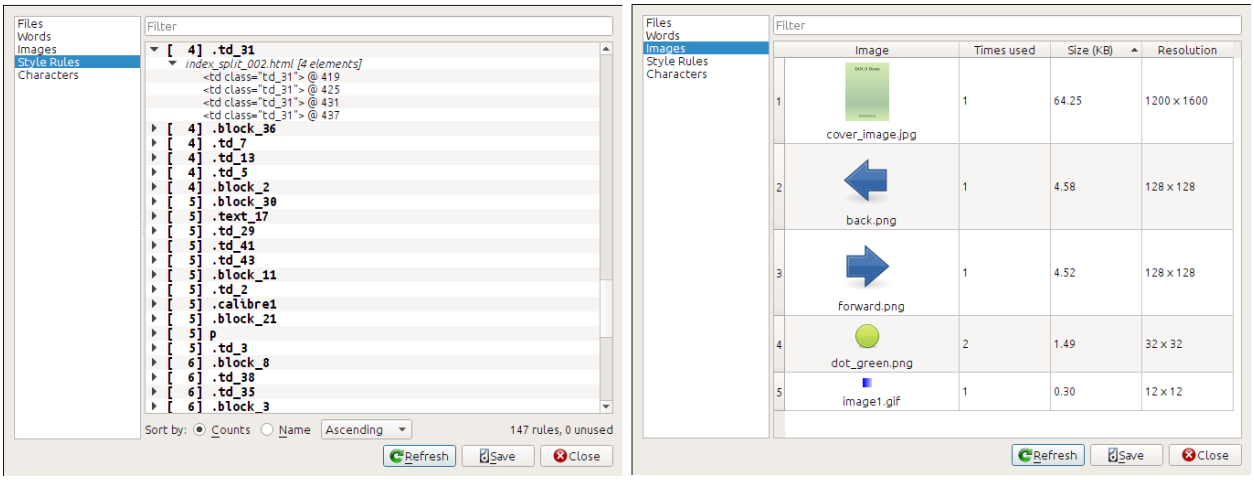
First give your snippet a name, something descriptive, to help identify the snippet in the future. Then specify the *trigger*. A trigger is simply the text that you have to type in the editor before pressing `Control+J` in order to expand the snippet.

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

O próximo passo é testar o seu novo trecho criado. Use a caixa *Teste* na parte inferior. Digite o texto de ativação e pressione `Control+J` para expandir o trecho e mover-se entre os placeholders.

5.8.9 A ferramenta de relatórios

The editor includes a nice *Reports* tool (via *Tools → Reports*) that shows summaries of the files, images, links, words, characters and styles used in the book. Every line in the report is hot-linked. Double clicking a line jumps to the place in the book where that item is used or defined (as appropriate). For example, in the *Links* view, you can double click entries the *Source* column to jump to where the link is defined and entries in the *Target* column to jump to where the link points.



5.9 Características especiais no editor de código

The calibre HTML editor is very powerful. It has many features that make editing of HTML (and CSS) easier.

5.9.1 Realçar sintaxe

The HTML editor has very sophisticated syntax highlighting. Features include:

- The text inside bold, italic and heading tags is made bold/italic
- As you move your cursor through the HTML, the matching HTML tags are highlighted, and you can jump to the opening or closing tag with the keyboard shortcuts `Ctrl+{` and `Ctrl+}`. Similarly, you can select the contents of a tag with `Ctrl+Alt+T` or `Ctrl+Shift+T`.
- Invalid HTML is highlighted with a red underline
- Spelling errors in the text inside HTML tags and attributes such as title are highlighted. The spell checking is language aware, based on the value of the `lang` attribute of the current tag and the overall book language.
- CSS embedded inside `<style>` tags is highlighted
- Special characters that can be hard to distinguish such as non-breaking spaces, different types of hyphens, etc. are highlighted.
- Links to other files in `<a>` tags, `<img>` and `<link>` tags all have the filenames highlighted. If the filename they point to does not exist, the filename is marked with a red underline.

5.9.2 Ajuda de conteúdo sensível

You can right click on an HTML tag name or a CSS property name to get help for that tag or property.

You can also hold down the `Ctrl` key and click on any filename inside a link tag to open that file in the editor automatically. Similarly, `Ctrl` clicking a class name will take you to the first style rule that matches the tag and class.

Right clicking a class name in an HTML file will allow you to rename the class, changing all occurrences of the class throughout the book and all its stylesheets.

5.9.3 Auto-completar

When editing an e-book, one of the most tedious tasks is creating links to other files inside the book, or to CSS stylesheets, or images. You have to figure out the correct filename and relative path to the file. The editor has auto-complete to make that easier.

As you type a filename, the editor automatically pops up suggestions. Simply use the `Tab` key to select the correct file name. The editor even offers suggestions for links pointing to an anchor inside another HTML file. After you type the # character, the editor will show you a list of all anchors in the target file, with a small snippet of text to help you choose the right anchor.

Note that unlike most other completion systems, the editor's completion system uses subsequence matching. This means that you can type just two or three letters from anywhere in the filename to complete the filename. For example, say you want the filename `../images/arrow1.png`, you can simply type `ia1` and press `Tab` to complete the filename. When searching for matches, the completion system prioritizes letters that are at the start of a word, or immediately after a path separator. Once you get used to this system, you will find it saves you a lot of time and effort.

5.9.4 Trechos

O editor Calibre de ebook suporta *trechos*. Um trecho é um pedaço de texto que é reutilizado com frequência ou contém algum texto redundante. O editor permite que você insira um trecho com apenas algumas combinações de teclas. Os trechos são uma funcionalidade poderosa, com muitos recursos, como marcadores de espaço entre os quais você pode saltar, espelhamento automático de texto repetido e assim por diante. Para obter mais informações, consulte *Trechos* (página 102).

CAPÍTULO 6

O servidor de conteúdos Calibre

The calibre *Content server* allows you to access your calibre libraries and read books directly in a browser on your favorite mobile phone or tablet device. As a result, you do not need to install any dedicated book reading/management apps on your phone. Just use the browser. The server downloads and stores the book you are reading in an off-line cache so that you can read it even when there is no internet connection.

Conteúdos

- *Aceder ao servidor a partir de outros dispositivos* (página 110)
  - *Aceder ao servidor a partir de dispositivos na sua rede local* (página 110)
  - *Aceder ao servidor a partir de qualquer local na Internet* (página 111)
- *A interface do servidor* (página 111)
  - *A lista de livros* (página 112)
  - *O leitor de livros* (página 112)
- *Browser support* (página 112)
- *Enabling offline support* (página 112)
- *Managing user accounts from the command-line only* (página 113)
- *Integrating the calibre Content server into other servers* (página 113)
  - *Using a full virtual host* (página 113)
  - *Usar um prefixo de URL* (página 114)
- *Creating a service for the calibre server on a modern Linux system* (página 115)

To start the server, click the *Connect/share* button and choose *Start Content server*. You might get a message from your computer's firewall or anti-virus program asking if it is OK to allow access to `calibre.exe`. Click the `Allow` or `OK` button. Then open a browser (preferably Chrome or Firefox) in your computer and type in the following address:

http://127.0.0.1:8080

This will open a page in the browser showing you your calibre libraries, click on any one and browse the books in it. Click on a book, and it will show you all the metadata about the book, along with buttons to *Read book* and *Download book*. Click the *Read book* button to start reading the book.

Nota: The address used above `http://127.0.0.1:8080` will only work on the computer that is running calibre. To access the server from other computers/phones/tablets/etc. you will need to do a little more work, as described in the next section.

6.1 Aceder ao servidor a partir de outros dispositivos

There are two types of remote device access that you will typically need. The first, simpler kind is from within your home network. If you are running calibre on a computer on your home network and you have also connected your other devices to the same home network, then you should be easily able to access the server on those devices.

6.1.1 Aceder ao servidor a partir de dispositivos na sua rede local

After starting the server in calibre as described above, click the *Connect/share* button again. Instead of the *Start Content server* action, you should see a *Stop Content server* action instead. To the right of this action will be listed an IP address and port number. These look like a bunch of numbers separated by periods. For example:

```
Stop Content server [192.168.1.5, port 8080]
```

These numbers tell you what address to use to connect to the server in your devices. Following the example above, the address becomes:

```
http://192.168.1.5:8080
```

The first part of the address is always `http://` the next part is the IP address, which is the numbers before the comma and finally we have the port number which must be added to the IP address with a colon (`:`). If you are lucky, that should be all you need and you will be looking at the calibre libraries on your device. If not, read on.

Trouble-shooting the home network connection

If you are unable to access the server from your device, try the following steps:

1. Check that the server is running by opening the address `http://127.0.0.1:8080` in a browser running on the same computer as the server.
2. Check that your firewall/anti-virus is allowing connections to your computer on the port `8080` and to the calibre program. The easiest way to eliminate the firewall/anti-virus as the source of problems is to temporarily turn them both off and then try connecting. You should first disconnect from the internet, before turning off the firewall, to keep your computer safe.
3. Check that your device and computer are on the same network. This means they should both be connected to the same wireless router. In particular neither should be using a cellular or ISP provided direct-WiFi connection.
4. If you have non-standard networking setup, it might be that the IP address shown on the *Connect/share* menu is incorrect. In such a case you will have to figure out what the correct IP address to use is, yourself. Unfortunately, given the infinite diversity of network configurations possible, it is not possible to give you a roadmap for doing so.

5. If you have setup a username and password, first try it without that to see if it is causing issues. Some e-ink devices have browsers that do not handle authentication. You can sometimes workaround this by including the username and password in the URL, for example: `http://username:password@192.168.1.2:8080`.
6. If you are stuck, you can always ask for help in the calibre user forums[42].

6.1.2 Aceder ao servidor a partir de qualquer local na Internet

Aviso: Before doing this you should turn on username/password protection in the server, otherwise anyone in the world will be able to access your books. Go to *Preferences → Sharing → Sharing over the net* and enable the option to *Require username and password to access the content server*.

While the particular details on setting up internet access vary depending on the network configuration and type of computer you are using, the basic schema is as follows.

1. Find out the external IP address of the computer you are going to run the server on. You can do that by visiting the site What is my IP address[43] in a browser running on the computer.
2. If the computer is behind a router, enable port forwarding on the router to forward the port `8080` (or whatever port you choose to run the calibre Content server on) to the computer.
3. Make sure the calibre server is allowed through any firewalls/anti-virus programs on your computer.
4. Now you should be able to access the server on any internet-connected device using the IP address you found in the first step. For example, if the IP address you found was `123.123.123.123` and the port you are using for the calibre server is `8080`, the address to use on your device becomes: `http://123.123.123.123:8080`.
5. Optionally, use a service like no-ip[44] to setup an easy to remember address to use instead of the IP address you found in the first step.

Nota: For maximum security, you should also enable HTTPS on the Content server. You can either do so directly in the server by providing the path to the HTTPS certificate to use in the advanced configuration options for the server, or you can setup a reverse proxy as described below, to use an existing HTTPS setup.

6.2 A interface do servidor

The server interface is a simplified version of the main calibre interface, optimised for use with touch screens. The home screen shows you books you are currently reading as well as allowing to choose a calibre library you want to browse. The server in calibre gives you access to all your libraries, not just a single one, as before.

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

6.2.1 A lista de livros

The server book list is a simple grid of covers. Tap on a cover to see the detailed metadata for a book, or to read the book. If you prefer a more detailed list, you can change the default view by clicking the three vertical dots in the top right corner.

Sorting and searching of the book list should be familiar to calibre users. They can be accessed by clicking their icons in the top right area. They both work exactly the same as in the main calibre program. The search page even allows you to construct search queries by clicking on authors/tags/etc., just as you can using the Tag browser in the main program.

A much loved feature of the main program, *Virtual libraries* is present in the server interface as well. Click the three vertical dots in the top right corner to choose a Virtual library.

6.2.2 O leitor de livros

You can read any book in your calibre library by simply tapping on it and then tapping the *Read book* button. The books reader is very simple to operate. You can both tap and swipe to turn pages. Swiping up/down skips between chapters. Tapping the top quarter of the screen gets you the detailed controls and viewer preferences.

If you leave the Content server running, you can even open the same book on multiple devices and it will remember your last read position. If it does not you can force a sync by tapping in the top quarter and choosing *Sync*.

6.3 Browser support

The new calibre server makes lots of use of advanced HTML 5 and CSS 3 features. As such it requires an up-to-date browser to use. It has been tested on Android Chrome and iOS Safari as well as Chrome and Firefox on the desktop.

The server is careful to use functionality that has either been already standardised or is on the standards track. As such if it does not currently work with your favorite browser, it probably will once that browser has caught up.

If you are using a particularly old or limited browser or you don't like to run JavaScript, you can use the *mobile* view, by simply adding `/mobile` to the server address.

Nota: On iOS, Apple allows only a single browser engine, so Firefox, Chrome and Safari are all actually the same browser under the hood. The new server interface requires iOS 10.3.2 or newer. On Android, the server has been tested with Chrome version 58 and newer.

6.4 Enabling offline support

Browser makers have been trying to force people to use SSL by disabling advanced features in their browsers for plain HTTP connections. One such casualty is offline support. So you may need to enable HTTPS on the server to get offline support working. In addition, in Firefox on Android, you will need to type `about:config` and search for `browser.tabs.useCache` and toggle it to `true`.

6.5 Managing user accounts from the command-line only

The calibre program has a nice section in *Preferences* to allow you to manage user accounts for the server. However, if you want to run the standalone server and cannot run the main calibre program on the same computer/user account, you can also manage users using just the command-line.

You can manage user accounts using the `--manage-users` option to the standalone `calibre-server` program. Suppose you want to store the user database in the folder `/srv/calibre`, then you create it by running:

```
calibre-server --userdb /srv/calibre/users.sqlite --manage-users
```

Just follow the prompts to create user accounts, set their permission, etc. Once you are done, you can run the server as:

```
calibre-server --userdb /srv/calibre/users.sqlite --enable-auth
```

It will use the user accounts you created in the previous step.

6.6 Integrating the calibre Content server into other servers

Here, we will show you how to integrate the calibre Content server into another server. The most common reason for this is to make use of SSL or to serve the calibre library as part of a larger site. The basic technique is to run the calibre server and setup a reverse proxy to it from the main server.

A reverse proxy is when your normal server accepts incoming requests and passes them onto the calibre server. It then reads the response from the calibre server and forwards it to the client. This means that you can simply run the calibre server as normal without trying to integrate it closely with your main server.

6.6.1 Using a full virtual host

The simplest configuration is to dedicate a full virtual host to the calibre server. In this case, run the calibre server as:

```
calibre-server
```

Now setup the virtual host in your main server, for example, for nginx:

```
http {
    client_max_body_size 64M;  # needed to upload large books
}

server {
    listen [::]:80;
    server_name myserver.example.com;

    location / {
        proxy_pass http://127.0.0.1:8080;
    }
}
```

Or, for Apache:

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

(continues on next page)

(continuação da página anterior)

```
<VirtualHost *:80>
    ServerName myserver.example.com
    AllowEncodedSlashes On
    ProxyPreserveHost On
    ProxyPass "/"  "http://localhost:8080/"
</VirtualHost>
```

6.6.2 Usar um prefixo de URL

If you do not want to dedicate a full virtual host to calibre, you can have it use a URL prefix. Start the calibre server as:

```
calibre-server --url-prefix /calibre --port 8080
```

The key parameter here is `--url-prefix /calibre`. This causes the Content server to serve all URLs prefixed by `/calibre`. To see this in action, visit `http://localhost:8080/calibre` in your browser. You should see the normal Content server website, but now it will run under `/calibre`.

With nginx, the required configuration is:

```
http {
    client_max_body_size 64M;  # needed to upload large books
}

proxy_set_header X-Forwarded-For $remote_addr;
location /calibre/ {
    proxy_buffering off;
    proxy_pass http://127.0.0.1:8080$request_uri;
}
location /calibre {
    # we need a trailing slash for the Application Cache to work
    rewrite /calibre /calibre/ permanent;
}
```

For Apache, first enable the proxy modules in Apache, by adding the following to `httpd.conf`:

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

The exact technique for enabling the proxy modules will vary depending on your Apache installation. Once you have the proxy modules enabled, add the following rules to `httpd.conf` (or if you are using virtual hosts to the conf file for the virtual host in question):

```
AllowEncodedSlashes On
RewriteEngine on
RewriteRule ^/calibre/(.*) http://127.0.0.1:8080/calibre/$1 [proxy]
RedirectMatch permanent ^/calibre$ /calibre/
```

That's all, you will now be able to access the calibre Content server under the `/calibre` URL in your main server. The above rules pass all requests under `/calibre` to the calibre server running on port 8080 and thanks to the `--url-prefix` option above, the calibre server handles them transparently.

Nota: When using a reverse proxy, you should tell the calibre Content server to only listen on localhost, by using `--listen-on 127.0.0.1`. That way, the server will only listen for connections coming from the same computer,

i.e. from the reverse proxy.

Nota: If you have setup SSL for your main server, you should tell the calibre server to use basic authentication instead of digest authentication, as it is faster. To do so, pass the `--auth-mode=basic` option to `calibre-server`.

6.7 Creating a service for the calibre server on a modern Linux system

You can easily create a service to run calibre at boot on a modern (systemd[45]) based Linux system. Just create the file `/etc/systemd/system/calibre-server.service` with the contents shown below:

```
[Unit]
Description=calibre Content server
After=network.target

[Service]
Type=simple
User=mylinuxuser
Group=mylinuxgroup
ExecStart=/opt/calibre/calibre-server "/path/to/calibre library folder"

[Install]
WantedBy=multi-user.target
```

Change `mylinuxuser` and `mylinuxgroup` to whatever user and group you want the server to run as. This should be the same user and group that own the files in the calibre library folder. Note that it is generally not a good idea to run the server as root. Also change the path to the calibre library folder to suit your system. You can add multiple libraries if needed. See the help for the `calibre-server` command.

Now run:

```
sudo systemctl start calibre-server
```

to start the server. Check its status with:

```
sudo systemctl status calibre-server
```

To make it start at boot, run:

```
sudo systemctl enable calibre-server
```

Nota: The calibre server *does not* need a running X server, but it does need the X libraries installed as some components it uses link against them.

Nota: The calibre server also supports systemd socket activation, so you can use that, if needed, as well.

[45] https://www.freedesktop.org/wiki/Software/systemd/

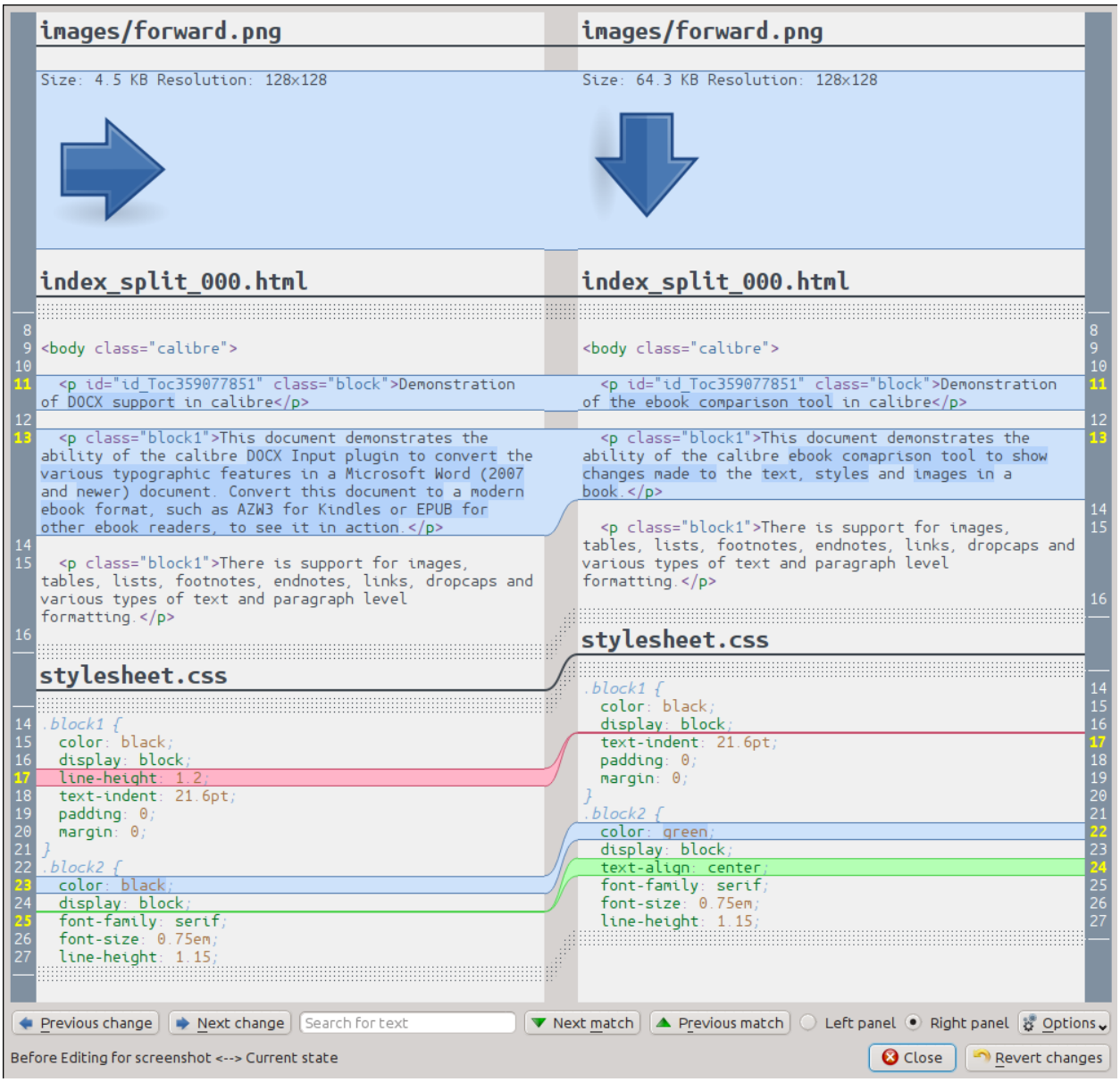
CAPÍTULO 7

Comparar livros digitais

O Calibre inclui uma ferramenta integrada de comparação de ebooks que pode ser usada para ver o que mudou dentro de um ebook após o editar ou converter. Podem ser comparados livros nos formatos EPUB ou AZW3.

Para a usar pode abrir o ebook na ferramenta de *Editar livros digitais* (página 75) e de seguida clicar *Ficheiro → Comparar com outro livro* ou então usar o painel *Detalhes do livro* (página 20). Se efetuar uma conversão de EPUB para EPUB, o ficheiro EPUB original será guardado como ORIGINAL_EPUB. Bastará clicar como o botão direito na entrada ORIGINAL_EPUB no painel de detalhes do Livro e selecione *Comparar com formato EPUB*.

A ferramenta de comparação que irá abrir será parecida com a imagem abaixo. Irá mostrar as diferenças de texto, estilos e imagens nos livros escolhidos.



7.1 Compreender a vista de comparação.

Como pode ver na imagem acima, a janela de comparação mostra a diferença entre os dois livros, lado a lado. Apenas as diferenças, com algumas linhas de contexto ao redor, são apresentadas. Isto torna mais fácil ver apenas o que mudou dentro dum grande documento como um livro.

Textos adicionado é mostrado num fundo verde, textos removidos num fundo vermelho e textos alterados num fundo azul.

O número das linhas de todo o texto modificado, é apresentado no lado, tornando mais fácil ir para uma alteração específica do editor. Quando a ferramenta de comparação é aberta de dentro do editor, podemos também usar um duplo-clique na linha no painel direito para ir à mesma linha no editor automaticamente.

Uma técnica útil para quando compara livros consiste em dizer à ferramenta de comparação para embelezar os ficheiros de texto e de estilo antes de calcular as diferenças. Tal pode muitas vezes resultar em diferenças mais claras e fáceis de rastrear. Para fazer isto, clique no botão *Opções* situado no fundo à direita e escolha *Embelezar ficheiros antes de*

comparar. Note que o embelezamento pode causar alguns efeitos indesejáveis, dado que pode provocar a alteração de markup inválido de modo a torná-lo válido. Você pode ainda alterar a quantidade de linhas de contexto à volta das diferenças através do botão *Opções*.

Pode pesquisar por qualquer texto nas diferenças através da barra de pesquisa localizada ao fundo. Tem de especificar qual o painel a pesquisar, o *Esquerda* ou o *Direita*.

7.2 Invocar a ferramenta de comparação

A ferramenta de comparação torna-se mais útil quando se tem duas versões do mesmo livro e se pretende verificar o que há de diferente entre eles. Para esse fim, existem vários modos de iniciar a ferramenta.

7.2.1 Comparar dois ficheiros de ebook

Abra o primeiro ficheiro na ferramenta *Editar livros digitais* (página 75). De seguida, clique *Ficheiro → Comparar com outro livro* e selecione o segundo ficheiro (que tem de estar no mesmo formato do primeiro). A vista de comparação será aberta com o ficheiro que está a ser editado à direita e o segundo ficheiro à esquerda.

7.2.2 Comparar o ORIGINAL_FMT com o FMT

Quando executa no Calibre uma conversão de um FMT para ele mesmo, o ficheiro original é guardado como ORIGINAL_FMT. Pode verificar o que mudou com a conversão clicando com o botão direito na entrada do ORIGINAL_FMT no painel *Detalhes do livro* (página 20) na janela principal do Calibre e selecionar *Comparar com o FMT*. A vista de comparação abrirá com o ORIGINAL_FMT à esquerda e o FMT à direita.

7.2.3 Comparar um ponto de verificação com o estado atual do livro enquanto se edita

A ferramenta *Editar livros digitais* (página 75) tem uma funcionalidade muito útil, denominada *Pontos de controlo* (página 87). Esta permite-lhe guardar o estado atual do livro sob a forma de um *ponto de controlo*, para o qual pode reverter se não gostar das alterações efetuadas desde que criou o ponto de controlo. São também criados automaticamente pontos de controlo quando se executam algumas ações automáticas no editor. Pode ver a lista de pontos de controlo acedendo ao *Ver → Pontos de controlo* e de seguida usar o botão *Comparar* para comparar o livro no ponto de controlo selecionado face ao estado atual. A ferramenta de comparação mostrará o ponto de controlo à esquerda e o estado atual à direita.

CAPÍTULO 8

A editar metadados do livro

Conteúdos

- *Editar detalhes dos livros individualmente* (página 121)
  - *Descarregar metadados* (página 122)
  - *Gerir formatos de ebooks* (página 122)
  - *Tudo sobre capas* (página 122)
- *Editar metadados de vários livros ao mesmo tempo* (página 122)
  - *Pesquisar e substituir* (página 123)
  - *Descarregamento de metadados em lote* (página 124)

E-books come in all shapes and sizes and more often than not, their metadata (things like title/author/series/publisher) is incomplete or incorrect. The simplest way to change metadata in calibre is to simply double click on an entry and type in the correct replacement. For more sophisticated, «power editing» use the edit metadata tools discussed below.

8.1 Editar detalhes dos livros individualmente

Click the book you want to edit and then click the *Edit metadata* button or press the `E` key. A dialog opens that allows you to edit all aspects of the metadata. It has various features to make editing faster and more efficient. A list of the commonly used tips:

- Pode clicar no botão entre o título e o autor para os trocar automaticamente
- You can click the button next to author sort to have calibre automatically fill it in using the sort values stored with each author. Use the *Manage authors* dialog to see and change the authors" sort values. This dialog can be opened by clicking and holding the button next to author sort.
- You can click the button next to tags to use the *Tag editor* to manage the tags associated with the book.

- The «Ids» box can be used to enter an ISBN (and many other types of id), it will have a red background if you enter an invalid ISBN. It will be green for valid ISBNs.
- The author sort box will be red if the author sort value differs from what calibre thinks it should be.

8.1.1 Descarregar metadados

The nicest feature of the edit metadata dialog is its ability to automatically fill in many metadata fields by getting metadata from various websites. Currently, calibre uses Google Books and Amazon. The metadata download can fill in Title, author, series, tags, rating, description and ISBN for you.

To use the download, fill in the title and author fields and click the *Fetch metadata* button. calibre will present you with a list of books that most closely match the title and author. If you fill in the ISBN field first, it will be used in preference to the title and author. If no matches are found, try making your search a little less specific by including only some key words in the title and only the author last name.

8.1.2 Gerir formatos de ebooks

In calibre, a single book entry can have many different *formats* associated with it. For example you may have obtained the Complete Works of Shakespeare in EPUB format and later converted it to MOBI to read on your Kindle. calibre automatically manages multiple formats for you. In the *Available formats* section of the Edit metadata dialog, you can manage these formats. You can add a new format, delete an existing format and also ask calibre to set the metadata and cover for the book entry from the metadata in one of the formats.

8.1.3 Tudo sobre capas

You can ask calibre to download book covers for you, provided the book has a known ISBN. Alternatively you can specify a file on your computer to use as the cover. calibre can even generate a default cover with basic metadata on it for you. You can drag and drop images onto the cover to change it and also right click to copy/paste cover images.

In addition, there is a button to automatically trim borders from the cover, in case your cover image has an ugly border.

8.2 Editar metadados de vários livros ao mesmo tempo

First select the books you want to edit by holding `Ctrl` or `Shift` and clicking on them. If you select more than one book, clicking the *Edit metadata* button will cause the *Bulk* metadata edit dialog to open. Using this dialog, you can quickly set the author/publisher/rating/tags/series etc of a bunch of books to the same value. This is particularly useful if you have just imported a number of books that have some metadata in common. This dialog is very powerful, for example, it has a *Search and replace* tab that you can use to perform bulk operations on metadata and even copy metadata from one column to another.

The normal edit metadata dialog also has *Next* and *Previous* buttons that you can use to edit the metadata of several books one after the other.

8.2.1 Pesquisar e substituir

The *Edit metadata for many books* dialog allows you to perform arbitrarily powerful search and replace operations on the selected books. By default it uses a simple text search and replace, but it also support *regular expressions*. For more on regular expressions, see *Tudo sobre utilização de expressões regulares no Calibre* (página 208).

As noted above, there are two search and replace modes: character match and regular expression. Character match will look in the *Search field* you choose for the characters you type in the *search for* box and replace those characters with what you type in the *replace with* box. Each occurrence of the search characters in the field will be replaced. For example, assume the field being searched contains *a bad cat*. If you search for *a* to be replaced with *HELLO*, then the result will be *HELLO bHELLOd cHELLOt*.

If the field you are searching on is a *multiple* field like tags, then each tag is treated separately. For example, if your tags contain *Horror, Scary*, the search expression *r,* will not match anything because the expression will first be applied to *Horror* and then to *Scary*.

If you want the search to ignore upper/lowercase differences, uncheck the *Case sensitive* box.

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

- *Lower case* – change all the characters in the field to lower case
- *Upper case* – altera todos os caracteres no campo para maiúsculas.
- *Title case* – Maiusculizar cada palavra no resultado

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

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

The third and most important is that the replace string can make reference to parts of the search string by using backreferences. A backreference is `\\n` where n is an integer that refers to the n'th parenthesized group in the search expression. For example, given the same example as above, *a bad cat*, a search expression *a (…) (…)*, and a replace expression *a \2 \1*, the result will be *a cat bad*. Please see the *Tudo sobre utilização de expressões regulares no Calibre* (página 208) for more information on backreferences.

One useful pattern: assume you want to change the case of an entire field. The easiest way to do this is to use character mode, but lets further assume you want to use regular expression mode. The search expression should be *(^.*$)*, the replace expression should be *\1*, and the desired case change function should be selected.

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

Search and replace is done after all the other metadata changes in the other tabs are applied. This can lead to some confusion, because the test boxes will show the information before the other changes, but the operation will be applied after the other changes. If you have any doubts about what is going to happen, do not mix search/replace with other changes.

8.2.2 Descarregamento de metadados em lote

If you want to download the metadata for multiple books at once, right-click the *Edit metadata* button and select *Download metadata*. You can choose to download only metadata, only covers, or both.

CAPÍTULO 9

Questões frequentes

Conteúdos

- *Conversão de formatos de ebooks* (página 125)
- *Integração de dispositivos* (página 128)
- *Library management* (página 135)
- *Diversos* (página 140)

9.1 Conversão de formatos de ebooks

Conteúdos

- *Quais os formatos de conversão o calibre de suportar para/de?* (página 126)
- *Quais são os melhores formatos fonte para conversão?* (página 126)
- *Converti um ficheiro PDF mas o resultado tem vários problemas?* (página 126)
- *Como converto meu arquivo que contém caracteres que não são da língua inglesa, ou que contém aspas inteligentes?* (página 126)
- *Qual é o problema com o Índice em arquivos MOBI?* (página 127)
- *Como converto uma coleção de arquivos HTML em uma ordem específica?* (página 127)
- *O EPUB que criei com o Calibre não é válido?* (página 128)
- *Como faço para usar alguns dos recursos avançados das ferramentas de conversão?* (página 128)

9.1.1 Quais os formatos de conversão o calibre de suportar para/de?

calibre suporta a conversão de vários formatos de entrada para vários formatos de saída. Consegue converter todoo os formatos de entrada da seguinta lista, para todo formato de saída.

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

Formatos de saída: AZW3, EPUB, DOCX, FB2, HTMLZ, OEB, LIT, LRF, MOBI, PDB, PMLZ, RB, PDF, RTF, SNB, TCR, TXT, TXTZ, ZIP

Nota: PRC is a generic format, calibre supports PRC files with TextRead and MOBIBook headers. PDB is also a generic format. calibre supports eReader, Plucker (input only), PML and zTxt PDB files. DJVU support is only for converting DJVU files that contain embedded text. These are typically generated by OCR software. MOBI books can be of two types Mobi6 and KF8. calibre fully supports both. MOBI files often have .azw or .azw3 file extensions. DOCX files from Microsoft Word 2007 and newer are supported.

9.1.2 Quais são os melhores formatos fonte para conversão?

Em ordem decrescente de preferência: LIT, MOBI, AZW, EPUB, AZW3, FB2, FBZ, DOCX, HTML, PRC, ODT, RTF, PDB, TXT, PDF

9.1.3 Converti um ficheiro PDF mas o resultado tem vários problemas?

PDF é um formato terrível de onde se converter. Para uma lista dos vários problemas que encontrará ao converter um PDF, veja: *Converter documentos de PDF* (página 70).

9.1.4 Como converto meu arquivo que contém caracteres que não são da língua inglesa, ou que contém aspas inteligentes?

Existem dois aspetos neste problema:

1. Knowing the encoding of the source file: calibre tries to guess what character encoding your source files use, but often, this is impossible, so you need to tell it what encoding to use. This can be done in the GUI via the *Input character encoding* field in the *Look & feel → Text* section of the conversion dialog. The command-line tools have an `ebook-convert-txt-input --input-encoding` option.
2. When adding HTML files to calibre, you may need to tell calibre what encoding the files are in. To do this go to *Preferences → Advanced → Plugins → File type* and customize the *HTML to ZIP* plugin, telling it what encoding your HTML files are in. Now when you add HTML files to calibre they will be correctly processed. HTML files from different sources often have different encodings, so you may have to change this setting repeatedly. A common encoding for many files from the web is `cp1252` and I would suggest you try that first. Note that when converting HTML files, leave the input encoding setting mentioned above blank. This is because the *HTML to ZIP* plugin automatically converts the HTML files to a standard encoding (UTF-8).

9.1.5 Qual é o problema com o Índice em arquivos MOBI?

The first thing to realize is that most e-books have two tables of contents. One is the traditional Table of Contents, like the ToC you find in paper books. This Table of Contents is part of the main document flow and can be styled however you like. This ToC is called the *content ToC*.

Then there is the *metadata ToC*. A metadata ToC is a ToC that is not part of the book text and is typically accessed by some special button on a reader. For example, in the calibre E-book viewer, you use the Show Table of Contents button to see this ToC. This ToC cannot be styled by the book creator. How it is represented is up to the viewer program.

In the MOBI format, the situation is a little confused. This is because the MOBI format, alone amongst mainstream e-book formats, *does not* have decent support for a metadata ToC. A MOBI book simulates the presence of a metadata ToC by putting an *extra* content ToC at the end of the book. When you click Goto Table of Contents on your Kindle, it is to this extra content ToC that the Kindle takes you.

Now it might well seem to you that the MOBI book has two identical ToCs. Remember that one is semantically a content ToC and the other is a metadata ToC, even though both might have exactly the same entries and look the same. One can be accessed directly from the Kindle's menus, the other cannot.

When converting to MOBI, calibre detects the *metadata ToC* in the input document and generates an end-of-file ToC in the output MOBI file. You can turn this off by an option in the MOBI Output settings. You can also tell calibre whether to put it at the start or the end of the book via an option in the MOBI Output settings. Remember this ToC is semantically a *metadata ToC*, in any format other than MOBI it *cannot not be part of the text*. The fact that it is part of the text in MOBI is an accident caused by the limitations of MOBI. If you want a ToC at a particular location in your document text, create one by hand. So we strongly recommend that you leave the default as it is, i.e. with the metadata ToC at the end of the book. Also note that if you disable the generation of the end-of-file ToC the resulting MOBI file may not function correctly on a Kindle, since the Kindle's use the metadata ToC for many things, including the Page Flip feature.

If you have a hand edited ToC in the input document, you can use the ToC detection options in calibre to automatically generate the metadata ToC from it. See the conversion section of the User Manual for more details on how to use these options.

Finally, I encourage you to ditch the content ToC and only have a metadata ToC in your e-books. Metadata ToCs will give the people reading your e-books a much superior navigation experience (except on the Kindle, where they are essentially the same as a content ToC).

Nota: The newer AZW3 format has proper support for a metadata ToC. However, the Kindle firmware tends to malfunction if you disable the generation of the end-of-file inline ToC. So it is recommended that you leave the generated ToC alone. If you create an AZW3 file with a metadata ToC and no end-of-file generated ToC, some features on the Kindle will not work, such as the Page Flip feature.

9.1.6 Como converto uma coleção de arquivos HTML em uma ordem específica?

In order to convert a collection of HTML files in a specific order, you have to create a table of contents file. That is, another HTML file that contains links to all the other files in the desired order. Such a file looks like:

```
<html>
   <body>
     <h1>Table of Contents</h1>
     <p style="text-indent:0pt">
        <a href="file1.html">First File</a><br/>
        <a href="file2.html">Second File</a><br/>
        .
        .
```

(continues on next page)

(continuação da página anterior)

```
        .
      </p>
    </body>
</html>
```

Então, basta adicionar este arquivo HTML à GUI e usar o botão *Convert* para criar seu e-book. É possível utilizar a opção na seção *Índice de Conteúdo* na caixa de diálogo de conversão para controlar como o Índice é gerado.

Nota: By default, when adding HTML files, calibre follows links in the files in *depth first* order. This means that if file A.html links to B.html and C.html and D.html, but B.html also links to D.html, then the files will be in the order A.html, B.html, D.html, C.html. If instead you want the order to be A.html, B.html, C.html, D.html then you must tell calibre to add your files in *breadth first* order. Do this by going to *Preferences → Advanced → Plugins → File type* and customizing the *HTML to ZIP* plugin.

9.1.7 O EPUB que criei com o Calibre não é válido?

calibre does not guarantee that an EPUB produced by it is valid. The only guarantee it makes is that if you feed it valid XHTML 1.1 + CSS 2.1 it will output a valid EPUB. calibre tries hard to ensure that EPUBs it produces actually work as intended on a wide variety of devices, a goal that is incompatible with producing valid EPUBs, and one that is far more important to the vast majority of its users. If you need a tool that always produces valid EPUBs, calibre is not for you. This means, that if you want to send a calibre produced EPUB to an online store that uses an EPUB validity checker, you have to make sure that the EPUB is valid yourself, calibre will not do it for you – in other words you must feed calibre valid XHTML + CSS as the input documents.

9.1.8 Como faço para usar alguns dos recursos avançados das ferramentas de conversão?

You can get help on any individual feature of the converters by mousing over it in the GUI or running `ebook-convert dummy.html .epub -h` at a terminal. A good place to start is to look at the following demo file that demonstrates some of the advanced features html-demo.zip[46].

9.2 Integração de dispositivos

Conteúdos

- *Que dispositivos é que o Calibre suporta?* (página 129)
- *Como posso ajudar para que meu dispositivo seja suportado pelo calibre?* (página 129)
- *O meu dispositivo não está a ser detetado pelo Calibre?* (página 129)
- *Meu dispositivo não é padrão ou é incomum. O que posso fazer para me conectar a ele?* (página 130)
- *How do I use calibre with my iPad/iPhone/iPod touch?* (página 130)
- *How do I use calibre with my Android phone/tablet or Kindle Fire HD?* (página 131)

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

- *Can I access my calibre books using the web browser in my Kindle or other reading device?* (página 132)
- *I cannot send emails using calibre?* (página 132)
- *My device is getting mounted read-only in Linux, so calibre cannot connect to it?* (página 133)
- *Why does calibre not support collections on the Kindle or shelves on the Nook?* (página 133)
- *I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?* (página 134)
- *Covers for books I send to my e-ink Kindle show up momentarily and then are replaced by a generic cover?* (página 134)
- *As capas dos meus arquivos MOBI deixaram de aparecer no Kindle para PC/Android/iPad, etc.* (página 135)
- *I transferred some books to my Kindle using calibre and they did not show up?* (página 135)

9.2.1 Que dispositivos é que o Calibre suporta?

calibre can directly connect to all the major (and most of the minor) e-book reading devices, smartphones, tablets, etc. In addition, using the *Connect to folder* function you can use it with any e-book reader that exports itself as a USB disk. Finally, you can connect wirelessly to any device that has a web browser using the calibre Content server.

9.2.2 Como posso ajudar para que meu dispositivo seja suportado pelo calibre?

If your device appears as a USB disk to the operating system, adding support for it to calibre is very easy. We just need some information from you:

- Lista completa de formatos de ebook suportados pelo seu dispositivo.
- Is there a special folder on the device in which all e-book files should be placed? Also does the device detect files placed in sub-folders?
- We also need information about your device that calibre will collect automatically. First, if your device supports SD cards, insert them. Then connect your device to the computer. In calibre go to *Preferences → Miscellaneous* and click the «Debug device detection» button. This will create some debug output. Copy it to a file and repeat the process, this time with your device disconnected from your computer.
- Send both the above outputs to us with the other information and we will write a device driver for your device.

Once you send us the output for a particular operating system, support for the device in that operating system will appear in the next release of calibre. To send us the output, open a bug report and attach the output to it. See how to report bugs[47].

9.2.3 O meu dispositivo não está a ser detetado pelo Calibre?

Siga estes passos para encontrar o problema:

- Make sure that you are connecting only a single device to your computer at a time. Do not have another calibre supported device like an iPhone/iPad etc. at the same time.
- If you are connecting an Apple iDevice (iPad, iPod Touch, iPhone), Apple no longer allows third party software to connect to their devices using a USB cable. Instead use a wireless connection, via the calibre Content server.
- If you are connecting a Kindle Fire or other Android device, read the note under *Usar um cabo USB* (página 131).

[47] https://calibre-ebook.com/pt/bugs

- On macOS if you get permission errors when connecting a device to calibre, you can fix that by looking under *System Preferences > Security and Privacy > Privacy > Files and Folders*.
- Make sure you are running the latest version of calibre (currently 6.14.1). The latest version can always be downloaded from the calibre website[48]. You can tell what version of calibre you are currently running by looking at the bottom line of the main calibre window.
- Ensure your operating system is seeing the device. That is, the device should show up in Windows Explorer (in Windows) or Finder (in macOS).
- In calibre, go to *Preferences → Ignored Devices* and check that your device is not being ignored
- Se todos os passos acima falharem, entre em *Preferências → Avançadas → Diversos* e clique *Depurar deteção de dispositivo* certificando-se de que o seu dispositivo está ligado. Publique a informação resultante como um problema no Acompanhamento de erros Calibre[49].

9.2.4 Meu dispositivo não é padrão ou é incomum. O que posso fazer para me conectar a ele?

In addition to the *Connect to folder* function found under the *Connect/share* button, calibre provides a `User defined` device plugin that can be used to connect to any USB device that shows up as a disk drive in your operating system. Note: on Windows, the device must have a drive letter for calibre to use it. See the device plugin `Preferences -> Plugins -> Device plugins -> User defined` and `Preferences -> Miscellaneous -> Get information to setup the user defined device` for more information. Note that if you are using the user defined plugin for a device normally detected by a builtin calibre plugin, you must disable the builtin plugin first, so that your user defined plugin is used instead.

9.2.5 How do I use calibre with my iPad/iPhone/iPod touch?

An easy way to browse your calibre collection from your Apple device is by using *O servidor de conteúdos Calibre* (página 109), which makes your collection available over the net. First perform the following steps in calibre

- Set the Preferred Output Format in calibre to EPUB (The output format can be set under *Preferences → Interface → Behavior*)
- Set the output profile to iPad (this will work for iPhone/iPods as well), under *Preferences → Conversion → Common options → Page setup*
- Convert the books you want to read on your iDevice to EPUB format by selecting them and clicking the *Convert* button.
- Turn on the Content server by clicking the *Connect/share* button and leave calibre running. You can also tell calibre to automatically start the Content server via *Preferences → Sharing → Sharing over the net*.

The Content server allows you to read books directly in Safari itself. In addition, there are many apps for your iDevice that can connect to the calibre Content server. Examples include: Marvin, Mapleread and iBooks itself.

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

Usar o servidor de conteúdos

Start the Safari browser and type in the IP address and port of the computer running the calibre server, like this:

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

Replace `192.168.1.2` with the local IP address of the computer running calibre. See *O servidor de conteúdos Calibre* (página 109) for details on running the server and finding out the right IP address to use.

You will see a list of books in Safari, tap on any book and you will be given the option to either download it, or read it in the browser itself. If you choose to download it, Safari will ask you if you want to open it with iBooks.

Many reading apps support browsing the calibre library directly via its OPDS support. In such apps you can go to the online catalog screen and add the IP address of the calibre server to browse and download books from your calibre library within the app.

9.2.6 How do I use calibre with my Android phone/tablet or Kindle Fire HD?

There are two ways that you can connect your Android device to calibre. Using a USB cable – or wirelessly, over the air. The first step to using an Android device is installing an e-book reading application on it. There are many free and paid e-book reading applications for Android: Some examples (in no particular order): FBReader[50], Moon+[51], Mantano[52], Aldiko[53], Kindle[54].

Usar um cabo USB

Simply plug your device into the computer with a USB cable. calibre should automatically detect the device and then you can transfer books to it by clicking the *Send to device* button. Note that on macOS and Linux only a single program can connect to an Android device at a time, so make sure the device is not opened in the OS File manager, or the Android File Transfer utility, etc.

Nota: With newer Android devices, you might have to jump through a few hoops to get the connection working, as Google really does not want you to be independent of its cloud. First, unlock the screen before plugging in the USB cable. When you plugin in the USB cable you will get a popup notification. Make sure it says some thing like «Transferring Media files» or «MTP (Media Transfer mode)». If it does not, tap the notification, and change the mode to Media Transfer (MTP). You may need to restart calibre at this point in order for your device to be recognized. Finally, you might get a popup on the device every time calibre or the operating system actually tries to connect to it, asking for permission, tap OK.

Nota: With the Kindle Fire 8 or newer there is an icon that shows up when the USB cable is plugged in, showing that the device is charging. Tap that and switch the device to data transfer mode, and then start calibre, it should then be detected.

---

[50] https://play.google.com/store/apps/details?id=org.geometerplus.zlibrary.ui.android&hl=en
[51] https://play.google.com/store/apps/details?id=com.flyersoft.moonreader&hl=en
[52] https://play.google.com/store/apps/details?id=com.mantano.reader.android.lite&hl=en
[53] https://play.google.com/store/apps/details?id=com.aldiko.android&hl=en
[54] https://play.google.com/store/apps/details?id=com.amazon.kindle&feature=related_apps

Sem fios

calibre has a builtin web server, the *O servidor de conteúdos Calibre* (página 109). It makes your calibre collection available over the net. You can browse it on your device using a simple browser or a dedicated application. First perform the following steps in calibre:

- Set the *Preferred Output Format* in calibre to EPUB for normal Android devices or MOBI for Kindles (The output format can be set under *Preferences → Interface → Behavior*)
- Convert the books you want to read on your device to EPUB/MOBI format by selecting them and clicking the *Convert* button.
- Turn on the *Content server* in calibre's preferences and leave calibre running.

Now on your Android device, open the browser and browse to

http://192.168.1.2:8080/

Replace `192.168.1.2` with the local IP address of the computer running calibre. See *O servidor de conteúdos Calibre* (página 109) for details on running the server and finding out the right IP address to use.

You can now browse your book collection and download books from calibre to your device to open with whatever e-book reading software you have on your Android device.

Many reading apps support browsing the calibre library directly via its OPDS support. In such apps you can go to the online catalog screen and add the IP address of the calibre server to browse and download books from your calibre library within the app.

9.2.7 Can I access my calibre books using the web browser in my Kindle or other reading device?

calibre has a *Content server* that exports the books in calibre as a web page. See *O servidor de conteúdos Calibre* (página 109) for details.

Some devices, like the Kindle (1/2/DX), do not allow you to access port 8080 (the default port on which the content server runs). In that case, change the port in the calibre Preferences to 80. (On some operating systems, you may not be able to run the server on a port number less than 1024 because of security settings. In this case the simplest solution is to adjust your router to forward requests on port 80 to port 8080).

Also some devices do not have browsers advanced enough to run the app-like interface used by the Content server. For such devices, you can simply add `/mobile` to the server URL to get a simplified, non-JavaScript interface.

9.2.8 I cannot send emails using calibre?

Because of the large amount of spam in email, sending email can be tricky, as different mail servers use different strategies to block email. The most common problem is if you are sending email directly (without a mail relay) in calibre. Many servers (for example, Amazon) block email that does not come from a well known relay. The most robust way to setup email sending in calibre is to do the following:

- Criar uma conta GMX gratuita em GMX[55].
- Goto *Preferences → Sharing → Sharing books by email* in calibre and click the *Use GMX* button and fill in the information asked for.
- Log into your GMX account on the website and enable SMTP sending (*Settings->POP3 & IMAP->Send and receive emails via external program*)

[55] https://www.gmx.com

- calibre will then be able to use GMX to send the mail.
- If you are sending to your Kindle, remember to update the email preferences on your Amazon Kindle page to allow email sent from your GMX email address. Also note that Amazon does not allow email delivery of AZW3 and new style (KF8) MOBI files. Finally, Amazon recently started sending confirmation emails that you have to click on back to your GMX account before the book is actually delivered.

Even after doing this, you may have problems. One common source of problems is that some poorly designed antivirus programs block calibre from opening a connection to send email. Try adding an exclusion for calibre in your antivirus program.

Nota: Microsoft/GMX can disable your account if you use it to send large amounts of email. So, when using these services to send mail calibre automatically restricts itself to sending one book every five minutes. If you don't mind risking your account being blocked you can reduce this wait interval by going to *Preferences → Advanced → Tweaks* in calibre.

Nota: Google recently deliberately broke their email sending protocol (SMTP) support in an attempt to force everyone to use their web interface so they can show you more ads. They are trying to claim that SMTP is insecure, that is incorrect and simply an excuse. Use some other email provider instead.

Nota: If you are concerned about giving calibre access to your email account, simply create a new free email account with GMX or Outlook and use it only for calibre.

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

Linux kernels mount devices read-only when their filesystems have errors. You can repair the filesystem with:

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

Replace /dev/sdc with the path to the device node of your device. You can find the device node of your device, which will always be under /dev by examining the output of:

```
mount
```

9.2.10 Why does calibre not support collections on the Kindle or shelves on the Nook?

Neither the Kindle nor the Nook provide any way to manipulate collections over a USB connection. If you really care about using collections, I would urge you to sell your Kindle/Nook and get a Kobo. Only Kobo seems to understand that life is too short to be entering collections one by one on an e-ink screen

Note that in the case of the Kindle, there is a way to manipulate collections via USB, but it requires that the Kindle be rebooted *every time* it is disconnected from the computer, for the changes to the collections to be recognized. As such, it is unlikely that any calibre developers will ever feel motivated enough to support it. There is however, a calibre plugin that allows you to create collections on your Kindle from the calibre metadata. It is available from here[56].

[56] https://www.mobileread.com/forums/showthread.php?t=244202

Nota: Amazon have removed the ability to manipulate collections completely in their newer models, like the Kindle Touch and Kindle Fire, making even the above plugin useless, unless you root your Kindle and install custom firmware.

9.2.11 I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?

The Kobo has very buggy firmware. Connecting to it has been known to fail at random. Certain combinations of motherboard, USB ports/cables/hubs can exacerbate this tendency to fail. If you are getting an error when connecting to your touch with calibre try the following, each of which has solved the problem for *some* calibre users.

- Connect the Kobo directly to your computer, not via USB Hub
- Try a different USB cable and a different USB port on your computer
- Log out of the Kobo and log in again, this causes it to rebuild the database, fixing corrupted database errors.
- Try upgrading the firmware on your Kobo Touch to the latest
- Try resetting the Kobo (sometimes this cures the problem for a little while, but then it re-appears, in which case you have to reset again and again)
- Try only putting one or two books onto the Kobo at a time and do not keep large collections on the Kobo

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

This happens because of an Amazon bug. They try to download a cover for the book from their servers and when that fails, they replace the existing cover that calibre created with a generic cover. For details see this forum thread[57]. As of version 4.17, calibre has a workaround, where if you connect the Kindle to calibre after the covers have been destroyed by Amazon, calibre will restore them automatically. So in order to see the covers on your Kindle, you have to:

1) Enviar livro para o Kindle com o Calibre
2) Disconnect the Kindle and wait for Amazon to destroy the cover
3) Religar o Kindle ao Calibre

Note that this workaround only works for books sent with calibre 4.17 or later. Alternately, simply keep your Kindle in airplane mode, you don't really want Amazon knowing every book you read anyway. I encourage you to contact Amazon customer support and complain loudly about this bug. Maybe Amazon will listen.

Nota: If the workaround is not working for you make sure you Kindle firmware is at least version 5.12.5, released in April 2020.

[57] https://www.mobileread.com/forums/showthread.php?t=329945

9.2.13 As capas dos meus arquivos MOBI deixaram de aparecer no Kindle para PC/Android/iPad, etc.

This is caused by a bug in the Amazon software. You can work around it by going to *Preferences → Conversion → Output Options → MOBI output* and setting the *Enable sharing of book content* option. If you are reconverting a previously converted book, you will also have to enable the option in the conversion dialog for that individual book (as per book conversion settings are saved and take precedence).

Note that doing this will mean that the generated MOBI will show up under personal documents instead of Books on the Kindle Fire and Amazon whispersync will not work, but the covers will. It's your choice which functionality is more important to you. I encourage you to contact Amazon and ask them to fix this bug.

The bug in Amazon's software is that when you put a MOBI file on a Kindle, unless the file is marked as a Personal document, Amazon assumes you bought the book from it and tries to download the cover thumbnail for it from its servers. When the download fails, it refuses to fallback to the cover defined in the MOBI file. This is likely deliberate on Amazon's part to try to force authors to sell only through them. In other words, the Kindle only displays covers for books marked as Personal Documents or books bought directly from Amazon.

If you send a MOBI file to an e-ink Kindle with calibre using a USB connection, calibre works around this Amazon bug by uploading a cover thumbnail itself. However, that workaround is only possible when using a USB connection and sending with calibre. Note that if you send using email, Amazon will automatically mark the MOBI file as a Personal Document and the cover will work, but the book will show up in Personal Documents.

9.2.14 I transferred some books to my Kindle using calibre and they did not show up?

Books sent to the Kindle only show up on the Kindle after they have been *indexed* by the Kindle. This can take some time. If the book still does not show up after some time, then it is likely that the Kindle indexer crashed. Sometimes a particular book can cause the indexer to crash. Unfortunately, Amazon has not provided any way to deduce which book is causing a crash on the Kindle. Your only recourse is to either reset the Kindle, or delete all files from its memory using Windows Explorer (or whatever file manager you use) and then send the books to it again, one by one, until you discover the problem book. Once you have found the problem book, delete it off the Kindle and do a MOBI to MOBI or MOBI to AZW3 conversion in calibre and then send it back. This will most likely take care of the problem.

9.3 Library management

Conteúdos

- *Onde é que os ficheiros dos livros são guardados?* (página 136)
- *How does calibre manage author names and sorting?* (página 136)
- *Why doesn't calibre let me store books in my own folder structure?* (página 137)
- *Why doesn't calibre have a column for foo?* (página 138)
- *Posso ter uma coluna para mostrar os formatos ou o ISBN?* (página 138)
- *Como mover os meus dados Calibre de um computador para outro?* (página 138)
- *A lista de livros no Calibre está vazia!* (página 139)
- *I am getting errors with my calibre library on a networked drive/NAS?* (página 139)

9.3.1 Onde é que os ficheiros dos livros são guardados?

When you first run calibre, it will ask you for a folder in which to store your books. Whenever you add a book to calibre, it will copy the book into that folder. Books in the folder are nicely arranged into sub-folders by Author and Title. Note that the contents of this folder are automatically managed by calibre, **do not** add any files/folders manually to this folder, as they may be automatically deleted. If you want to add a file associated to a particular book, use the top right area of *Edit metadata* dialog to do so. Then, calibre will automatically put that file into the correct folder and move it around when the title/author changes.

Metadata about the books is stored in the file `metadata.db` at the top level of the library folder. This file is a sqlite database. When backing up your library make sure you copy the entire folder and all its sub-folders.

The library folder and all its contents make up what is called a calibre library. You can have multiple such libraries. To manage the libraries, click the calibre icon on the toolbar. You can create new libraries, remove/rename existing ones and switch between libraries easily.

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

9.3.2 How does calibre manage author names and sorting?

Author names are complex, especially across cultures, see this note[58] for some of the complexities. calibre has a very flexible strategy for managing author names. The first thing to understand is that books and authors are separate entities in calibre. A book can have more than one author, and an author can have more than one book. You can manage the authors of a book by the edit metadata dialog. You can manage individual authors by right clicking on the author in the Tag browser on the left of the main calibre window and selecting *Manage authors*. Using this dialog you can change the name of an author and also how that name is sorted. This will automatically change the name of the author in all the books of that author. When a book has multiple authors, separate their names using the & character.

Now coming to author name sorting:

- When a new author is added to calibre (this happens whenever a book by a new author is added), calibre automatically computes a sort string for both the book and the author.
- Authors in the Tag browser are sorted by the sort value for the **authors**. Remember that this is different from the Author sort field for a book.
- By default, this sort algorithm assumes that the author name is in `First name Last name` format and generates a `Last name, First name` sort value.
- You can change this algorithm by going to *Preferences → Advanced → Tweaks* and setting the *author_sort_copy_method* tweak.
- You can force calibre to recalculate the author sort values for every author by right clicking on any author and selecting *Manage authors*, then pushing the *Recalculate all author sort values* button. Do this after you have set the author_sort_copy_method tweak to what you want.
- You can force calibre to recalculate the author sort values for all books by using the bulk metadata edit dialog (select all books and click edit metadata, check the *Automatically set author sort* checkbox, then press OK).
- When recalculating the author sort values for books, calibre uses the author sort values for each individual author. Therefore, ensure that the individual author sort values are correct before recalculating the books" author sort values.
- You can control whether the Tag browser display authors using their names or their sort values by setting the *categories_use_field_for_author_name* tweak in *Preferences → Advanced → Tweaks*

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

Note that you can set an individual author's sort value to whatever you want using *Manage authors*. This is useful when dealing with names that calibre will not get right, such as complex multi-part names like Miguel de Cervantes Saavedra or when dealing with Asian names like Sun Tzu.

With all this flexibility, it is possible to have calibre manage your author names however you like. For example, one common request is to have calibre display author names LN, FN. To do this, and if the note below does not apply to you, then:

- Set the `author_sort_copy_method` tweak to `copy` as described above.
- Restart calibre. Do not change any book metadata before doing the remaining steps.
- Change all author names to LN, FN using the Manage authors dialog.
- Depois de alterar todos os autores, primia o botão *Recalcular todos os valores de alfabetação de autores*
- Press OK, at which point calibre will change the authors in all your books. This can take a while.

Nota:

When changing from FN LN to LN, FN, it is often the case that the values in author_sort are already in LN, FN format. If this is your case, then do the following:

- Set the `author_sort_copy_method` tweak to `copy` as described above.
- Restart calibre. Do not change any book metadata before doing the remaining steps.
- Open the Manage authors dialog. Press the `copy all author sort values to author` button.
- Check through the authors to be sure you are happy. You can still press Cancel to abandon the changes. Once you press OK, there is no undo.
- Press OK, at which point calibre will change the authors in all your books. This can take a while.

9.3.3 Why doesn't calibre let me store books in my own folder structure?

The whole point of calibre's library management features is that they provide a search and sort based interface for locating books that is *much* more efficient than any possible folder scheme you could come up with for your collection. Indeed, once you become comfortable using calibre's interface to find, sort and browse your collection, you won't ever feel the need to hunt through the files on your disk to find a book again. By managing books in its own folder structure of Author -> Title -> Book files, calibre is able to achieve a high level of reliability and standardization. To illustrate why a search/tagging based interface is superior to folders, consider the following. Suppose your book collection is nicely sorted into folders with the following scheme:

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

Now this makes it very easy to find for example all science fiction books by Isaac Asimov in the Foundation series. But suppose you want to find all unread science fiction books. There's no easy way to do this with this folder scheme, you would instead need a folder scheme that looks like:

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

In calibre, you would instead use tags to mark genre and read status and then just use a simple search query like `tag:scifi and not tag:read`. calibre even has a nice graphical interface, so you don't need to learn its search language instead you can just click on tags to include or exclude them from the search.

To those of you that claim that you need access to the filesystem, so that you can have access to your books over the network, calibre has an excellent Content server that gives you access to your calibre library over the net.

Se está preocupado com a possibilidade de um dia o Calibre ser descontinuado, deixando todos os seus livros encalhados na sua estrutura de pastas, explore a poderosa funcionalidade do Calibre *Exportar para disco* que lhe permite exportar todos os seus ficheiros para uma estrutura de pastas com uma complexidade personalizada e baseada nos respetivos metadados.

Finally, the reason there are numbers at the end of every title folder, is for *robustness*. That number is the id number of the book record in the calibre database. The presence of the number allows you to have multiple records with the same title and author names. It is also part of what allows calibre to magically regenerate the database with all metadata if the database file gets corrupted. Given that calibre's mission is to get you to stop storing metadata in filenames and stop using the filesystem to find things, the increased robustness afforded by the id numbers is well worth the uglier folder names.

If you are still not convinced, then I'm afraid calibre is not for you. Look elsewhere for your book cataloguing needs. Just so we're clear, **this is not going to change**. Kindly do not contact us in an attempt to get us to change this.

9.3.4 Why doesn't calibre have a column for foo?

calibre is designed to have columns for the most frequently and widely used fields. In addition, you can add any columns you like. Columns can be added via *Preferences → Interface → Add your own columns*. Watch the tutorial UI Power tips[59] to learn how to create your own columns, or read this blog post[60].

You can also create «virtual columns» that contain combinations of the metadata from other columns. In the add column dialog use the *Quick create* links to easily create columns to show the book ISBN or formats. You can use the powerful calibre template language to do much more with columns. For more details, see *A linguagem de modelos do Calibre* (página 157).

9.3.5 Posso ter uma coluna para mostrar os formatos ou o ISBN?

Yes, you can. Follow the instructions in the answer above for adding custom columns.

9.3.6 Como mover os meus dados Calibre de um computador para outro?

Você pode exportar todos os dados do Calibre (livros, configurações e plugins) e depois fazer a sua importação para outro computador. Primeiro, vamos ver como exportar os dados:

- Right click the calibre icon in the main calibre toolbar and select *Export/import all calibre data*. Note that if there is currently a device connected, this menu option will not be available – so, disconnect any connected devices. Then click the button labelled *Export all your calibre data*. You will see a list of all your calibre libraries. Click OK and choose an empty folder somewhere on your computer. The exported data will be saved in this folder. Simply copy this folder to your new computer and follow the instructions below to import the data.
- Install calibre on your new computer and run through the *Welcome wizard*, it does not matter what you do there, as you will be importing your old settings in the next step. You will now have an empty calibre, with just the *Getting Started* guide in your library. Once again, right click the calibre button and choose *Export/import all calibre data*. Then click the button labelled *Import previously exported data*. Select the folder with the exported data that you copied over earlier. You will now have a list of libraries you can import. Go through the list one by one, and select the new location for each library (a location is just an empty folder somewhere on your computer). Click OK. After the import completes, calibre will restart, with all your old libraries, settings and calibre plugins.

Nota: This import/export functionality is only available from calibre version 2.47 onwards. If you have an older version of calibre, or if you encounter problems with the import/export, you can just copy over your calibre library folder manually, as described in the next paragraph.

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

Simply copy the calibre library folder from the old to the new computer. You can find out what the library folder is by clicking the calibre icon in the toolbar. Choose the *Switch/create calibre library* action and you will see the path to the current calibre library.

Now on the new computer, start calibre for the first time. It will run the *Welcome wizard* asking you for the location of the calibre library. Point it to the previously copied folder. If the computer you are transferring to already has a calibre installation, then the *Welcome wizard* won't run. In that case, right-click the calibre icon in the toolbar and point it to the newly copied folder. You will now have two calibre libraries on your computer and you can switch between them by clicking the calibre icon on the toolbar. Transferring your library in this manner preserver all your metadata, tags, custom columns, etc.

9.3.7 A lista de livros no Calibre está vazia!

In order to understand why that happened, you have to understand what a calibre library is. At the most basic level, a calibre library is just a folder. Whenever you add a book to calibre, that book's files are copied into this folder (arranged into sub folders by author and title). Inside the calibre library folder, at the top level, you will see a file called metadata.db. This file is where calibre stores the metadata like title/author/rating/tags etc. for *every* book in your calibre library. The list of books that calibre displays is created by reading the contents of this metadata.db file.

There can be two reasons why calibre is showing a empty list of books:

- A pasta da sua biblioteca Calibre sofreu uma mudança de local. Isto pode acontecer se ela estava num disco externo e a letra da drive desse disco foi alterada, ou se você acidentalmente moveu a pasta. Neste caso, o Calibre não encontra sua biblioteca e então arranca com uma biblioteca vazia no lugar da biblioteca ausente. Para resolver isto, na barra de ferramentas do Calibre, clique com o botão direito no ícone de bibliotecas Calibre e selecione Alternar/criar biblioteca. Clique no pequeno ícone de pasta azul para selecionar o novo local da sua biblioteca Calibre e clique OK. Se não sabe qual o novo local, procure no seu computador pelo ficheiro `metadata.db`.
- Your metadata.db file was deleted/corrupted. In this case, you can ask calibre to rebuild the metadata.db from its backups. Right click the calibre icon in the calibre toolbar and select Library maintenance->Restore database. calibre will automatically rebuild metadata.db.

9.3.8 I am getting errors with my calibre library on a networked drive/NAS?

Não coloque a sua biblioteca Calibre numa drive de rede.

A filesystem is a complex beast. Most network filesystems lack various filesystem features that calibre uses. Some don't support file locking, some don't support hardlinking, some are just flaky. Additionally, calibre is a single user application, if you accidentally run two copies of calibre on the same networked library, bad things will happen. Finally, different OSes impose different limitations on filesystems, so if you share your networked drive across OSes, once again, bad things *will happen*.

Consider using the calibre Content server to make your books available on other computers. Run calibre on a single computer and access it via the Content server or a Remote Desktop solution.

If you must share the actual library, use a file syncing tool like DropBox or rsync instead of a networked drive. If you are using a file-syncing tool it is **essential** that you make sure that both calibre and the file syncing tool do not try to access the calibre library at the same time. In other words, **do not** run the file syncing tool and calibre at the same time.

Even with these tools there is danger of data corruption/loss, so only do this if you are willing to live with that risk. In particular, be aware that **Google Drive** is incompatible with calibre, if you put your calibre library in Google Drive, **you will suffer data loss**. See this thread[61] for details.

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

9.4 Diversos

Conteúdos

- *Amazon is stopping email delivery of MOBI files?* (página 140)
- *I want calibre to download news from my favorite news website.* (página 141)
- *Porque escolheu o nome Calibre?* (página 141)
- *Why does calibre show only some of my fonts on macOS?* (página 141)
- *O Calibre não arranca em Windows?* (página 141)
- *calibre freezes/crashes occasionally?* (página 142)
- *The calibre E-book viewer and Edit book tools do not work on Windows?* (página 143)
- *Using the viewer or doing any conversions results in a permission denied error on Windows* (página 143)
- *calibre is not starting/crashing on macOS?* (página 143)
- *I get only a black or white screen when running the calibre E-book viewer?* (página 144)
- *I downloaded the installer, but it is not working?* (página 144)
- *My antivirus program claims calibre is a virus/trojan?* (página 144)
- *How do I backup calibre?* (página 145)
- *How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?* (página 145)
- *I am getting a «Permission Denied» error?* (página 145)
- *Can I have the comment metadata show up on my reader?* (página 146)
- *How do I get calibre to use my HTTP proxy?* (página 146)
- *I want some feature added to calibre. What can I do?* (página 146)
- *Why doesn't calibre have an automatic update?* (página 146)
- *Como é o Calibre licenciado?* (página 147)
- *How do I run calibre from my USB stick?* (página 147)
- *How do I run parts of calibre like news download and the Content server on my own Linux server?* (página 147)

9.4.1 Amazon is stopping email delivery of MOBI files?

Amazon have announced[62] that they will stop accepting MOBI files emailed to `@kindle.com` email addresses. You can instruct calibre to send EPUB instead of MOBI by going to *Preferences → Sharing books by email* and then removing MOBI from the list of formats to send to your `@kindle.com` email address and adding EPUB instead.

Note however, that Amazon's EPUB intake is very flawed, they will reject a number of EPUB files that work everywhere else. In such cases you can try the following trick:

1. Convert the EPUB file to MOBI
2. Then convert the MOBI file back to EPUB and send the resulting EPUB file

[62] https://blog.the-ebook-reader.com/2022/05/03/amazon-dropping-mobi-support-on-send-to-kindle-apps/

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

Nota: If you were previously using email delivery of periodicals downloaded by calibre, you will be better off sending those by USB cable or downloading them from the calibre Content server via the Kindle's built-in browser. However, if you want to continue using email delivery you can try changing the output format in Preferences->Behavior to EPUB, then calibre will download the news in EPUB format. Whether Amazon will accept the EPUB or not is a whole other question.

9.4.2 I want calibre to download news from my favorite news website.

If you are reasonably proficient with computers, you can teach calibre to download news from any website of your choosing. To learn how to do this see *Adicionar o seu sítio de notícias favorito* (página 29).

Otherwise, you can request a particular news site by posting in the calibre Recipes forum[63].

9.4.3 Porque escolheu o nome Calibre?

A sua escolha:

- Converter And LIBRary for E-books
- Um produto de alto *calibre*
- A tribute to the SONY Librie which was the first e-ink based e-book reader
- Foi a escolha da minha mulher ;-)

calibre is pronounced as cal-i-ber *not* ca-li-bre. If you're wondering, calibre is the British/commonwealth spelling for caliber. Being Indian, that's the natural spelling for me.

9.4.4 Why does calibre show only some of my fonts on macOS?

calibre embeds fonts in e-book files it creates. E-book files support embedding only TrueType and OpenType (.ttf and .otf) fonts. Most fonts on macOS systems are in .dfont format, thus they cannot be embedded. calibre shows only TrueType and OpenType fonts found on your system. You can obtain many such fonts on the web. Simply download the .ttf/.otf files and add them to the Library/Fonts folder in your home folder.

9.4.5 O Calibre não arranca em Windows?

There can be several causes for this:

- If you get an error about calibre not being able to open a file because it is in use by another program, do the following:
  - Desinstalar o Calibre
  - Reiniciar o seu computador
  - Re-install calibre. But do not start calibre from the installation wizard.
  - Temporarily disable your antivirus program (disconnect from the Internet before doing so, to be safe)

[63] https://www.mobileread.com/forums/forumdisplay.php?f=228

  - Look inside the folder you chose for your calibre library. If you see a file named metadata.db, delete it.
  - Iniciar o Calibre
  - From now on you should be able to start calibre normally.

- If you get an error about a Python function terminating unexpectedly after upgrading calibre, first uninstall calibre, then delete the folders (if they exists) `C:\Program Files\Calibre` and `C:\Program Files\Calibre2`. Now re-install and you should be fine.
- If you get an error in the *Welcome wizard* on an initial run of calibre, try choosing a folder like `C:\library` as the calibre library (calibre sometimes has trouble with library locations if the path contains non-English characters, or only numbers, etc.)
- Try running it as administrator (Right click on the icon and select *Run as administrator*)

If it still won't launch, start a command prompt (press the Windows key and R; then type **`cmd.exe`** in the Run dialog that appears). At the command prompt type the following command and press Enter:

```
calibre-debug -g
```

Partilhe qualquer saída obtida numa mensagem de ajuda no Fórum[64].

9.4.6 calibre freezes/crashes occasionally?

There are several possible things I know of, that can cause this:

- You recently connected an external monitor or TV to your computer. In this case, whenever calibre opens a new window like the edit metadata window or the conversion dialog, it appears on the second monitor where you don't notice it and so you think calibre has frozen. Disconnect your second monitor and restart calibre.
- The following programs have been reported to cause crashes in calibre: If you are running any of these, close them before starting calibre, or uninstall them: *RoboForm*, *Logitech SetPoint Settings*, *Constant Guard Protection by Xfinity*, *Spybot*, *Killer Network Manager*, *Nahimic UI Interface*, *Acronis True Image*.
- You are using a Wacom branded USB mouse/tablet. There is an incompatibility between Wacom drivers and the graphics toolkit calibre uses. Try using a non-Wacom mouse.
- Em algumas versões de 64 bits do Windows existem programas de segurança/configurações que impedem a versão de 64 bits do Calibre de funcionar corretamente. Se está a usar a versão de 64 bits do calibre, tente mudar para a versão de 32 bits.
- If the crash happens when you are trying to copy text from the calibre E-book viewer, it is most likely caused by some clipboard monitoring/managing application you have running. Turn it off and you should be fine.
- If the crashes happen specifically when you are using a file dialog, like clicking on the *Add books* button or the *Save to Disk* button, then you have some software that has installed broken Shell extensions on your computer. Known culprits include: *SpiderOak*, *odrive sync* and *Dell Backup and Recovery* and *NetDrive*. If you have one of these, uninstall them and you will be fine. You can also use the NirSoft Shell Extension Viewer[65] to see what shell extensions are installed on your system and disable them individually, if you don't want to uninstall the full program. Remember to use «Restart Explorer» or reboot your computer after disabling the shell extensions.

If none of the above apply to you, then there is some other program on your computer that is interfering with calibre. First reboot your computer in safe mode, to have as few running programs as possible, and see if the crashes still happen. If they do not, then you know it is some program causing the problem. The most likely such culprit is a program that modifies other programs" behavior, such as an antivirus, a device driver, something like RoboForm (an automatic form filling app) or an assistive technology like Voice Control or a Screen Reader.

[64] https://www.mobileread.com/forums/forumdisplay.php?f=166
[65] https://www.nirsoft.net/utils/shexview.html

The only way to find the culprit is to eliminate the programs one by one and see which one is causing the issue. Basically, stop a program, run calibre, check for crashes. If they still happen, stop another program and repeat.

9.4.7 The calibre E-book viewer and Edit book tools do not work on Windows?

These two programs use hardware acceleration as they embed a version of the Chrome browser to render HTML. If they do not work it will be because of incompatibility with your system's GPU (graphics) drivers. Try updating these first, and reboot. If that does not fix it, you can set the `QTWEBENGINE_CHROMIUM_FLAGS` environment variable to the value `--disable-gpu` to turn off hardware acceleration. See this page[66] for details.

9.4.8 Using the viewer or doing any conversions results in a permission denied error on Windows

Something on your computer is preventing calibre from accessing its own temporary files. Most likely the permissions on your `Temp` folder are incorrect. Go to the folder file:*C:\Users\USERNAME\AppData\Local* in Windows Explorer and then right click on the file:*Temp* folder, select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

Some users have reported that running the following command in an Administrator Command Prompt fixed their permissions. To get an Administrator Command Prompt search for cmd.exe in the start menu, then right click on the command prompt entry and select *Run as administrator*. At the command prompt type the following command and press Enter:

```
icacls "%appdata%\..\Local\Temp" /reset /T
```

Alternately, you can run calibre as Administrator, but doing so will cause some functionality, such as drag and drop to not work.

Finally, some users have reported that disabling UAC fixes the problem.

9.4.9 calibre is not starting/crashing on macOS?

One common cause of failures on macOS is the use of accessibility technologies that are incompatible with the graphics toolkit calibre uses. Try turning off VoiceOver if you have it on. Also go to System Preferences->System->Universal Access and turn off the setting for enabling access for assistive devices in all the tabs. Another cause can be some third party apps that modify system behavior, such as Smart Scroll.

You can obtain debug output about why calibre is not starting by running *Console.app*. Debug output will be printed to it. If the debug output contains a line that looks like:

```
Qt: internal: -108: Error ATSUMeasureTextImage text/qfontengine_mac.mm
```

then the problem is probably a corrupted font cache. You can clear the cache by following these instructions[67]. If that doesn't solve it, look for a corrupted font file on your system, in ~/Library/Fonts or the like. An easy way to check for corrupted fonts in macOS is to start the «Font Book» application, select all fonts and then in the File menu, choose «Validate fonts».

[66] https://doc.qt.io/qt-5/qtwebengine-debugging.html
[67] https://www.macworld.com/article/1139383/fontcacheclear.html

9.4.10 I get only a black or white screen when running the calibre E-book viewer?

This will be because of an incompatibility between Qt WebEngine, which the viewer uses to render and the GPU drivers on your system. First try upgrading the GPU drivers. If that does not help, you can try turning off hardware acceleration in Qt WebEngine by setting the environment variable `QTWEBENGINE_CHROMIUM_FLAGS` to the value `--disable-gpu`. See *Variáveis de ambiente* (página 282) for how to change environment variables.

9.4.11 I downloaded the installer, but it is not working?

Downloading from the Internet can sometimes result in a corrupted download. If the calibre installer you downloaded is not opening, try downloading it again. If re-downloading it does not work, download it from an alternate location[68]. If the installer still doesn't work, then something on your computer is preventing it from running.

- Try temporarily disabling your antivirus program (Microsoft Security Essentials, or Kaspersky or Norton or McAfee or whatever). This is most likely the culprit if the upgrade process is hanging in the middle.
- Similarly, if the installer is failing/rolling back and you have Microsoft PowerToys running, quit it.
- Try rebooting your computer and running a registry cleaner like Wise registry cleaner[69].
- Try a clean install. That is, uninstall calibre, delete `C:\Program Files\Calibre2` (or wherever you previously chose to install calibre). Then re-install calibre. Note that uninstalling does not touch your books or settings.
- Try downloading the installer with an alternate browser. For example if you are using Microsoft Edge, try using Firefox or Chrome instead.
- If you get an error about a missing DLL on Windows, then most likely, the permissions on your temporary folder are incorrect. Go to the folder `C:\Users\USERNAME\AppData\Local` in Windows Explorer and then right click on the `Temp` folder and select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

If you still cannot get the installer to work and you are on Windows, you can use the calibre portable install[70], which does not need an installer (it is just a ZIP file).

9.4.12 My antivirus program claims calibre is a virus/trojan?

The first thing to check is that you are downloading calibre from the official website[71]. Make sure you are clicking the download links on the left, not the advertisements on the right. calibre is a very popular program and unscrupulous people try to setup websites offering it for download to fool the unwary.

If you have the official download and your antivirus program is still claiming calibre is a virus, then, your antivirus program is wrong. Antivirus programs use heuristics, patterns of code that «look suspicious» to detect viruses. It's rather like racial profiling. calibre is a completely open source product. You can actually browse the source code yourself (or hire someone to do it for you) to verify that it is not a virus. Please report the false identification to whatever company you buy your antivirus software from. If the antivirus program is preventing you from downloading/installing calibre, disable it temporarily, install calibre and then re-enable it.

[68] https://github.com/kovidgoyal/calibre/releases/latest
[69] https://www.wisecleaner.com
[70] https://calibre-ebook.com/pt/download_portable
[71] https://calibre-ebook.com/pt/download

9.4.13 How do I backup calibre?

The most important thing to backup is the calibre library folder, that contains all your books and metadata. This is the folder you chose for your calibre library when you ran calibre for the first time. You can get the path to the library folder by clicking the calibre icon on the main toolbar. You must backup this complete folder with all its files and sub-folders.

You can switch calibre to using a backed up library folder by simply clicking the calibre icon on the toolbar and choosing your backup library folder. A backed up library folder backs up your custom columns and saved searches as well as all your books and metadata.

If you want to backup the calibre configuration/plugins, you have to backup the config folder. You can find this config folder via *Preferences → Miscellaneous*. Note that restoring configuration folders is not officially supported, but should work in most cases. Just copy the contents of the backup folder into the current configuration folder to restore.

9.4.14 How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?

Most purchased EPUB books have *DRM* (página 365). This prevents calibre from opening them. You can still use calibre to store and transfer them to your e-book reader. First, you must authorize your reader on a Windows machine with Adobe Digital Editions. Once this is done, EPUB books transferred with calibre will work fine on your reader. When you purchase an epub book from a website, you will get an «.acsm» file. This file should be opened with Adobe Digital Editions, which will then download the actual «.epub» e-book. The e-book file will be stored in the folder «My Digital Editions», from where you can add it to calibre.

9.4.15 I am getting a «Permission Denied» error?

A permission denied error can occur because of many possible reasons, none of them having anything to do with calibre.

- You can get permission denied errors if you are using an SD card with write protect enabled.
- On macOS if you get permission errors when connecting a device to calibre, you can fix that by looking under *System Preferences > Security and Privacy > Privacy > Files and Folders*.
- If you, or some program you used changed the file permissions of the files in question to read only.
- If there is a filesystem error on the device which caused your operating system to mount the filesystem in read only mode or mark a particular file as read only pending recovery.
- If the files have their owner set to a user other than you.
- If your file is open in another program.
- If the file resides on a device, you may have reached the limit of a maximum of 256 files in the root of the device. In this case you need to reformat the device/sd card referred to in the error message with a FAT32 filesystem, or delete some files from the SD card/device memory.

You will need to fix the underlying cause of the permissions error before resuming to use calibre. Read the error message carefully, see what file it points to and fix the permissions on that file or its containing folders.

9.4.16 Can I have the comment metadata show up on my reader?

Most readers do not support this. You should complain to the manufacturer about it and hopefully if enough people complain, things will change. In the meantime, you can insert the metadata, including comments into a «Jacket page» at the start of the e-book, by using the option to «Insert metadata as page at start of book» during conversion. The option is found in the *Structure detection* section of the conversion settings. Note that for this to have effect you have to *convert* the book. If your book is already in a format that does not need conversion, you can convert from that format to the same format.

Another alternative is to create a catalog in e-book form containing a listing of all the books in your calibre library, with their metadata. Click-and-hold the *Convert* button to access the catalog creation tool. And before you ask, no you cannot have the catalog «link directly to» books on your reader.

9.4.17 How do I get calibre to use my HTTP proxy?

By default, calibre uses whatever proxy settings are set in your OS. Sometimes these are incorrect, for example, on Windows if you don't use Microsoft Edge then the proxy settings may not be up to date. You can tell calibre to use a particular proxy server by setting the `http_proxy` and `https_proxy` environment variables. The format of the variable is: `http://username:password@servername` you should ask your network administrator to give you the correct value for this variable. Note that calibre only supports HTTP proxies not SOCKS proxies. You can see the current proxies used by calibre in Preferences->Miscellaneous.

9.4.18 I want some feature added to calibre. What can I do?

Tem duas alternativas:

1. Create a patch by hacking on calibre and send it to me for review and inclusion. See Development[72].
2. Open a bug requesting the feature[73]. Remember that while you may think your feature request is extremely important/essential, calibre developers might not agree. Fortunately, calibre is open source, which means you always have the option of implementing your feature yourself, or hiring someone to do it for you. Furthermore, calibre has a comprehensive plugin architecture, so you might be able to develop your feature as a plugin, see *Criar os seus próprios plugins para aumentar as funcionalidades do Calibre* (página 219).

9.4.19 Why doesn't calibre have an automatic update?

Por vários motivos:

- *There is no need to update every week*. If you are happy with how calibre works turn off the update notification and be on your merry way. Check back to see if you want to update once a year or so. There is a check box to turn off the update notification, on the update notification itself.
- Atualmente, os downloads de Calibre usam *cerca de 150 TB de largura de banda por mês <dynamic/downloads>*. A implementação de atualizações automáticas aumentaria isto ainda mais e acabaria custando milhares de euros por mês, que alguém teria de pagar.
- If I implement a dialog that downloads the update and launches it, instead of going to the website as it does now, that would save the most ardent calibre updater, *at most five clicks a week*. There are far higher priority things to do in calibre development.
- If you really, really hate downloading calibre every week but still want to be up to the latest, I encourage you to run from source, which makes updating trivial. Instructions are *available here* (página 337).

[72] https://calibre-ebook.com/pt/get-involved
[73] https://calibre-ebook.com/pt/bugs

- Existem atualizadores automáticos de terceiros para o Calibre criados por utilizadores Calibre no fórum do Calibre[74].

9.4.20 Como é o Calibre licenciado?

O Calibre é licenciado sob os termos da GNU General Public License v3 (uma licença de código aberto). Isto quer dizer que você é livre de redistribuir o Calibre desde que torne disponível o código fonte. Assim, se pretender incluir o Calibre num CD com o seu produto, tem também de incluir o código fonte do Calibre nesse CD. O código fonte pode ser descarregado em[75]. Você tem toda a liberdade para usar os resultados das conversões produzidas pelo Calibre de qualquer modo que pretenda. Não pode usar quer o código quer as bibliotecas do Calibre no seu software se este não for distribuído como software de código aberto. Para detalhes, ver The GNU GPL v3[76].

9.4.21 How do I run calibre from my USB stick?

Está disponível uma versão portátil do Calibre aqui:[77].

9.4.22 How do I run parts of calibre like news download and the Content server on my own Linux server?

First, you must install calibre onto your Linux server. If your server is using a modern Linux distribution, you should have no problems installing calibre onto it.

Nota: calibre needs GLIBC >= 2.31 and libstdc++ >= 6.0.28. If you have an older server, you will either need to compile these from source, or use calibre 3.48 which requires GLIBC >= 2.17 or 2.85.1 which requires GLIBC >= 2.13 or calibre 1.48 which requires only GLIBC >= 2.10. In addition, although the calibre command line utilities do not need a running X server, some of them do require the X server libraries to be installed on your system. This is because of Qt, which is used for various image processing tasks, and links against these libraries. If you get an ImportError about some Qt modules, you are likely missing some X libraries. Typical candidates are: `libxcb-xinerama0`, `libegl1`, `libopengl0`.

You can run the calibre server via the command:

```
/opt/calibre/calibre-server /path/to/the/library/you/want/to/share
```

You can download news and convert it into an e-book with the command:

```
/opt/calibre/ebook-convert "Title of news source.recipe" outputfile.epub
```

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

You can email downloaded news with the command:

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

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

Finally, you can add downloaded news to the calibre library with:

[74] https://www.mobileread.com/forums/forumdisplay.php?f=238
[75] https://download.calibre-ebook.com
[76] https://www.gnu.org/licenses/gpl.html
[77] https://calibre-ebook.com/pt/download_portable

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

Remember to read the *Interface de linha de comandos* (página 295) section of the calibre User Manual to learn more about these, and other commands.

CAPÍTULO 10

Tutoriais

Aqui você irá encontrar tutoriais para poder começar a utilizar recursos mais avançados do Calibre, como XPath e modelos.

10.1 Gerir subgrupos de livros, por exemplo «género»

Some people wish to organize the books in their library into subgroups, similar to subfolders. The most commonly provided reason is to create genre hierarchies, but there are many others. One user asked for a way to organize textbooks by subject and course number. Another wanted to keep track of gifts by subject and recipient. This tutorial will use the genre example for the rest of this post.

Before going on, please note that we are not talking about folders on the hard disk. Subgroups are not file folders. Books will not be copied anywhere. calibre's library file structure is not affected. Instead, we are presenting a way to organize and display subgroups of books within a calibre library.

- *Configuração* (página 151)
- *A pesquisar* (página 153)
- *Restrições* (página 154)
- *Funções úteis de modelos* (página 154)

Os requisitos geralmente fornecidos para subgrupos, como géneros, são:

- A subgroup (e.g., a genre) must contain (point to) books, not categories of books. This is what distinguishes subgroups from calibre user categories.
- Um livro pode estar em vários subgrupos (géneros). Isto distingue subgrupos de pastas de ficheiros físicos.
- Os subgrupos (géneros) devem formar uma hierarquia; subgrupos podem conter subgrupos.

Tags give you the first two. If you tag a book with the genre then you can use the Tag browser (or search) to find the books with that genre, giving you the first. Many books can have the same tag(s), giving you the second. The problem is

that tags don't satisfy the third requirement. They don't provide a hierarchy.

The calibre hierarchy feature gives you the third – the ability to see the genres in a "tree" and the ability to easily search for books in genre or sub-genre. For example, assume that your genre structure is similar to the following:

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

By using the hierarchy feature, you can see these genres in the Tag browser in tree form, as shown in the screen image. In this example the outermost level (Genre) is a custom column that contains the genres. Genres containing sub-genres appear with a small triangle next to them. Clicking on that triangle will open the item and show the sub-genres, as you can see with History and Science Fiction.

Clicking on a genre can search for all books with that genre or children of that genre. For example, clicking on Science Fiction can give all three of the child genres, Alternate History, Military, and Space Opera. Clicking on Alternate History will give books in that genre, ignoring those in Military and Space Opera. Of course, a book can have multiple genres. If a book has both Space Opera and Military genres, then you will see that book if you click on either genre. Searching is discussed in more detail below.

Another thing you can see from the image is that the genre Military appears twice, once under History and once under

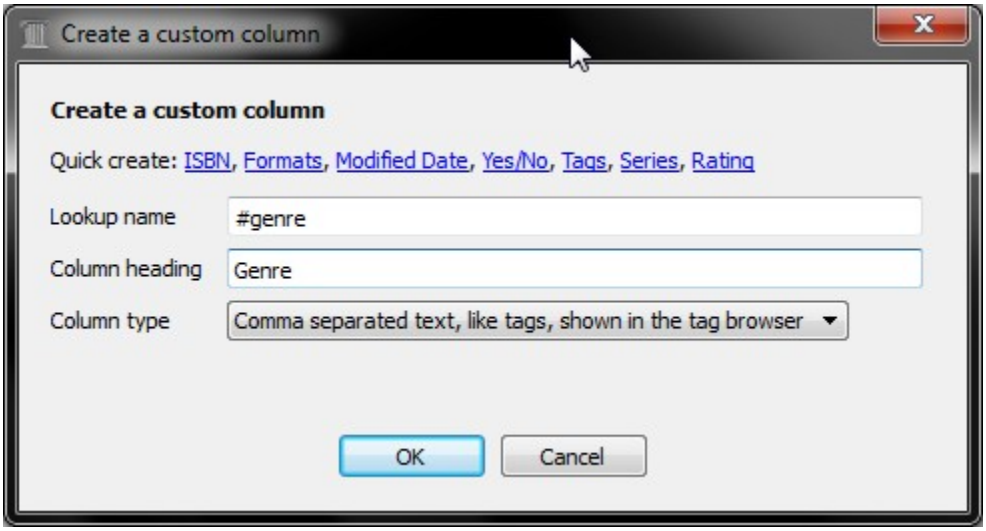
Science Fiction. Because the genres are in a hierarchy, these are two separate genres. A book can be in one, the other, or (doubtfully in this case) both. For example, the books in Winston Churchill's «The Second World War» could be in «History.Military». David Weber's Honor Harrington books could be in «Science Fiction.Military», and for that matter also in «Science Fiction.Space Opera.»

Once a genre exists, that is at least one book has that genre, you can easily apply it to other books by dragging the books from the library view onto the genre you want the books to have. You can also apply genres in the metadata editors; more on this below.

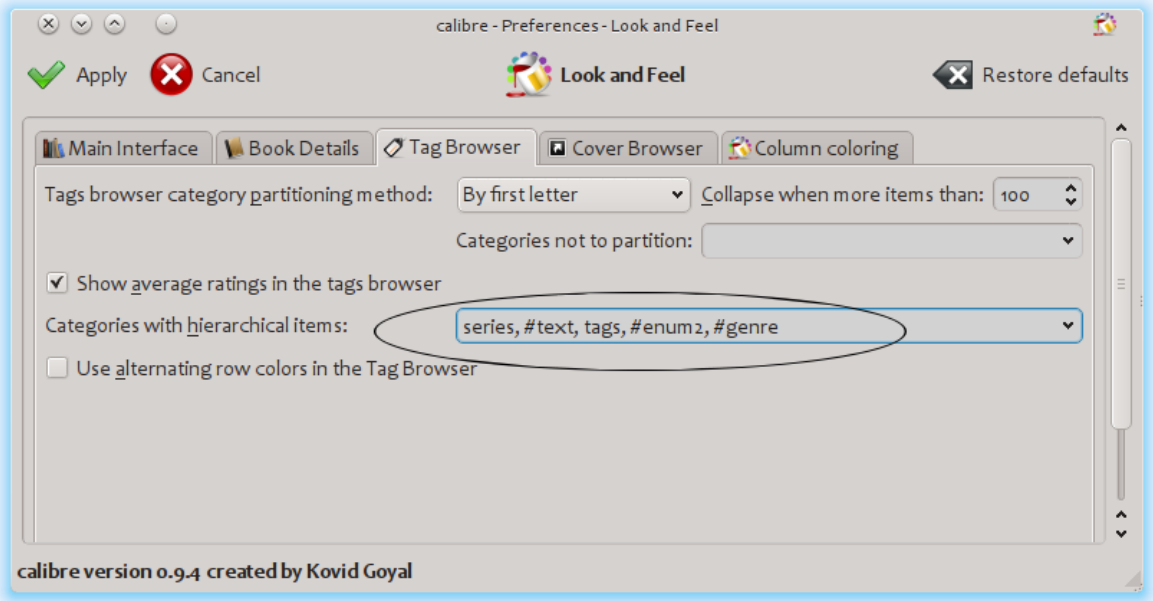
10.1.1 Configuração

By now, your question might be «How was all of this setup?» There are three steps: 1) create the custom column, 2) tell calibre that the new column is to be treated as a hierarchy, and 3) add genres.

You create the custom column in the usual way, using Preferences -> Add your own columns. This example uses «#genre» as the lookup name and «Genre» as the column heading. The column type is «Comma-separated text, like tags, shown in the Tag browser.»

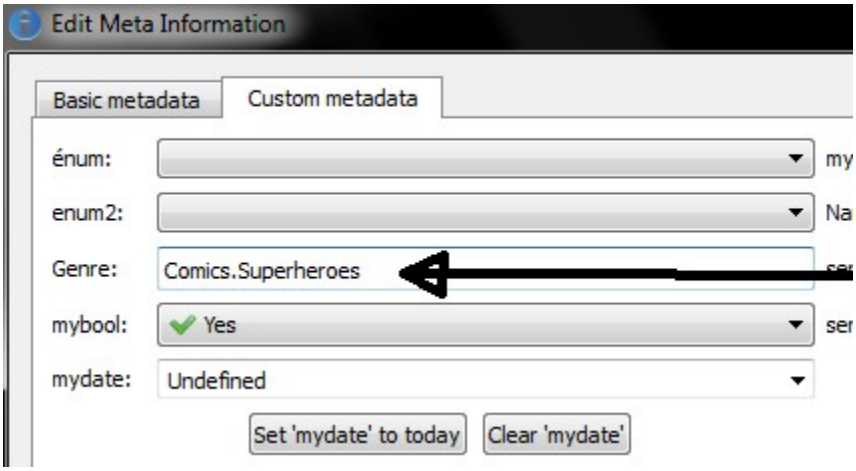


Then after restarting calibre, you must tell calibre that the column is to be treated as a hierarchy. Go to *Preferences* → *Look & feel* → *Tag browser* and enter the lookup name «#genre» into the «Categories with hierarchical items» box. Press *Apply*, and you are done with setting up.



At the point there are no genres in the column. We are left with the last step: how to apply a genre to a book. A genre does not exist in calibre until it appears on at least one book. To learn how to apply a genre for the first time, we must go into some detail about what a genre looks like in the metadata for a book.

A hierarchy of "things" is built by creating an item consisting of phrases separated by periods. Continuing the genre example, these items would «History.Military», «Mysteries.Vampire», «Science Fiction.Space Opera», etc. Thus to create a new genre, you pick a book that should have that genre, edit its metadata, and enter the new genre into the column you created. Continuing our example, if you want to assign a new genre «Comics» with a sub-genre «Superheroes» to a book, you would "edit metadata" for that (comic) book, choose the Custom metadata tab, and then enter «Comics.Superheroes» as shown in the following (ignore the other custom columns):

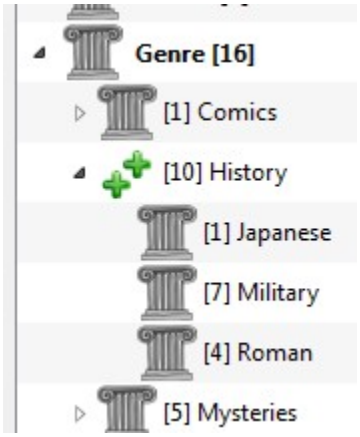


After doing the above, you see in the Tag browser:

From here on, to apply this new genre to a book (a comic book, presumably), you can either drag the book onto the genre, or add it to the book using edit metadata in exactly the same way as done above.

Nota: Hierarchical display only works if the Tag browser is set to sort items by name. This is the default and can be checked by clicking the *Configure* button at the bottom of the Tag browser.

10.1.2 A pesquisar



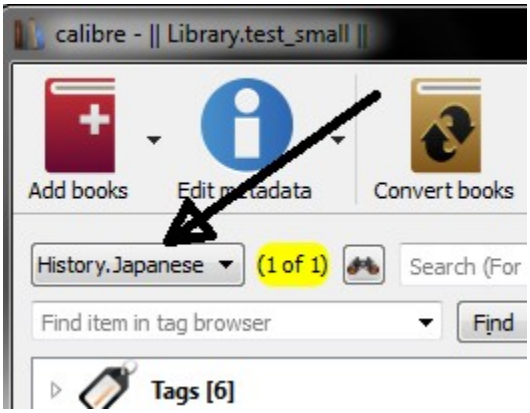
The easiest way to search for genres is using the Tag browser, clicking on the genre you wish to see. Clicking on a genre with children will show you books with that genre and all child genres. However, this might bring up a question. Just because a genre has children doesn't mean that it isn't a genre in its own right. For example, a book can have the genre «History» but not «History.Military». How do you search for books with only «History»?

The Tag browser search mechanism knows if an item has children. If it does, clicking on the item cycles through 5 searches instead of the normal three. The first is the normal green plus, which shows you books with that genre only (e.g., History). The second is a doubled plus (shown above), which shows you books with that genre and all sub-genres (e.g., History and History.Military). The third is the normal red minus, which shows you books without that exact genre. The fourth is a doubled minus, which shows you books without that genre or sub-genres. The fifth is back to the beginning, no mark, meaning no search.

10.1.3 Restrições

If you search for a genre then create a saved search for it, you can use the "restrict to" box to create a Virtual library of books with that genre. This is useful if you want to do other searches within the genre or to manage/update metadata for books in the genre. Continuing our example, you can create a Saved search named "History.Japanese" by first clicking on the genre Japanese in the Tag browser to get a search into the search field, entering History.Japanese into the saved search field, then pushing the «Save search» button (the green box with the white plus, on the right-hand side).

After creating the saved search, you can use it as a restriction.



10.1.4 Funções úteis de modelos

You might want to use the genre information in a template, such as with save to disk or send to device. The question might then be «How do I get the outermost genre name or names?» A calibre template function, subitems, is provided to make doing this easier.

For example, assume you want to add the outermost genre level to the save-to-disk template to make genre folders, as in «History/The Gathering Storm - Churchill, Winston». To do this, you must extract the first level of the hierarchy and add it to the front along with a slash to indicate that it should make a folder. The template below accomplishes this:

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

See *The template language* (página 157) for more information about templates and the `subitems()` function.

10.2 Tutorial XPath

Neste tutorial, você receberá uma sucinta introdução ao XPath[78] uma linguagem de consulta que pode ser usada para selecionar partes aleatórias de documentos **`HTML<https://en.wikipedia.org/wiki/HTML>`_** de forma segmentada. O *XPath* é um padrão muito difundido e uma consulta ao *Google* resultará em um grande volume de informações. Este tutorial, entretanto, se concentra no uso de tarefas relacionadas a livros eletrônicos, como encontrar cabeçalhos de capítulos em um documento *HTML* não estruturado.

Conteúdos

- *Selecionar por nome de etiqueta* (página 155)
- *Selecionar por atributos* (página 156)
- *Selecionar por conteúdo da etiqueta* (página 156)
- *Ebook de amostra* (página 156)
- *Funções nucleares do XPath* (página 157)

10.2.1 Selecionar por nome de etiqueta

A forma mais simples de seleção é selecionar etiquetas pelo nome. Por exemplo, suponha que você deseja selecionar todas as tags `<h2>` em um documento. A consulta XPath para isso é simplesmente:

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

The prefix *//* means *search at any level of the document*. Now suppose you want to search for `<span>` tags that are inside `<a>` tags. That can be achieved with:

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

Se você quiser buscar por tags em um determinado nível do documento, mude o prefixo:

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

Isto corresponderá apenas com `<p>Um livro eletrônico curto para demonstrar o uso da XPath. </p>` na :ref:`sample_ebook` mas não em qualquer uma das outras tags `<p>`. O prefixo `'h'` nos exemplos acima é necessário para combinar etiquetas *XHTML*. Isto porque internamente o *Calibre* representa todo o conteúdo *XHTML*. As tags *XHTML* têm um *namespace* e `h:` é o prefixo de *namespace* para tags *HTML*.

Agora suponha que você queira selecionar ambas as etiquetas `<h1>` e `<h2>`. Para fazer isso, é necessário uma construção XPath chamada *predicado*. Um *predicate* é simplesmente um teste usado para selecionar tags. Testes podem ser arbitrariamente poderosos e, no decorrer do tutorial, você verá mais exemplos poderosos. Cria-se um predicado ao colocar a expressão teste entre colchetes:

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

Existem vários novos recursos nesta expressão do XPath. A primeira é o uso do caractere universal `*`. Significa *corresponder a qualquer tag*. Agora veja a expressão de teste``name()="h1" ou name()="h2". `:term:`name()` é um exemplo de uma *função interna*. Ela simplesmente resulta no nome da tag. Então, usando-a, podemos selecionar etiquetas cujos nomes são `h1` ou `h2`. Observe

[78] https://en.wikipedia.org/wiki/XPath

que a função :term:`name()` ignora namespaces, de modo a que não há necessidade para o prefixo ``h:. O XPath tem várias funções internas úteis. Mais algumas serão apresentadas neste tutorial.

10.2.2 Selecionar por atributos

Para selecionar tags com base em seu conteúdo, é necessário o uso de predicados:

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

Aqui, o operador `@` refere-se aos atributos da tag. Você pode usar alguma das *funções internas do XPath* (página 157) para executar correspondências mais sofisticadas em valores de atributos.

10.2.3 Selecionar por conteúdo da etiqueta

Usando XPath, você também pode selecionar tags baseado nos textos que elas contêm. A melhor maneira para fazer isso é usar o poder das *expressões regulares* através da função interna *re:test()*:

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

Aqui o operador `.` refere-se ao conteúdo da tag, assim como o operador `@` refere-se aos seus atributos.

10.2.4 Ebook de amostra

```
<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 Funções nucleares do XPath

name()
O nome da etiqueta atual

contains()
`contains(s1, s2)` devolve *true* se s1 contém s2.

re:test()
`re:test(src, pattern, flags)` retorna `True` se a string `src` corresponde à expressão regular `pattern` Uma *flag* particularmente útil é `i`, que torna a correspondência insensível a maiúsculas e minúsculas. Uma boa introdução sobre a sintaxe para expressões regulares pode ser encontrada em sintaxe regexp[79]

10.3 A linguagem de modelos do Calibre

The calibre template language is a calibre-specific language used throughout calibre for tasks such as specifying file paths, formatting values, and computing the value for user-specified columns. Examples:

- Specify the folder structure and file names when saving files from the calibre library to the disk or e-book reader.
- Define rules for adding icons and colors to the calibre book list.
- Define *virtual columns* that contain data from other columns.
- Advanced library searching.
- Advanced metadata search and replace.

The language is built around the notion of a *template*, which specifies which book metadata to use, computations on that metadata, and how it is to be formatted.

10.3.1 Temas básicos

A basic template consists one or more `template expressions`. A `template expression` consists of text and names in curly brackets (`{}`) that is replaced by the corresponding metadata from the book being processed. For example, the default template in calibre used for saving books to device has 4 `template expressions`:

```
{author_sort}/{title}/{title} - {authors}
```

For the book «The Foundation» by «Isaac Asimov» the will become:

```
Asimov, Isaac/The Foundation/The Foundation - Isaac Asimov
```

The slashes are not `template expressions` because they are in between in `{}`. Such text is left where it appears. For example, if the template is:

```
{author_sort} Some Important Text {title}/{title} - {authors}
```

then for «The Foundation» the template produces:

```
Asimov, Isaac Some Important Text The Foundation/The Foundation - Isaac Asimov
```

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

A `template expression` can access all the metadata available in calibre, including custom columns (columns you create yourself), by using a column's `lookup name`. To find the lookup name for a *column* (sometimes called *fields*), hover your mouse over the column header in calibre's book list. Lookup names for custom columns always begin with `#`. For series type columns there is an additional field named `#lookup name_index` that is the series index for that book in the series. For example, if you have a custom series column named `#myseries`, there will also be a column named `#myseries_index`. The standard series column's index is named `series_index`.

In addition to the standard column based fields, you also can use:

- `{formats}` - A list of formats available in the calibre library for a book
- `{identifiers:select(isbn)}` - The ISBN of the book

If the metadata for the field for a given book is not defined then the field in the template is replaced by the empty string (`''`). For example, consider the following template:

```
{author_sort}/{series}/{title} {series_index}
```

If Asimov's book «Second Foundation» is in the series «Foundation» then the template produces:

```
Asimov, Isaac/Foundation/Second Foundation 3
```

If a series has not been entered for the book then the template produces:

```
Asimov, Isaac/Second Foundation
```

The template processor automatically removes multiple slashes and leading or trailing spaces.

10.3.2 Formatação avançada

In addition to metadata substitution, templates can conditionally include additional text and control how substituted data is formatted.

Conditionally including text

Sometimes you want text to appear in the output only if a field is not empty. A common case is `series` and `series_index` where you want either nothing or the two values separated by a hyphen. calibre handles this case using a special `template expression` syntax.

For example and using the above Foundation example, assume you want the template to produce *Foundation - 3 - Second Foundation*. This template produces that output:

```
{series} - {series_index} - {title}
```

However, if a book has no series the template will produce - - *the title*, which is probably not what you want. Generally, people want the result be the title without the extraneous hyphens. You can accomplish this using the following template syntax:

```
{field:|prefix_text|suffix_text}
```

This `template expression` says that if `field` has the value *XXXX* then the result will be *prefix_textXXXXXsuffix_text*. If `field` is empty (has no value) then the result will be the empty string (nothing) because the prefix and suffix are ignored. The prefix and suffix can contain blanks.

Do not use subtemplates (`{ … }`) or functions (see below) in the prefix or the suffix.

Using this syntax, we can solve the above no-series problem with the template:

```
{series}{series_index:| - | - }{title}
```

The hyphens will be included only if the book has a series index, which it has only if it has a series. Continuing the Foundation example again, the template will produce *Foundation - 1 - Second Foundation*.

Notes:

- You must include the colon after the `lookup name` if you are using a prefix or a suffix.
- You must either use either no or both `|` characters. Using one, as in `{field:| - }`, is not allowed.
- It is OK to provide no text for either the prefix or the suffix, such as in `{series:|| - }`. The template `{title:||}` is the same as `{title}`.

Formatting

Suppose you want the `series_index` to be formatted as three digits with leading zeros. This does the trick:

`{series_index:0>3s}` - Three digits with leading zeros

For trailing zeros, use:

`{series_index:0<3s}` - Three digits with trailing zeros

If you use series indices with fractional values, e.g., 1.1, you might want the decimal points to line up. For example, you might want the indices 1 and 2.5 to appear as 01.00 and 02.50 so that they will sort correctly on a device that does lexical sorting. To do this, use:

`{series_index:0>5.2f}` - Five characters consisting of two digits with leading zeros, a decimal point, then 2 digits after the decimal point.

If you want only the first two letters of the data, use:

`{author_sort:.2}` - Only the first two letters of the author sort name

Much of the calibre template language formatting comes from Python. For more details on the syntax of these advanced formatting operations see the Python documentation[80].

10.3.3 Using templates to define custom columns

Templates can be used to display information that isn't in calibre metadata, or to display metadata differently from calibre's normal format. For example, you might want to show the `ISBN`, a field that calibre does not display. You can accomplish this creating a custom column with the type *Column built from other columns* (hereafter called *composite columns*) and providing a template to generate the displayed text. The column will display the result of evaluating the template. For example, to display the ISBN, create the column and enter `{identifiers:select(isbn)}` in the template box. To display a column containing the values of two series custom columns, separated by a comma, use `{#series1:||, }{#series2}`.

Colunas compostas podem usar qualquer opção de modelo, incluindo formatação.

Note: You cannot edit the data displayed in a composite column. Instead you edit the source columns. If you edit a composite column, for example by double-clicking it, calibre will open the template for editing, not the underlying data.

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

10.3.4 Using functions in templates - Single Function Mode

Suppose you want to display the value of a field in upper case when that field is normally in title case. You can do this using *template functions*. For example, to display the title in upper case use the `uppercase` function, as in `{title:uppercase()}`. To display it in title case, use `{title:titlecase()}`.

Functions go into the format part of the template, after the `:` and before the first `|` or the closing `}` if no prefix/suffix is used. If you have both a format and a function reference, the function comes after a second `:`. Functions return the value of the column specified in the template, suitably modified.

The syntax for using functions is one of:

```
{lookup_name:function(arguments)}
{lookup_name:format:function(arguments)}
{lookup_name:function(arguments)|prefix|suffix}
{lookup_name:format:function(arguments)|prefix|suffix}
```

Function names must always be followed by opening and closing parentheses. Some functions require extra values (arguments), and these go inside the parentheses. Arguments are separated by commas. Literal commas (commas as text, not argument separators) must be preceded by a backslash (`\`) . The last (or only) argument cannot contain a textual closing parenthesis.

Functions are evaluated before format specifications and the prefix/suffix. See further down for an example of using both a format and a function.

Important: If you have programming experience, please note that the syntax in *Single Function Mode* is not what you expect. Strings are not quoted and spaces are significant. All arguments are considered to be constants; there are no expressions.

Do not use subtemplates (`{ ... }`) as function arguments. Instead, use *Template Program Mode* (página 175) and *General Program Mode* (página 163).

Some functions require regular expressions. In the template language regular expression matching is case-insensitive.

In the function documentation below, the notation `[something]*` means that `something` can be repeated zero or more times. The notation `[something]+` means that the `something` is repeated one or more times (must exist at least one time).

The functions intended for use in Single Function Mode are:

- `capitalize()` – returns the value with the first letter upper case and the rest lower case.
- `contains(pattern, text if match, text if not match)` – checks if the value is matched by the regular expression `pattern`. Returns `text if match` if the pattern matches the value, otherwise returns `text if no match`.
- `count(separator)` – interprets the value as a list of items separated by `separator` and returns the number of items in the list. Most lists use a comma as the separator, but `authors` uses an ampersand (&). Examples: `{tags:count(,)}`, `{authors:count(&)}`. Aliases: `count()`, `list_count()`
- `format_number(template)` – interprets the value as a number and formats that number using a Python formatting template such as `{0:5.2f}` or `{0:,d}` or `${0:5,.2f}`. The formatting template must begin with `{0:` and end with `}` as in the above examples. Exception: you can leave off the leading «{0:» and trailing «}» if the format template contains only a format. See the template language and the Python documentation[81] for more examples. Returns the empty string if formatting fails.
- `human_readable()` – expects the value to be a number and returns a string representing that number in KB, MB, GB, etc.

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

- `ifempty(text if empty)` – if the value is not empty then return the value of the field, otherwise return *text if empty*.
- `in_list(separator, [ pattern, found_val, ]* not_found_val)` – interpret the value as a list of items separated by `separator`, checking the `pattern` against each item in the list. If the `pattern` matches an item then return `found_val`, otherwise return `not_found_val`. The pair `pattern` and `found_value` can be repeated as many times as desired, permitting returning different values depending on the item's value. The patterns are checked in order, and the first match is returned.
- `language_strings(localize)` – return the language names[82] for the language codes[83] passed in as the value. Example: `{languages:language_strings()}`. If `localize` is zero, return the strings in English. If `localize` is not zero, return the strings in the language of the current locale. `Lang_codes` is a comma-separated list.
- `list_item(index, separator)` – interpret the value as a list of items separated by `separator`, returning the "index'th item. The first item is number zero. The last item has the index `-1` as in `list_item(-1, separator)`. If the item is not in the list, then the empty string is returned.
- `lookup([ pattern, key, ]* else_key)` – The patterns will be checked against the value in order. If a pattern matches then the value of the field named by `key` is returned. If no pattern matches then the value of the field named by `else_key` is returned. See``switch`` (below).
- `lowercase()` – returns the value of the field in lower case.
- `rating_to_stars(use_half_stars)` – Returns the rating as string of star (★) characters. The value must be a number between 0 and 5. Set use_half_stars to 1 if you want half star characters for fractional numbers available with custom ratings columns.
- `re(pattern, replacement)` – return the value after applying the regular expression. All instances of `pattern` in the value are replaced with `replacement`. The template language uses case insensitive Python regular expressions[84].
- `select(key)` – interpret the value as a comma-separated list of items with each item having the form `id:value` (the calibre `identifier` format). The function finds the first pair with the id equal to key and returns the corresponding value. If no id matches then the function returns the empty string.
- `shorten(left chars, middle text, right chars)` – Devolve uma versão abreviada do valor, que consiste em caracteres `left chars` do início do valor, seguido por `middle text`, seguido pelos caracteres `right chars` do fim do valor. `Left chars` e `right chars` devem ser inteiros não negativos. Por exemplo, suponha que pretende exibir títulos com uma dimensão de, no máximo, 15 caracteres. Um modelo que permite isto será `{title:shorten(9,-,5)}` ou seja mostrar os primeiros 9 caracteres do título, um separador - e os últimos 5 caracteres. No caso de um livro com o título *Antigas Leis Inglesas nos Tempos de Ivanhoe*, o resultado será *Antigas I-nhoe*. Note que se o comprimento do campo for menor que os `caracteres à esquerda` + `caracteres à direita` + o comprimento do `texto do meio`, então o campo será usado sem alterações. Por exemplo, o título *A Cúpula* não seria alterado.
- `str_in_list(separator, [ string, found_val, ]+ not_found_val)` – interpret the value as a list of items separated by `separator` then compare `string` against each value in the list. The `string` is not a regular expression. If `string` is equal to any item (ignoring case) then return the corresponding `found_val`. If `string` contains `separators` then it is also treated as a list and each subvalue is checked. The `string` and `found_value` pairs can be repeated as many times as desired, permitting returning different values depending on string's value. If none of the strings match then `not_found_value` is returned. The strings are checked in order. The first match is returned.
- `subitems(start_index, end_index)` – This function breaks apart lists of tag-like hierarchical items such as genres. It interprets the value as a comma-separated list of tag-like items, where each item is a period-

[82] https://www.loc.gov/standards/iso639-2/php/code_list.php
[83] https://www.loc.gov/standards/iso639-2/php/code_list.php
[84] https://docs.python.org/3/library/re.html

-separated list. It returns a new list made by extracting from each item the components from `start_index` to `end_index`, then merging the results back together. Duplicates are removed. The first subitem in a period--separated list has an index of zero. If an index is negative then it counts from the end of the list. As a special case, an end_index of zero is assumed to be the length of the list.

Exemplos:

- 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)}` devolve «A, D»
  - `{#genre:subitems(0,2)}` devolve «A.B, D.E»

- `sublist(start_index, end_index, separator)` – interpret the value as a list of items separated by `separator`, returning a new list made from the items from `start_index` to `end_index`. The first item is number zero. If an index is negative, then it counts from the end of the list. As a special case, an end_index of zero is assumed to be the length of the list.

  Examples assuming that the tags column (which is comma-separated) contains «A, B ,C»:

  - `{tags:sublist(0,1,\,)}` returns «A»
  - `{tags:sublist(-1,0,\,)}` returns «C»
  - `{tags:sublist(0,-1,\,)}` returns «A, B»
- `swap_around_articles(separator)` – returns the value with articles moved to the end. The value can be a list, in which case each item in the list is processed. If the value is a list then you must provide the `separator`. If no `separator` is provided then the value is treated as being a single value, not a list. The *articles* are those used by calibre to generate the `title_sort`.
- `swap_around_comma()` – given a value of the form `B, A`, return `A B`. This is most useful for converting names in LN, FN format to FN LN. If there is no comma in the value then the function returns the value unchanged.
- `switch([pattern, value,]+ else_value)` – for each `pattern, value` pair, checks if the value matches the regular expression `pattern` and if so returns the associated `value`. If no `pattern` matches, then `else_value` is returned. You can have as many `pattern, value` pairs as you wish. The first match is returned.
- `test(text if not empty, text if empty)` – return `text if not empty` if the value is not empty, otherwise return `text if empty`.
- `titlecase()` – returns the value of the field in title case.
- `transliterate()` – Return a string in a latin alphabet formed by approximating the sound of the words in the source field. For example, if the source field is `Фёдор Михáйлович Достоевский` this function returns `Fiodor Mikhailovich Dostoievskii`.
- `uppercase()` – returns the value of the field in upper case.

Using functions and formatting in the same template

Suppose you have an integer custom column `#myint` that you want displayed with leading zeros, as in `003`. One way to do this is to use a format of `0>3s`. However, by default if a number (integer or float) equals zero then the value is displayed as the empty string so zero values will produce the empty string, not `000`. If you want to see `000` values then you use both the format string and the `ifempty` function to change the empty value back to a zero. The template would be:

```
{#myint:0>3s:ifempty(0)}
```

Note that you can use the prefix and suffix as well. If you want the number to appear as `[003]` or `[000]`, then use the template:

```
{#myint:0>3s:ifempty(0)|[|]}
```

10.3.5 General Program Mode

General Program Mode (*GPM*) replaces *template expressions* with a program written in the *template language*. The syntax of the language is defined by the following grammar:

```
program          ::= 'program:' expression_list
expression_list  ::= top_expression [ ';' top_expression ]*
top_expression   ::= or_expression
or_expression    ::= and_expression [ '||' and_expression ]*
and_expression   ::= not_expression [ '&&' not_expression ]*
not_expression   ::= [ '!' not_expression ]* | concatenate_expr
concatenate_expr::= compare_expr [ '&' compare_expr ]*
compare_expr     ::= add_sub_expr [ compare_op add_sub_expr ]
compare_op       ::= '==' | '!=' | '>=' | '>' | '<=' | '<' | 'in' | 'inlist' |
                     '==#' | '!=#' | '>=#' | '>#' | '<=#' | '<#'
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
```

(continues on next page)

(continuação da página anterior)

```
stop_expr       ::= top_expression
step_expr       ::= top_expression
limit_expr      ::= top_expression
```

Notes:

- a `top_expression` always has a value. The value of an `expression_list` is the value of the last `top_expression` in the list. For example, the value of the expression list `1;2;'foobar';3` is `3`.
- In a logical context, any non-empty value is `True`
- In a logical context, the empty value is `False`
- Strings and numbers can be used interchangeably. For example, `10` and `'10'` are the same thing.
- Comments are lines starting with a "#" character. Comments beginning later in a line are not supported.

Precedência de operador

The operator precedence (order of evaluation) from highest (evaluated first) to lowest (evaluated last) is:

- Function calls, constants, parenthesized expressions, statement expressions, assignment expressions, field references.
- Unary plus (`+`) and minus (`-`). These operators evaluate right to left.

  These and all the other arithmetic operators return integers if the expression results in a fractional part equal to zero. For example, if an expression returns `3.0` it is changed to `3`.
- Multiply (`*`) and divide (`/`). These operators are associative and evaluate left to right. Use parentheses if you want to change the order of evaluation.
- Add (`+`) and subtract (`-`). These operators are associative and evaluate left to right.
- Numeric and string comparisons. These operators return `'1'` if the comparison succeeds, otherwise the empty string (`''`). Comparisons are not associative: `a < b < c` is a syntax error.
- String concatenation (`&`). The `&` operator returns a string formed by concatenating the left-hand and right-hand expressions. Example: `'aaa' & 'bbb'` returns `'aaabbb'`. The operator is associative and evaluates left to right.
- Unary logical not (`!`). This operator returns `'1'` if the expression is False (evaluates to the empty string), otherwise `''`.
- Logical and (`&&`). This operator returns "1" if both the left-hand and right-hand expressions are True, or the empty string `''` if either is False. It is associative, evaluates left to right, and does short-circuiting[85].
- Logical or (`||`). This operator returns `'1'` if either the left-hand or right-hand expression is True, or `''` if both are False. It is associative, evaluates left to right, and does short-circuiting[86]. It is an *inclusive or*, returning `'1'` if both the left- and right-hand expressions are True.

Referência de campos

A `field_reference` evaluates to the value of the metadata field named by lookup name that follows the `$` or `$$`. Using `$` is equivalent to using the `field()` function. Using `$$` is equivalent to using the `raw_field` function. Examples:

[85] https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html
[86] https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html

```
* $authors ==> field('authors')
* $#genre ==> field('#genre')
* $$pubdate ==> raw_field('pubdate')
* $$#my_int ==> raw_field('#my_int')
```

Expressões If

`If` expressions first evaluate the `condition`. If the `condition` is True (a non-empty value) then the `expression_list` in the `then` clause is evaluated. If it is False then if present the `expression_list` in the `elif` or `else` clause is evaluated. The `elif` and `else` parts are optional. The words `if`, `then`, `elif`, `else`, and `fi` are reserved; you cannot use them as identifier names. You can put newlines and white space wherever they make sense. The `condition` is a `top_expression` not an `expression_list`; semicolons are not allowed. The `expression_lists` are semicolon-separated sequences of `top_expressions`. An `if` expression returns the result of the last `top_expression` in the evaluated `expression_list`, or the empty string if no expression list was evaluated.

Examples:

```
* program: if field('series') then 'yes' else 'no' fi
* program:
      if field('series') then
          a = 'yes';
          b = 'no'
      else
          a = 'no';
          b = 'yes'
      fi;
      strcat(a, '-', b)
```

Exemplo `if` aninhado:

```
program:
  if field('series') then
    if check_yes_no(field('#mybool'), '', '', '1') then
      'yes'
    else
      'no'
    fi
  else
    'no series'
  fi
```

As said above, an `if` produces a value. This means that all the following are equivalent:

```
* program: if field('series') then 'foo' else 'bar' fi
* program: if field('series') then a = 'foo' else a = 'bar' fi; a
* program: a = if field('series') then 'foo' else 'bar' fi; a
```

As a last example, this program returns the value of the `series` column if the book has a series, otherwise the value of the `title` column:

```
program: field(if field('series') then 'series' else 'title' fi)
```

Expressões For

The `for` expression iterates over a list of values, processing them one at a time. The `list_expression` must evaluate either to a metadata field `lookup name` e.g., `tags` or `#genre`, or to a list of values. The *range() function* (página 173) (see below) generates a list of numbers. If the result is a valid `lookup name` then the field's value is fetched and the separator specified for that field type is used. If the result isn't a valid lookup name then it is assumed to be a list of values.

The list is assumed to be separated by commas unless the optional keyword `separator` is supplied, in which case the list values must be separated by the result of evaluating the `separator_expr`. A separator cannot be used if the list is generated by `range()`. Each value in the list is assigned to the specified variable then the `expression_list` is evaluated. You can use `break` to jump out of the loop, and `continue` to jump to the beginning of the loop for the next iteration.

Example: This template removes the first hierarchical name for each value in Genre (`#genre`), constructing a list with the new names:

```
program:
  new_tags = '';
  for i in '#genre':
    j = re(i, '^.*?\.(.*)$', '\1');
    new_tags = list_union(new_tags, j, ',')
  rof;
  new_tags
```

If the original Genre is *History.Military, Science Fiction.Alternate History, ReadMe* then the template returns *Military, Alternate History, ReadMe*. You could use this template in calibre's *Edit metadata in bulk* → *Search & replace* with *Search for* set to `template` to strip off the first level of the hierarchy and assign the resulting value to Genre.

Note: the last line in the template, `new_tags`, isn't strictly necessary in this case because `for` returns the value of the last top_expression in the expression list. The value of an assignment is the value of its expression, so the value of the `for` statement is what was assigned to `new_tags`.

Function definition

If you have code in a template that repeats then you can put that code into a local function. The `def` keyword starts the definition. It is followed by the function name, the argument list, then the code in the function. The function definition ends with the `fed` keyword.

Arguments are positional. When a function is called the supplied arguments are matched left to right against the defined parameters, with the value of the argument assigned to the parameter. It is an error to provide more arguments than defined parameters. Parameters can have default values, such as `a = 25`. If an argument is not supplied for that parameter then the default value is used, otherwise the parameter is set to the empty string.

The `return` statement can be used in a local function.

A function must be defined before it can be used.

Example: This template computes an approximate duration in years, months, and days from a number of days. The function `to_plural()` formats the computed values. Note that the example also uses the `&` operator:

```
program:
      days = 2112;
      years = floor(days/360);
      months = floor(mod(days, 360)/30);
      days = days - ((years*360) + (months * 30));

      def to_plural(v, str):
              if v == 0 then return '' fi;
              return v & ' ' & (if v == 1 then str else str & 's' fi) & ' '
      fed;

      to_plural(years, 'year') & to_plural(months, 'month') & to_plural(days,'day')
```

Operadores relacionais

Relational operators return `'1'` if the comparison is true, otherwise the empty string ("").

There are two forms of relational operators: string comparisons and numeric comparisons.

String comparisons do case-insensitive string comparison using lexical order. The supported string comparison operators are ==, !=, <, <=, >, >=, in, and inlist. For the in operator, the result of the left hand expression is interpreted as a regular expression pattern. The in operator is True if the value of left-hand regular expression matches the value of the right hand expression. The inlist operator is true if the left hand regular expression matches any one of the items in the right hand list where the items in the list are separated by commas. The matches are case-insensitive.

The numeric comparison operators are ==#, !=#, <#, <=#, >#, >=#. The left and right expressions must evaluate to numeric values with two exceptions: both the string value «None» (undefined field) and the empty string evaluate to the value zero.

Exemplos:

- program: field('series') == 'foo' returns '1' if the book's series is "foo", otherwise ''.
- program: 'f.o' in field('series') returns '1' if the book's series matches the regular expression f.o (e.g., *foo*, *Off Onyx*, etc.), otherwise ''.
- program: 'science' inlist field('#genre') returns '1' if any of the book's genres match the regular expression science, e.g., *Science*, *History of Science*, *Science Fiction* etc.), otherwise ''.
- program: '^science$' inlist field('#genre') returns '1' if any of the book's genres exactly match the regular expression ^science$, e.g., *Science*. The genres *History of Science* and *Science Fiction* don't match. If there isn't a match then returns ''.
- program: if field('series') != 'foo' then 'bar' else 'mumble' fi returns 'bar' if the book's series is not foo. Otherwise it returns 'mumble'.
- program: if field('series') == 'foo' || field('series') == '1632' then 'yes' else 'no' fi returns 'yes' if series is either 'foo' or '1632', otherwise 'no'.
- program: if '^(foo|1632)$' in field('series') then 'yes' else 'no' fi returns 'yes' if series is either 'foo' or '1632', otherwise 'no'.
- program: if 11 > 2 then 'yes' else 'no' fi returns 'no' because the > operator does a lexical comparison.
- program: if 11 ># 2 then 'yes' else 'no' fi returns 'yes' because the ># operator does a numeric comparison.

Funções adicionais disponíveis

The following functions are available in addition to those described in *Single Function Mode* (página 160).

In *GPM* the functions described in *Single Function Mode* all require an additional first parameter specifying the value to operate upon. All parameters are expression_lists (see the grammar above).

- add(x [, y]*) – returns the sum of its arguments. Throws an exception if an argument is not a number. In most cases you can use the + operator instead of this function.
- and(value [, value]*) – returns the string «1» if all values are not empty, otherwise returns the empty string. You can have as many values as you want. In most cases you can use the && operator instead of this function. One reason not to replace and with && is if short-circuiting can change the results because of side effects. For example, and(a='',b=5) will always do both assignments, where the && operator won't do the second.
- assign(id, val) – assigns val to id, then returns val. id must be an identifier, not an expression. In most cases you can use the = operator instead of this function.
- approximate_formats() – return a comma-separated list of formats associated with the book. There is no guarantee that the list is correct, although it probably is. This and other zero-parameter functions can be called in Template Program Mode (see below) using the template {:'approximate_formats()'}. Note that resulting format names are always uppercase, as in EPUB. The approximate_formats() function is significantly faster than the formats_... functions discussed below.

- `author_links(val_separator, pair_separator)` – returns a string containing a list of authors and those authors" link values in the form:

```
author1 val_separator author1_link pair_separator author2 val_separator author2_
↪link etc.
```

  An author is separated from its link value by the `val_separator` string with no added spaces. `author:linkvalue` pairs are separated by the `pair_separator` string argument with no added spaces. It is up to you to choose separator strings that do not occur in author names or links. An author is included even if the author link is empty.

- `author_sorts(val_separator)` – returns a string containing a list of author's sort values for the authors of the book. The sort is the one in the author metadata information (different from the author_sort in books). The returned list has the form `author sort 1 val_separator author sort 2` etc. with no added spaces. The author sort values in this list are in the same order as the authors of the book. If you want spaces around `val_separator` then include them in the `val_separator` string.

- `book_count(query, use_vl)` – returns the count of books found by searching for `query`. If `use_vl` is `0` (zero) then virtual libraries are ignored. This function and its companion `book_values()` are particularly useful in template searches, supporting searches that combine information from many books such as looking for series with only one book. It cannot be used in composite columns unless the tweak `allow_template_database_functions_in_composites` is set to True. It can be used only in the GUI.

  For example this template search uses this function and its companion to find all series with only one book:

  1) Define a stored template (using *Preferences → Advanced → Template functions*) named `series_only_one_book` (the name is arbitrary). The template is:

```
program:
    vals = globals(vals='');
    if !vals then
        all_series = book_values('series', 'series:true', ',', 0);
        for series in all_series:
            if book_count('series:="' & series & '"', 0) == 1 then
                vals = list_join(',', vals, ',', series, ',')
            fi
        rof;
        set_globals(vals)
    fi;
    str_in_list(vals, ',', $series, 1, '')
```

     The first time the template runs (the first book checked) it stores the results of the database lookups in a `global` template variable named `vals`. These results are used to check subsequent books without redoing the lookups.

  2) Use the stored template in a template search:

```
template:"program: series_only_one_book()#@#:n:1"
```

  Using a stored template instead of putting the template into the search eliminates problems caused by the requirement to escape quotes in search expressions.

- `book_values(column, query, sep, use_vl)` – returns a list of the unique values contained in the column `column` (a lookup name), separated by `sep`, in the books found by searching for `query`. If `use_vl` is `0` (zero) then virtual libraries are ignored. This function and its companion `book_count()` are particularly useful in template searches, supporting searches that combine information from many books such as looking for series with only one book. It cannot be used in composite columns unless the tweak `allow_template_database_functions_in_composites` is set to True. It can be used only in the GUI.

- `booksize()` – returns the value of the calibre "size" field. Returns "" if there are no formats.
- `check_yes_no(field_name, is_undefined, is_false, is_true)` – checks if the value of the yes/no field named by the lookup name `field_name` is one of the values specified by the parameters, returning `'yes'` if a match is found otherwise returning the empty string. Set the parameter `is_undefined`, `is_false`, or `is_true` to 1 (the number) to check that condition, otherwise set it to 0. Example:

  `check_yes_no("#bool", 1, 0, 1)` returns `'yes'` if the yes/no field `#bool` is either True or undefined (neither True nor False).

  More than one of `is_undefined`, `is_false`, or `is_true` can be set to 1.
- `ceiling(x)` – returns the smallest integer greater than or equal to `x`. Throws an exception if `x` is not a number.
- `character(character_name)` – returns the character named by character_name. For example, `character('newline')` returns a newline character (`'\n'`). The supported character names are `newline`, `return`, `tab`, and `backslash`.
- `cmp(x, y, lt, eq, gt)` – compares `x` and `y` after converting both to numbers. Returns `lt` if `x <# y`, `eq` if `x ==# y`, otherwise `gt`. This function can usually be replaced with one of the numeric compare operators (==#, <#, >#, etc).
- `connected_device_name(storage_location_key)` – if a device is connected then return the device name, otherwise return the empty string. Each storage location on a device has its own device name. The `storage_location_key` names are `'main'`, `'carda'` and `'cardb'`. This function works only in the GUI.
- `connected_device_uuid(storage_location_key)` – if a device is connected then return the device uuid (unique id), otherwise return the empty string. Each storage location on a device has a different uuid. The `storage_location_key` location names are `'main'`, `'carda'` and `'cardb'`. This function works only in the GUI.
- `current_library_name()` – return the last name on the path to the current calibre library.
- `current_library_path()` – return the full path to the current calibre library.
- `current_virtual_library_name()` – return the name of the current virtual library if there is one, otherwise the empty string. Library name case is preserved. Example: `program: current_virtual_library_name()`. This function works only in the GUI.
- `date_arithmetic(date, calc_spec, fmt)` – Calculate a new date from `date` using `calc_spec`. Return the new date formatted according to optional `fmt`: if not supplied then the result will be in ISO format. The calc_spec is a string formed by concatenating pairs of `vW` (`valueWhat`) where `v` is a possibly-negative number and W is one of the following letters:
  - `s`: add `v` seconds to `date`
  - `m`: add `v` minutes to `date`
  - `h`: add `v` hours to `date`
  - `d`: add `v` days to `date`
  - `w`: add `v` weeks to `date`
  - `y`: add `v` years to `date`, where a year is 365 days.

  Example: `'1s3d-1m'` will add 1 second, add 3 days, and subtract 1 minute from `date`.
- `days_between(date1, date2)` – return the number of days between `date1` and `date2`. The number is positive if `date1` is greater than `date2`, otherwise negative. If either `date1` or `date2` are not dates, the function returns the empty string.

- `divide(x, y)` – returns `x / y`. Throws an exception if either `x` or `y` are not numbers. This function can usually be replaced by the `/` operator.
- `eval(string)` – evaluates the string as a program, passing the local variables. This permits using the template processor to construct complex results from local variables. In *Template Program Mode* (página 175), because the *{* and *}* characters are interpreted before the template is evaluated you must use *[[* for the *{* character and *]]* for the `}` character. They are converted automatically. Note also that prefixes and suffixes (the *|prefix|suffix* syntax) cannot be used in the argument to this function when using *Template Program Mode* (página 175).
- `field(lookup_name)` – returns the value of the metadata field with lookup name `lookup_name`.
- `field_exists(field_name)` – checks if a field (column) with the lookup name `field_name` exists, returning `'1'` if so and the empty string if not.
- `finish_formatting(val, fmt, prefix, suffix)` – apply the format, prefix, and suffix to a value in the same way as done in a template like `{series_index:05.2f| - |- }`. This function is provided to ease conversion of complex single-function- or template-program-mode templates to *GPM* Templates. For example, the following program produces the same output as the above template:

```
program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")
```

  Another example: for the template `{series:re(([^\s])[^\s]+(\s|$),\1)}{series_index:0>2s| - | - }{title}` use:

```
program:
  strcat(
    re(field('series'), '([^\s])[^\s]+(\s|$)', '\1'),
       finish_formatting(field('series_index'), '0>2s', ' - ', ' - '),
       field('title')
  )
```

- `first_matching_cmp(val, [ cmp, result, ]* else_result)` – compares `val < cmp` in sequence, returning the associated result for the first comparison that succeeds. Returns else_result if no comparison succeeds. Example:

```
i = 10;
first_matching_cmp(i,5,"small",10,"middle",15,"large","giant")
```

  returns `"large"`. The same example with a first value of 16 returns `"giant"`.
- `first_non_empty(value [, value]*)` – returns the first `value` that is not empty. If all values are empty, then the empty string is returned. You can have as many values as you want.
- `floor(x)` – returns the largest integer less than or equal to `x`. Throws an exception if `x` is not a number.
- `format_date(val, format_string)` – format the value, which must be a date string, using the format_string, returning a string. The formatting codes are:
  - `d` : the day as number without a leading zero (1 to 31)
  - `dd` : the day as number with a leading zero (01 to 31)
  - `ddd` : the abbreviated localized day name (e.g. «Mon» to «Sun»).
  - `dddd` : the long localized day name (e.g. «Monday» to «Sunday»).
  - `M` : the month as number without a leading zero (1 to 12).
  - `MM` : the month as number with a leading zero (01 to 12)
  - `MMM` : the abbreviated localized month name (e.g. «Jan» to «Dec»).
  - `MMMM` : the long localized month name (e.g. «January» to «December»).

- `yy` : the year as two digit number (00 to 99).
- `yyyy` : the year as four digit number.
- `h` : the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
- `hh` : the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
- `m` : the minutes without a leading 0 (0 to 59)
- `mm` : the minutes with a leading 0 (00 to 59)
- `s` : the seconds without a leading 0 (0 to 59)
- `ss` : the seconds with a leading 0 (00 to 59)
- `ap` : use a 12-hour clock instead of a 24-hour clock, with "ap" replaced by the localized string for am or pm.
- `AP` : use a 12-hour clock instead of a 24-hour clock, with "AP" replaced by the localized string for AM or PM.
- `iso` : the date with time and timezone. Must be the only format present.
- `to_number` : convert the date & time into a floating point number (a *timestamp*)
- `from_number` : convert a floating point number (a *timestamp*) into an `iso` formatted date. If you want a different date format then add the desired formatting string after `from_number` and a colon (`:`). Example: `from_number:MMM dd yyyy`

You might get unexpected results if the date you are formatting contains localized month names, which can happen if you changed the date format tweaks to contain `MMMM`. In this case, instead of using the `field()` function as in:

```
format_date(field('pubdate'), 'yyyy')
```

use the `raw_field()` function as in:

```
format_date(raw_field('pubdate'), 'yyyy')
```

- `formats_modtimes(date_format_string)` – return a comma-separated list of colon-separated items `FMT:DATE` representing modification times for the formats of a book. The `date_format_string` parameter specifies how the date is to be formatted. See the `format_date()` function for details. You can use the `select` function to get the modification time for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_paths()` – return a comma-separated list of colon-separated items `FMT:PATH` giving the full path to the formats of a book. You can use the select function to get the path for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_sizes()` – return a comma-separated list of colon-separated `FMT:SIZE` items giving the sizes in bytes of the formats of a book. You can use the select function to get the size for a specific format. Note that format names are always uppercase, as in EPUB.
- `fractional_part(x)` – returns the value after the decimal point. For example, `fractional_part(3.14)` returns `0.14`. Throws an exception if `x` is not a number.
- `has_cover()` – return `'Yes'` if the book has a cover, otherwise the empty string.
- `is_marked()` – check whether the book is *marked* in calibre. If it is then return the value of the mark, either `'true'` (lower case) or a comma-separated list of named marks. Returns `''` (the empty string) if the book is not marked. This function works only in the GUI.
- `language_codes(lang_strings)` – return the language codes[87] for the language names passed in

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

lang_strings. The strings must be in the language of the current locale. `Lang_strings` is a comma-separated list.

- `list_contains(value, separator, [ pattern, found_val, ]* not_found_val)` – (Alias of `in_list`) Interpreting the value as a list of items separated by `separator`, evaluate the `pattern` against each value in the list. If the `pattern` matches any value then return `found_val`, otherwise return `not_found_val`. The `pattern` and `found_value` can be repeated as many times as desired, permitting returning different values depending on the search. The patterns are checked in order. The first match is returned. Aliases: `in_list()`, `list_contains()`
- `list_count(value, separator)` – interprets `value` as a list of items separated by `separator`, returning the count of items in the list. Aliases: `count()`, `list_count()`
- `list_count_matching(list, pattern, separator)` – interprets `list` as a list of items separated by `separator`, returning the number of items in the list that match the regular expression `pattern`. Aliases: `list_count_matching()`, `count_matching()`
- `list_difference(list1, list2, separator)` – return a list made by removing from `list1` any item found in `list2` using a case-insensitive comparison. The items in `list1` and `list2` are separated by separator, as are the items in the returned list.
- `list_equals(list1, sep1, list2, sep2, yes_val, no_val)` – return `yes_val` if `list1` and *list2* contain the same items, otherwise return `no_val`. The items are determined by splitting each list using the appropriate separator character (`sep1` or `sep2`). The order of items in the lists is not relevant. The comparison is case-insensitive.
- `list_intersection(list1, list2, separator)` – return a list made by removing from `list1` any item not found in `list2`, using a case-insensitive comparison. The items in `list1` and `list2` are separated by separator, as are the items in the returned list.
- `list_join(with_separator, list1, separator1 [, list2, separator2]*)` – return a list made by joining the items in the source lists (`list1` etc) using `with_separator` between the items in the result list. Items in each source `list[123...]` are separated by the associated `separator[123...]`. A list can contain zero values. It can be a field like `publisher` that is single-valued, effectively a one-item list. Duplicates are removed using a case-insensitive comparison. Items are returned in the order they appear in the source lists. If items on lists differ only in letter case then the last is used. All separators can be more than one character.

  Exemplo:

```
program:
  list_join('#@#', $authors, '&', $tags, ',')
```

  You can use `list_join` on the results of previous calls to `list_join` as follows:

```
program:
  a = list_join('#@#', $authors, '&', $tags, ',');
  b = list_join('#@#', a, '#@#', $#genre, ',', $#people, '&', 'some value', ',')
```

  You can use expressions to generate a list. For example, assume you want items for `authors` and `#genre`, but with the genre changed to the word «Genre: « followed by the first letter of the genre, i.e. the genre «Fiction» becomes «Genre: F». The following will do that:

```
program:
  list_join('#@#', $authors, '&', list_re($#genre, ',', '^(.).*$', 'Genre: \1'),
↪',')
```

- `list_re(src_list, separator, include_re, opt_replace)` – Construct a list by first separating `src_list` into items using the `separator` character. For each item in the list, check if it matches

include_re. If it does then add it to the list to be returned. If opt_replace is not the empty string then apply the replacement before adding the item to the returned list.

- list_re_group(src_list, separator, include_re, search_re [, template_for_group]*) – Like list_re except replacements are not optional. It uses re_group(item, search_re, template ...) when doing the replacements.
- list_remove_duplicates(list, separator) – return a list made by removing duplicate items in list. If items differ only in case then the last is returned. The items in list are separated by separator, as are the items in the returned list.
- list_sort(list, direction, separator) – return list sorted using a case-insensitive lexical sort. If direction is zero, list is sorted ascending, otherwise descending. The list items are separated by separator, as are the items in the returned list.
- list_split(list_val, sep, id_prefix) – splits list_val into separate values using sep, then assigns the values to local variables named id_prefix_N where N is the position of the value in the list. The first item has position 0 (zero). The function returns the last element in the list.

  Exemplo:

```
list_split('one:two:foo', ':', 'var')
```

  is equivalent to:

```
var_0 = 'one';
var_1 = 'two';
var_2 = 'foo
```

- list_union(list1, list2, separator) – return a list made by merging the items in list1 and list2, removing duplicate items using a case-insensitive comparison. If items differ in case, the one in list1 is used. The items in list1 and list2 are separated by separator, as are the items in the returned list. Aliases: merge_lists(), list_union()
- mod(x, y) – returns the floor of the remainder of x / y. Throws an exception if either x or y is not a number.
- multiply(x [, y]*) – returns the product of its arguments. Throws an exception if any argument is not a number. This function can usually be replaced by the * operator.
- not(value) – returns the string «1» if the value is empty, otherwise returns the empty string. This function can usually be replaced with the unary not (!) operator.
- ondevice() – return the string 'Yes' if ondevice is set, otherwise return the empty string.
- or(value [, value]*) – returns the string '1' if any value is not empty, otherwise returns the empty string. You can have as many values as you want. This function can usually be replaced by the || operator. A reason it cannot be replaced is if short-circuiting will change the results because of side effects.
- print(a [, b]*) – prints the arguments to standard output. Unless you start calibre from the command line (calibre-debug -g), the output will go into a black hole. The print function always returns its first argument.
- range(start, stop, step, limit) – returns a list of numbers generated by looping over the range specified by the parameters start, stop, and step, with a maximum length of limit. The first value produced is "start". Subsequent values next_v = current_v + step. The loop continues while next_v < stop assuming step is positive, otherwise while next_v > stop. An empty list is produced if start fails the test: start >= stop if step is positive. The limit sets the maximum length of the list and has a default of 1000. The parameters start, step, and limit are optional. Calling range() with one argument specifies stop. Two arguments specify start and stop. Three arguments specify start, stop, and step. Four arguments specify start, stop, step and limit. Examples:

```
range(5) -> '0, 1, 2, 3, 4'
range(0, 5) -> '0, 1, 2, 3, 4'
range(-1, 5) -> '-1, 0, 1, 2, 3, 4'
range(1, 5) -> '1, 2, 3, 4'
range(1, 5, 2) -> '1, 3'
range(1, 5, 2, 5) -> '1, 3'
range(1, 5, 2, 1) -> error(limit exceeded)
```

- `raw_field(lookup_name [, optional_default])` – returns the metadata field named by `lookup_name` without applying any formatting. It evaluates and returns the optional second argument `optional_default` if the field's value is undefined (`None`).
- `raw_list(lookup_name, separator)` – returns the metadata list named by `lookup_name` without applying any formatting or sorting, with the items separated by separator.
- `re_group(value, pattern [, template_for_group]*)` – return a string made by applying the regular expression pattern to `value` and replacing each matched instance with the the value returned by the corresponding template. In *Template Program Mode* (página 175), like for the `template` and the `eval` functions, you use `[[` for `{` and `]]` for `}`.

  The following example looks for a series with more than one word and uppercases the first word:

```
program: re_group(field('series'), "(\S* )(.*)", "{$:uppercase()}", "{$}")'}
```

- `round(x)` – returns the nearest integer to `x`. Throws an exception if `x` is not a number.
- `series_sort()` – devolve o valor de ordenação da série.
- `strcat(a [, b]*)` – can take any number of arguments. Returns a string formed by concatenating all the arguments.
- `strcat_max(max, string1 [, prefix2, string2]*)` – Returns a string formed by concatenating the arguments. The returned value is initialized to `string1`. Strings made from `prefix, string` pairs are added to the end of the value as long as the resulting string length is less than `max`. Prefixes can be empty. Returns `string1` even if `string1` is longer than `max`. You can pass as many `prefix, string` pairs as you wish.
- `strcmp(x, y, lt, eq, gt)` – does a case-insensitive lexical comparison of `x` and `y`. Returns `lt` if `x < y`, `eq` if `x == y`, otherwise `gt`. This function can often be replaced by one of the lexical comparison operators (==, >, <, etc.)
- `strcmpcase(x, y, lt, eq, gt)` – does a case-sensitive lexical comparison of `x` and `y`. Returns `lt` if `x < y`, `eq` if `x == y`, otherwise `gt`.

  Note: This is NOT the default behavior used by calibre, for example, in the lexical comparison operators (==, >, <, etc.). This function could cause unexpected results, preferably use `strcmp()` whenever possible.
- `strlen(value)` – Returns the length of the string `value`.
- `substr(str, start, end)` – returns the `start`'th through the `end`'th characters of `str`. The first character in `str` is the zero'th character. If `end` is negative, then it indicates that many characters counting from the right. If `end` is zero, then it indicates the last character. For example, `substr('12345', 1, 0)` returns `'2345'`, and `substr('12345', 1, -1)` returns `'234'`.
- `subtract(x, y)` – returns `x - y`. Throws an exception if either `x` or `y` are not numbers. This function can usually be replaced by the − operator.
- `switch_if([test_expression, value_expression,]+ else_expression)` – for each `test_expression, value_expression` pair, checks if `test_expression` is True (non-empty) and if so returns the result of `value_expression`. If no `test_expression` is True then the result of `else_expression` is returned. You can have as many ``test_expression, value_expression`` pairs as you want.

- `today()` – return a date+time string for today (now). This value is designed for use in *format_date* or *days_between*, but can be manipulated like any other string. The date is in ISO[88] date/time format.
- `template(x)` – evaluates `x` as a template. The evaluation is done in its own context, meaning that variables are not shared between the caller and the template evaluation.
- `to_hex(val)` – returns the string `val` encoded in hex. This is useful when constructing calibre URLs.
- `urls_from_identifiers(identifiers, sort_results)` – given a comma-separated list of `identifiers`, where an *identifier* is a colon-separated pair of values (`id_name:id_value`), returns a comma-separated list of HTML URLs generated from the identifiers. The list not sorted if sort_results is `0` (character or number), otherwise it is sorted alphabetically by the identifier name. The URLs are generated in the same way as the built-in identifiers column when shown in *Book details*.

10.3.6 More complex programs in template expressions - Template Program Mode

Template Program Mode (*TPM*) is a blend of *General Program Mode* (página 163) and *Single Function Mode* (página 160). *TPM* differs from Single Function Mode in that it permits writing template expressions that refer to other metadata fields, use nested functions, modify variables, and do arithmetic. It differs from *General Program Mode* in that the template is contained between `{` and `}` characters and doesn't begin with the word `program:`. The program portion of the template is a General Program Mode expression list.

Example: assume you want a template to show the series for a book if it has one, otherwise show the value of a custom field #genre. You cannot do this in the *Single Function Mode* (página 160) because you cannot make reference to another metadata field within a template expression. In *TPM* you can, as the following expression demonstrates:

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

O exemplo mostra várias coisas:

- *TPM* is used if the expression begins with `:'` and ends with `'}`. Anything else is assumed to be in *Single Function Mode* (página 160).
- the variable `$` stands for the field named in the template: the expression is operating upon, `#series` in this case.
- functions must be given all their arguments. There is no default value. For example, the standard built-in functions must be given an additional initial parameter indicating the source field.
- white space is ignored and can be used anywhere within the expression.
- constant strings are enclosed in matching quotes, either `'` or `"`.

All the functions listed under *Single Function Mode* and *General Program Mode* can be used in *TPM*.

In *TPM*, using `{` and `}` characters in string literals can lead to errors or unexpected results because they confuse the template processor. It tries to treat them as template expression boundaries, not characters. In some but not all cases you can replace a `{` with `[[` and a `}` with *]]*. Generally, if your program contains `{` and `}` characters then you should use *General Program Mode*.

As with *General Program Mode*, for functions documented under *Single Function Mode* (página 160) you must supply the value the function is to act upon as the first parameter in addition to the documented parameters. In *TPM* you can use `$` to access the value specified by the `lookup name` for the template expression.

[88] https://en.wikipedia.org/wiki/ISO_8601

10.3.7 Python Template Mode

Python Template Mode (PTM) lets you write templates using native python and the calibre API[89]. The database API will be of most use; further discussion is beyond the scope of this manual. PTM templates are faster and can do more complicated operations but you must know how to write code in python using the calibre API.

A PTM template begins with:

```
python:
def evaluate(book, context):
    # book is a calibre metadata object
    # context is an instance of calibre.utils.formatter.PythonTemplateContext,
    # which currently contains the following attributes:
    # db: a calibre legacy database object.
    # globals: the template global variable dictionary.
    # arguments: is a list of arguments if the template is called by a GPM template,␣
↪otherwise None.
    # funcs: used to call Built-in/User functions and Stored GPM/Python templates.
    # Example: context.funcs.list_re_group()

    # your Python code goes here
    return 'a string'
```

You can add the above text to your template using the context menu, usually accessed with a right click. The comments are not significant and can be removed. You must use python indenting.

The context object supports `str(context)` that returns a string of the context's contents, and `context.attributes` that returns a list of the attribute names in the context.

The `context.funcs` attribute allows calling Built-in and User template functions, and Stored GPM/Python templates, so that you can execute them directly in your code. The functions are retrieved using their names. If the name conflicts with a Python keyword, add an underscore to the end of the name. Examples:

```
context.funcs.list_re_group()
context.funcs.assert_()
```

Here is an example of a PTM template that produces a list of all the authors for a series. The list is stored in a *Column built from other columns, behaves like tags*. It shows in *Book details* and has the *on separate lines* checked (in *Preferences → Look & feel → Book details*). That option requires the list to be comma-separated. To satisfy that requirement the template converts commas in author names to semicolons then builds a comma-separated list of authors. The authors are then sorted, which is why the template uses author_sort.

```
python:
def evaluate(book, context):
    if book.series is None:
        return ''
    db = context.db.new_api
    ans = set()
    # Get the list of books in the series
    ids = db.search(f'series:"={book.series}"', '')
    if ids:
        # Get all the author_sort values for the books in the series
        author_sorts = (v for v in db.all_field_for('author_sort', ids).values())
        # Add the names to the result set, removing duplicates
        for aus in author_sorts:
            ans.update(v.strip() for v in aus.split('&'))
```

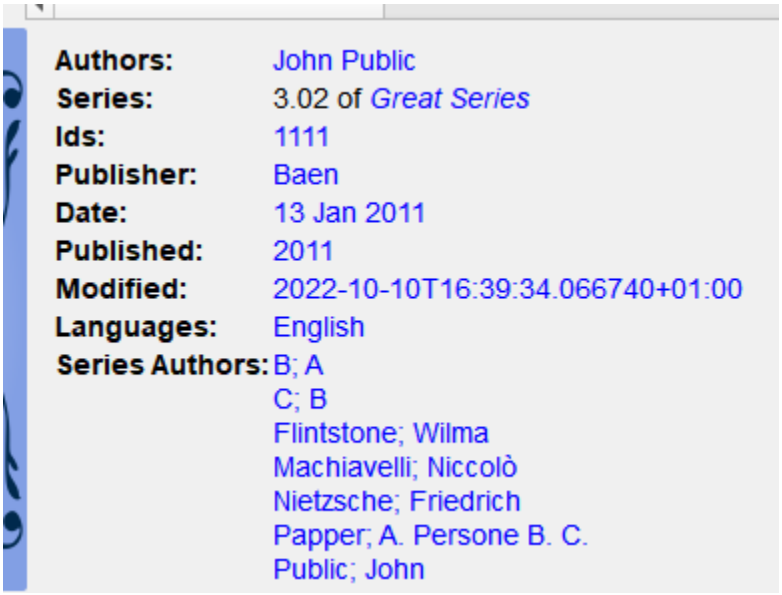
(continues on next page)

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

(continuação da página anterior)

```
    # Make a sorted comma-separated string from the result set
    return ', '.join(v.replace(',', ';') for v in sorted(ans))
```

The output in *Book details* looks like this:



10.3.8 Modelos guardados

Both *General Program Mode* (página 163) and *Python Template Mode* (página 176) support saving templates and calling those templates from another template, much like calling stored functions. You save templates using *Preferences → Advanced → Template functions*. More information is provided in that dialog. You call a template the same way you call a function, passing positional arguments if desired. An argument can be any expression. Examples of calling a template, assuming the stored template is named `foo`:

- `foo()` – call the template passing no arguments.
- `foo(a, b)` call the template passing the values of the two variables `a` and `b`.
- `foo(if field('series') then field('series_index') else 0 fi)` – if the book has a `series` then pass the `series_index`, otherwise pass the value `0`.

In GPM you retrieve the arguments passed in the call to the stored template using the `arguments` function. It both declares and initializes local variables, effectively parameters. The variables are positional; they get the value of the parameter given in the call in the same position. If the corresponding parameter is not provided in the call then `arguments` assigns that variable the provided default value. If there is no default value then the variable is set to the empty string. For example, the following `arguments` function declares 2 variables, `key`, `alternate`:

```
arguments(key, alternate='series')
```

Examples, again assuming the stored template is named `foo`:

- `foo('#myseries')` – argument `key` is assigned the value `'myseries'` and the argument `alternate` is assigned the default value `'series'`.
- `foo('series', '#genre')` the variable `key` is assigned the value `'series'` and the variable `alternate` is assigned the value `'#genre'`.

- `foo()` – the variable `key` is assigned the empty string and the variable `alternate` is assigned the value `'series'`.

In PTM the arguments are passed in the `arguments` parameter, which is a list of strings. There isn't any way to specify default values. You must check the length of the `arguments` list to be sure that the number of arguments is what you expect.

An easy way to test stored templates is using the `Template tester` dialog. For ease of access give it a keyboard shortcut in *Preferences → Advanced → Keyboard shortcuts → Template tester*. Giving the `Stored templates` dialog a shortcut will help switching more rapidly between the tester and editing the stored template's source code.

10.3.9 Providing additional information to templates

A developer can choose to pass additional information to the template processor, such as application-specific book metadata or information about what the processor is being asked to do. A template can access this information and use it during the evaluation.

Developer: how to pass additional information

The additional information is a Python dictionary containing pairs `variable_name: variable_value` where the values must be strings. The template can access the dictionary, creating template local variables named `variable_name` containing the value `variable_value`. The user cannot change the name so it is best to use names that won't collide with other template local variables, for example by prefixing the name with an underscore.

This dictionary is passed to the template processor (the `formatter`) using the named parameter `global_vars=your_dict`. The full method signature is:

```
def safe_format(self, fmt, kwargs, error_value, book,
                column_name=None, template_cache=None,
                strip_results=True, template_functions=None,
                global_vars={})
```

Template writer: how to access the additional information

You access the additional information (the `globals` dictionary) in a template using the template function:

```
globals(id[=expression] [, id[=expression]]*)
```

where `id` is any legal variable name. This function checks whether the additional information provided by the developer contains the name. If it does then the function assigns the provided value to a template local variable with that name. If the name is not in the additional information and if an `expression` is provided, the `expression` is evaluated and the result is assigned to the local variable. If neither a value nor an expression is provided, the function assigns the empty string (`''`) to the local variable.

A template can set a value in the `globals` dictionary using the template function:

```
set_globals(id[=expression] [, id[=expression]]*)
```

This function sets the `globals` dictionary key:value pair `id:value` where `value` is the value of the template local variable `id`. If that local variable doesn't exist then `value` is set to the result of evaluating `expression`.

10.3.10 Notes on the difference between modes

The three program modes, *Single Function Mode* (página 160) (SFM), *Template Program Mode* (página 175) (*TPM*), and *General Program Mode* (página 163) (*GPM*), work differently. SFM is intended to be "simple" so it hides a lot of programming language bits.

Diferenças:

- In SFM the value of the column is always passed as an "invisible" first argument to a function included in the template.
- SFM doesn't support the difference between variables and strings; all values are strings.
- The following SFM template returns either the series name or the string «no series»:

```
{series:ifempty(no series)}
```

  The equivalent template in *TPM* is

```
{series:'ifempty($, 'no series')'}
```

  The equivalent template in *GPM* is:

```
program: ifempty(field('series'), 'no series')
```

  The first argument to `ifempty` is the value of the field `series`. The second argument is the string `no series`. In SFM the first argument, the value of the field, is automatically passed (the invisible argument).

- Several template functions, for example `booksize()` and `current_library_name()`, take no arguments. Because of the "invisible argument" you cannot use these functions in SFM.
- Nested functions, where a function calls another function to compute an argument, cannot be used in SFM. For example this template, intended to return the first 5 characters of the series value uppercased, won't work in SFM:

```
{series:uppercase(substr(0,5))}
```

- *TPM* and *GPM* support nested functions. The above template in *TPM* would be:

```
{series:'uppercase(substr($, 0,5))'}
```

  In *GPM* it would be:

```
program: uppercase(substr(field('series'), 0,5))
```

- As noted in the above *Template Program Mode* (página 175) section, using `{` and `}` characters in *TPM* string literals can lead to errors or unexpected results because they confuse the template processor. It tries to treat them as template boundaries, not characters. In some but not all cases you can replace a `{` with `[[` and a `}` with *]]*. Generally, if your program contains `{` and `}` characters then you should use *General Program Mode*.

10.3.11 User-defined Python template functions

You can add your own Python functions to the template processor. Such functions can be used in any of the three template programming modes. The functions are added by going to *Preferences* → *Advanced* → *Template functions*. Instructions are shown in that dialog.

10.3.12 Notas especiais para modelos de exportar/enviar

Special processing is applied when a template is used in a *save to disk* or *send to device* template. The values of the fields are cleaned, replacing characters that are special to file systems with underscores, including slashes. This means that field text cannot be used to create folders. However, slashes are not changed in prefix or suffix strings, so slashes in these strings will cause folders to be created. Because of this, you can create variable-depth folder structure.

For example, assume we want the folder structure *series/series_index - title*, with the caveat that if series does not exist, then the title should be in the top folder. The template to do this is:

```
{series:||/}{series_index:|| - }{title}
```

A barra inclinada e o hífen aparecem apenas se a série não estiver vazia.

The lookup function lets us do even fancier processing. For example, assume that if a book has a series, then we want the folder structure *series/series index - title.fmt*. If the book does not have a series then we want the folder structure *genre/author_sort/title.fmt*. If the book has no genre then we want to use "Unknown". We want two completely different paths, depending on the value of series.

Para obter isto, fazemos:

1. Create a composite field (give it lookup name #aa) containing `{series}/{series_index} - {title}`. If the series is not empty, then this template will produce *series/series_index - title*.
2. Create a composite field (give it lookup name #bb) containing `{#genre:ifempty(Unknown)}/{author_sort}/{title}`. This template produces *genre/author_sort/title*, where an empty genre is replaced with *Unknown*.
3. Set the save template to `{series:lookup(.,#aa,#bb}`. This template chooses composite field `#aa` if series is not empty and composite field `#bb` if series is empty. We therefore have two completely different save paths, depending on whether or not *series* is empty.

10.3.13 Modelos e conectores

Plugboards are used for changing the metadata written into books during send-to-device and save-to-disk operations. A plugboard permits you to specify a template to provide the data to write into the book's metadata. You can use plugboards to modify the following fields: authors, author_sort, language, publisher, tags, title, title_sort. This feature helps people who want to use different metadata in books on devices to solve sorting or display issues.

When you create a plugboard, you specify the format and device for which the plugboard is to be used. A special device is provided, `save_to_disk`, that is used when saving formats (as opposed to sending them to a device). Once you have chosen the format and device, you choose the metadata fields to change, providing templates to supply the new values. These templates are *connected* to their destination fields, hence the name *plugboards*. You can of course use composite columns in these templates.

When a plugboard might apply (Content server, save to disk, or send to device), calibre searches the defined plugboards to choose the correct one for the given format and device. For example, to find the appropriate plugboard for an EPUB book being sent to an ANDROID device, calibre searches the plugboards using the following search order:

- a plugboard with an exact match on format and device, e.g., `EPUB` and `ANDROID`

- a plugboard with an exact match on format and the special `any device` choice, e.g., `EPUB` and `any device`
- a plugboard with the special `any format` choice and an exact match on device, e.g., `any format` and `ANDROID`
- um conector para `qualquer formato` e `qualquer dispositivo`

The tags and authors fields have special treatment, because both of these fields can hold more than one item. A book can have many tags and many authors. When you specify that one of these two fields is to be changed, the template's result is examined to see if more than one item is there. For tags, the result is cut apart wherever calibre finds a comma. For example, if the template produces the value `Thriller, Horror`, then the result will be two tags, `Thriller` and `Horror`. There is no way to put a comma in the middle of a tag.

The same thing happens for authors, but using a different character for the cut, a *&* (ampersand) instead of a comma. For example, if the template produces the value `Blogs, Joe&Posts, Susan`, then the book will end up with two authors, `Blogs, Joe` and `Posts, Susan`. If the template produces the value `Blogs, Joe;Posts, Susan`, then the book will have one author with a rather strange name.

Plugboards affect the metadata written into the book when it is saved to disk or written to the device. Plugboards do not affect the metadata used by `save to disk` and `send to device` to create the file names. Instead, file names are constructed using the templates entered on the appropriate preferences window.

10.3.14 Sugestões

- Use the Template Tester to test templates. Add the tester to the context menu for books in the library and/or give it a keyboard shortcut.
- Templates can use other templates by referencing composite columns built with the desired template. Alternatively, you can use Stored Templates.
- In a plugboard, you can set a field to empty (or whatever is equivalent to empty) by using the special template `{}`. This template will always evaluate to an empty string.
- The technique described above to show numbers even if they have a zero value works with the standard field series_index.

10.3.15 Referência para funções

Reference for all built-in template language functions

Here, we document all the built-in functions available in the calibre template language. Every function is implemented as a class in python and you can click the source links to see the source code, in case the documentation is insufficient. The functions are arranged in logical groups by type.

- *Arithmetic* (página 185)
  - *add(x [,y]*)* (página 185)
  - *ceiling(x)* (página 185)
  - *divide(x, y)* (página 185)
  - *floor(x)* (página 185)
  - *fraction_part(x)* (página 185)
  - *mod(x)* (página 186)

    - *multiply(x [, y]*)* (página 186)
    - *round(x)* (página 186)
    - *subtract(x, y)* (página 186)
- *Boolean* (página 186)
    - *and(value [, value]*)* (página 186)
    - *not(value)* (página 186)
    - *or(value [, value]*)* (página 187)
- *Date functions* (página 187)
    - *date_arithmetic(data, calc_spec, fmt)* (página 187)
    - *days_between(date1, date2)* (página 187)
    - *today()* (página 187)
- *Formatting values* (página 187)
    - *finish_formatting(val, fmt, prefix, suffix)* (página 187)
    - *format_date(val, format_string)* (página 188)
    - *format_number(v, template)* (página 188)
    - *human_readable(v)* (página 188)
    - *rating_to_stars(value, use_half_stars)* (página 188)
    - *urls_from_identifiers(identificadores, ordenar_resultados)* (página 188)
- *Get values from metadata* (página 189)
    - *annotation_count ()* (página 189)
    - *approximate_formats()* (página 189)
    - *author_links(val_separator, pair_separator)* (página 189)
    - *author_sorts(val_separator)* (página 189)
    - *booksize()* (página 189)
    - *connected_device_name(storage_location)* (página 190)
    - *connected_device_uuid(storage_location)* (página 190)
    - *current_library_name()* (página 190)
    - *current_library_path()* (página 190)
    - *current_virtual_library_name()* (página 190)
    - *field(lookup_name)* (página 190)
    - *formats_modtimes(date_format)* (página 191)
    - *formats_paths()* (página 191)
    - *formats_sizes()* (página 191)
    - *has_cover()* (página 191)
    - *is_marked()* (página 191)

- *language_codes(lang_strings)* (página 191)
- *language_strings(lang_codes, localize)* (página 192)
- *ondevice()* (página 192)
- *raw_field(lookup_name [, optional_default])* (página 192)
- *raw_list(lookup_name, separator)* (página 192)
- *series_sort()* (página 192)
- *user_categories()* (página 192)
- *virtual_libraries()* (página 193)

- *If-then-else* (página 193)
  - *check_yes_no(nome_do_campo, é_indefinido, é_falso, é_verdadeiro)* (página 193)
  - *contains(val, pattern, text if match, text if not match)* (página 193)
  - *field_exists(nome_campo)* (página 193)
  - *ifempty(val, texto se vazio)* (página 193)
  - *test(val, texto se não vazio, texto se vazio)* (página 194)
- *Iterating over values* (página 194)
  - *first_non_empty(value, value, …)* (página 194)
  - *lookup(val, [pattern, field,]+ else_field)* (página 194)
  - *switch(val, [pattern, value,]+ else_value)* (página 194)
  - *switch_if([test_expression, value_expression,]+ else_expression)* (página 194)
- *List lookup* (página 194)
  - *identifier_in_list(val, id, found_val, not_found_val)* (página 194)
  - *in_list(val, separador, [padrao, val_encontrado,]+ val_nao_encontrado)* (página 195)
  - *list_item(val, index, separator)* (página 195)
  - *select(key)* (página 195)
  - *str_in_list(val, separador, [string, val_encontrado,]+ val_nao_encontrado)* (página 195)
- *List manipulation* (página 195)
  - *count(val, separator)* (página 195)
  - *list_count_matching(lista, padrao, separador)* (página 196)
  - *list_difference(list1, list2, separator)* (página 196)
  - *list_equals(list1, sep1, list2, sep2, yes_val, no_val)* (página 196)
  - *list_intersection(list1, list2, separator)* (página 196)
  - *list_join(with_separator, list1, separator1 [, list2, separator2]*)* (página 196)
  - *list_re(src_list, separator, include_re, opt_replace)* (página 197)
  - *list_re_group(src_list, separador, include_re, search_re [, group_template]+)* (página 197)
  - *list_remove_duplicates(lista, separador)* (página 197)

- *list_sort(list, direction, separator)* (página 197)
- *list_split(list_val, sep, id_prefix)* (página 197)
- *list_union(list1, list2, separator)* (página 198)
- *range(start, stop, step, limit)* (página 198)
- *subitems(val, start_index, end_index)* (página 198)
- *sublist(val, start_index, end_index, separator)* (página 198)

Other (página 199)
- *arguments(id[=expression] [, id[=expression]]*)* (página 199)
- *assign(id, val)* (página 199)
- *globals(id[=expressão] [, id[=expressão]]*)* (página 199)
- *print(a[, b]*)* (página 199)

Recursion (página 199)
- *eval(template)* (página 199)
- *template(x)* (página 200)

Relational (página 200)
- *cmp(x, y, lt, eq, gt)* (página 200)
- *first_matching_cmp(val, cmp1, result1,]+, else_result)* (página 200)
- *strcmp(x, y, lt, eq, gt)* (página 200)
- *strcmpcase(x, y, lt, eq, gt)* (página 200)

String case changes (página 201)
- *capitalize(val)* (página 201)
- *lowercase(val)* (página 201)
- *titlecase(val)* (página 201)
- *uppercase(val)* (página 201)

String manipulation (página 201)
- *character(character_name)* (página 201)
- *re(val, pattern, replacement)* (página 201)
- *re_group(val, pattern [, template_for_group]*)* (página 202)
- *shorten(val, left chars, middle text, right chars)* (página 202)
- *strcat(a [, b]*)* (página 202)
- *strcat_max(max, string1 [, prefix2, string2]*)* (página 202)
- *strlen(a)* (página 202)
- *substr(str, start, end)* (página 203)
- *swap_around_articles(val, separator)* (página 203)
- *swap_around_comma(val)* (página 203)

- *to_hex(val)* (página 203)
- *transliterate(a)* (página 203)

- *Template database functions* (página 203)
  - *book_count(query, use_vl)* (página 203)
  - *book_values(column, query, sep, use_vl)* (página 204)
- *other* (página 204)
  - *set_globals(id[=expression] [, id[=expression]]*)* (página 204)
- *API of the Metadata objects* (página 204)

Arithmetic

add(x [,y]*)

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

add(x [,y]*) – devolve a soma dos seus argumentos. Lança uma exceção se um argumento não for um número. Esta função pode, muitas vezes, ser substituída pelo operador +.

ceiling(x)

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

ceiling(x) – devolve o menor inteiro maior que ou igual a x. Devolve uma exceção se x não for um número.

divide(x, y)

class calibre.utils.formatter_functions.**BuiltinDivide**

divide(x, y) – devolve x - y. Devolve uma exceção se x ou y não forem números. Muitas vezes, esta função pode ser substituída pelo operador / .

floor(x)

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

floor(x) – devolve o maior inteiro que seja menor que ou igual a x . Devolve uma exceção se x não for um número.

fraction_part(x)

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

fraction_part(x) – devolve o valor existente depois da casa decimal. Por exemplo, fractional_part(3.14) devolve 0.14. Lança uma exceção caso x não seja um número.

mod(x)

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

mod(x) – devolve floor(remainder of x / y). Devolve uma exceção se x ou y não forem números.

multiply(x [, y]*)

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

multiply(x [, y]*) – devolve x - y. Devolve uma exceção se algum dos argumentos não forem números. Muitas vezes, esta função pode ser substituída pelo operador * .

round(x)

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

round(x) – devolve o inteiro mais próximo de x. Devolve uma exceção se x não for um número.

subtract(x, y)

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

subtract(x, y) – devolve x - y. Devolve uma exceção se x ou y não forem números. Muitas vezes, esta função pode ser substituída pelo operador - .

Boolean

and(value [, value]*)

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

and(value [, value]*) – devolve a string «1» se todos os valores não estiverem vazios, caso contrário, devolve a string vazia. Esta função funciona bem com o teste ou first_non_empty. Pode ter quantos valores quiser. Em muitos casos o operador && pode substituir esta função.

not(value)

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

not(value) – devolve a string «1» se o valor estiver vazio, caso contrário, devolve a string vazia. Esta função funciona bem com o teste ou first_non_empty. Em muitos casos o operador ! pode substituir esta função.

or(value [, value]*)

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

or(value [, value]*) – devolve a string «1» se algum valor não estiver vazio, caso contrário, devolve a string vazia. Esta função funciona bem com o teste ou first_non_empty. Em muitos casos o operador || pode substituir esta função.

Date functions

date_arithmetic(data, calc_spec, fmt)

class calibre.utils.formatter_functions.**BuiltinDateArithmetic**

date_arithmetic(data, calc_spec, fmt) – Calcula uma nova data a partir de "data" usando "calc_spec". Devolve a nova data formatada de acordo com o parâmetro opcional "fmt": se não fornecido, então o resultado estará no formato iso. O "calc_spec" é uma sequência formada pela concatenação de pares de "vQ" (valorQue), onde "v" é número que pode ser negativo e Q é uma das letras a seguir: s: adiciona "v" segundos à "data" m: adiciona "v" minutos à "data" h: adiciona "v" horas à "data" d: adiciona "v" dias à "data" w: adiciona "v" semanas à "data" y: adiciona "v" anos à "data", em que um ano são 365 dias. Exemplo: "1s3d-1m" irá adicionar 1 segundo e 3 dias, e subtrai 1 minuto a partir da "data".

days_between(date1, date2)

class calibre.utils.formatter_functions.**BuiltinDaysBetween**

days_between(date1, date2) – devolve o número de dias entre date1 e date2. O número é positivo se date1 for maior do que date2, caso contrário, é negativo. Se date1 ou date2 não forem datas, a função devolve a string vazia.

today()

class calibre.utils.formatter_functions.**BuiltinToday**

today() – devolve um string com a data de hoje. Este valor é destinado ao uso em format_date e days_between, mas pode ser manipulado como cualquer outro string. A data está em formato ISO.

Formatting values

finish_formatting(val, fmt, prefix, suffix)

class calibre.utils.formatter_functions.**BuiltinFinishFormatting**

finish_formatting(val, fmt, prefix, suffix) – aplicar o formato, de prefixo, sufixo e para um valor da mesma forma como foi feito num modelo como *{series_index:05.2f| - |- }*. Por exemplo, o seguinte programa produz a mesma saída que o modelo acima: programa: finish_formatting(field(«series_index»), «05.2f», « - «, « - «)

format_date(val, format_string)

class calibre.utils.formatter_functions.**BuiltinFormatDate**

format_date(val, format_string) – formato do valor, o qual deve ser uma data, usando a_string do formato, retornando uma string. Os formatos dos códigos são: d : dia sem o número zero à esquerda (1 à 31); dd : dia como número e com um zero à esquerda (01 à 31); ddd : nome do dia traduzido abreviado (por exemplo, «Seg» a «Dom»); dddd : nome longo do dia traduzido abreviado (por exemplo, «Segunda» a «Domingo»); M : mês como um número e sem o zero à esquerda (1 à 12); MM : mês como número e com um zero à esquerda (01 à 12); MMM : nome do mês traduzido abreviado (por exemplo, «Jan» a «Dez»); MMMM : nome longo do mês traduzido (por exemplo, «Janeiro» a «Dezembro»); yy : ano como dois dígitos numerais (00 à 99); yyyy : ano como quatro dígitos numerais. h : horas sem 0 à esquerda (0 a 11 ou 0 a 23, dependendo de am/pm); hh : horas com 0 à esquerda (00 a 11 ou 00 a 23, dependendo de am/pm); m : minutos sem 0 à esquerda (0 a 59); mm : minutos com 0 à esquerda (00 a 59) s : segundos sem 0 à esquerda (0 a 59); ss : segundos com 0 à esquerda (00 à 59); ap : usar um relógio de 12-horas em vez de um relógio de 24-horas, com «ap» substituído pela string traduzida por am ou pm; AP : usar um relógio de 12-horas em vez de um relógio de 24-horas, com «AP» substituído pela string traduzida por AM ou PM; iso : data com hora e fuso-horário. Deve ser o único formato presente: to_number - data como número de ponto flutuante; from_number[:fmt] - formatar o carimbo de data usando fmt se presente, ou usar iso se ausente.

format_number(v, template)

class calibre.utils.formatter_functions.**BuiltinFormatNumber**

format_number(v, template) – formatar o número v usando um modelo de formatação Python como «{0:5.2f}» ou «{0:,d}» ou «${0:5,.2f}». A parte field_name do modelo deve ser um 0 (zero) (o «{0:» nos exemplos acima). Consulte a documentação do modelo de linguagem e do Python para mais exemplos. Você pode deixar de fora o «{0:» inicial e o «}» final se o modelo contiver apenas um formato. Devolve a string vazia se a formatação falhar.

human_readable(v)

class calibre.utils.formatter_functions.**BuiltinHumanReadable**

human_readable(v) – devolve o string que representa o número v em KB, MB, GB, etc.

rating_to_stars(value, use_half_stars)

class calibre.utils.formatter_functions.**BuiltinRatingToStars**

rating_to_stars(value, use_half_stars) – Devolve a classificação como sequência de caracteres de estrelas. O valor é um número entre 0 e 5. Defina use_half_stars como 1 se desejar caracteres de meia estrela para colunas de classificações personalizadas que suportem classificações não inteiras, por exemplo 2.5.

urls_from_identifiers(identificadores, ordenar_resultados)

class calibre.utils.formatter_functions.**BuiltinUrlsFromIdentifiers**

urls_from_identifiers(identificadores, ordenar_resultados) – dada uma lista separada por vírgulas de identificadores, onde um identificador é um par de valores separado por dois pontos (nome:valor_do_id), devolve uma lista separada por vírgulas com URL HTML, gerada a partir dos identificadores. A lista não será ordenada caso ordenar_resultados for 0 (caractere ou número). Caso contrário, será ordenada alfabeticamente pelo nome do identificador. Os URL são gerados da mesma forma que a coluna de identificadores predefinidos ao ser exibida no painel de detalhes do livro.

Get values from metadata

annotation_count ()

class calibre.utils.formatter_functions.**BuiltinAnnotationCount**

annotation_count () - devolve o número total de anotações de todos os tipos anexados ao livro atual. Esta função funciona apenas na IGU.

approximate_formats()

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

approximate_formats() – devolve uma lista de valores, separados por vírgulas, de formatos que numa determinada altura foram associados ao livro. Não há garantia de que esta lista esteja correta, embora provavelmente o esteja. Esta função pode ser chamada no Modo de programa por modelos usando o modelo «{:"approximate_formats()"}». Note que os nomes de formato são sempre maiúsculas, como em EPUB. Esta função só funciona na IGU. Se quiser usar estes valores nos modelos exportar-para-disco ou enviar-para-dispositivo, então deve criar uma coluna personalizada «Coluna obtida a partir de outras colunas». Use a função no modelo dessa coluna e use o valor dessa coluna nos seus modelos de exportar/enviar

author_links(val_separator, pair_separator)

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

author_links(val_separator, pair_separator) – devolve uma sequência contendo uma lista de autores e valores de ligação desses autores na forma author1 val_separator author1link pair_separator author2 val_separator author2link etc. Um autor é separado de seu valor de ligação pela sequência val_separator sem espaços adicionais. Pares autor:linkvalue são separados pelo argumento de sequência pair_separator sem espaços adicionais. Cabe-lhe escolher sequências de separação que não ocorram em nomes de autores ou ligações. Um autor será incluído, mesmo se a ligação de autor estiver em branco.

author_sorts(val_separator)

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

author_sorts(val_separator) – devolve uma sequência contendo uma lista de valores ordenados de autor para os autores do livro. A ordenação é a presente nos metadados de autor (diferente do author_sort em livros). A lista devolvida tem a forma autor ordenação 1 val_separator autor ordenação 2, etc. Os valores de autor ordenado nesta lista estão na mesma ordem que os autores do livro. Se pretender espaços em volta de val_separator então inclua-os na sequência separador

booksize()

class calibre.utils.formatter_functions.**BuiltinBooksize**

booksize() – devolve o valor do campo de tamanho. Esta função funciona apenas na IGU. Se quiser usar este valor nos modelos exportar-para-disco ou enviar-para-dispositivo, então deve fazer uma coluna personalizada «Coluna obtida a partir de outras colunas», use a função no modelo da coluna e use o valor dessa coluna nos seus modelos de exportar/enviar

connected_device_name(storage_location)

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceName**

connected_device_name(storage_location) – se um dispositivo estiver conectado, então devolve o nome do dispositivo; caso contrário, devolve uma cadeia de carateres vazia. Cada local de armazenamento num dispositivo pode ter um nome diferente. Os nomes dos locais são “main”, “carda” e “cardb”. Esta função funciona apenas na interface gráfica.

connected_device_uuid(storage_location)

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceUUID**

connected_device_uuid(storage_location) – se um dispositivo estiver conectado, então devolve o uuid (id único) do dispositivo; caso contrário, devolve uma cadeia de carateres vazia. Cada local de armazenamento num dispositivo tem um uuid diferente. Os nomes dos locais são “main”, “carda” e “cardb”. Esta função funciona apenas na interface gráfica.

current_library_name()

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**

current_library_name() – retorna o último nome do caminho para a biblioteca atual do Calibre. Esta função pode ser chamada no Modo de programa por modelos usando o modelo «{:”current_library_name()”}».

current_library_path()

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryPath**

current_library_path() – devolve o caminho para a biblioteca atual do Calibre. Esta função pode ser chamada no Modo de programa por modelos usando o modelo «{:”current_library_path()”}».

current_virtual_library_name()

class calibre.utils.formatter_functions.**BuiltinCurrentVirtualLibraryName**

current_virtual_library_name() – devolve o nome da biblioteca virtual atual (se existir uma), ou então a string vazia. A maiusculização do nome da biblioteca são preservadas. Exemplo: «program: current_virtual_library_name()».

field(lookup_name)

class calibre.utils.formatter_functions.**BuiltinField**

field(lookup_name) – devolve o campo de metadados nomeado pelo nome de pesquisa.

formats_modtimes(date_format)

class calibre.utils.formatter_functions.**BuiltinFormatsModtimes**

formats_modtimes(date_format) – cria uma lista, separada por vírgulas, de itens separados por dois pontos que representam horas de alteração para os formatos de um livro. O parâmetro date_format especifica como a data deve ser formatada. Ver a função date_format para mais detalhes. Pode usar a função de seleção para obter a hora de alteração para um formato específico. Note que os nomes dos formatos são sempre em maiúsculas, tal como em EPUB.

formats_paths()

class calibre.utils.formatter_functions.**BuiltinFormatsPaths**

formats_paths() – cria uma lista de itens, separados por dois pontos, que representam o caminho completo para os formatos de um livro. Pode usar a função de seleção para obter o caminho para um formato específico. Note que os nomes dos formatos são sempre em maiúsculas, tal como em EPUB.

formats_sizes()

class calibre.utils.formatter_functions.**BuiltinFormatsSizes**

formats_sizes() – cria uma lista de itens separados por dois pontos que representam tamanhos em bytes dos formatos de um livro. Pode usar a função de seleção para obter o tamanho de um formato específico. Note que os nomes dos formatos são sempre em maiúsculas, tal como em EPUB.

has_cover()

class calibre.utils.formatter_functions.**BuiltinHasCover**

has_cover() – devolve Sim se o livro tiver capa, caso contrário devolve um string vazio

is_marked()

class calibre.utils.formatter_functions.**BuiltinIsMarked**

is_marked() – verifica se o livro está marcado (“marked”) no Calibre. Se estiver, então devolve o valor da marcação, que será “true” ou uma lista separada por vírgulas de nomes das marcações. Devolve “” se o livro não estiver marcado.

language_codes(lang_strings)

class calibre.utils.formatter_functions.**BuiltinLanguageCodes**

language_codes(lang_strings) – mostra os códigos de idioma para as «strings» enviados a lang_strings. Estas strings tem de ser no mesmo idioma do idioma atual. Lang_strings é uma lista separada por vírgulas.

language_strings(lang_codes, localize)

class calibre.utils.formatter_functions.**BuiltinLanguageStrings**

language_strings(lang_codes, localize) – retornar as strings para os códigos de idioma passados em lang_codes. Se localize é zero, retornar as strings em Inglês. Se localize não é zero, retornar as strings no idioma da localidade atual. Lang_codes é uma lista separada por vírgula.

ondevice()

class calibre.utils.formatter_functions.**BuiltinOndevice**

ondevice() – devolve Yes se ondevice estiver definido, caso contrário devolve uma string vazia. Esta função aplica-se apenas à IGU. Se quiser usar este valor nos modelos exportar-para-disco ou enviar-para-dispositivo, então deve criar uma coluna personalizada «Coluna obtida a partir de outras colunas», usar a função no modelo dessa coluna, e usar o valor dessa coluna nos seus modelos de exportar/enviar

raw_field(lookup_name [, optional_default])

class calibre.utils.formatter_functions.**BuiltinRawField**

raw_field(lookup_name [, optional_default]) – devolve a lista de metadados nomeada por nome de pesquisa mas sem aplicar qualquer formatação. Irá avaliar e devolver o segundo argumento opcional "default" se o campo estiver como indefinido ("None")

raw_list(lookup_name, separator)

class calibre.utils.formatter_functions.**BuiltinRawList**

raw_list(lookup_name, separator) – devolve a lista de metadados nomeada por nome de pesquisa mas sem aplicar qualquer formatação ou ordenação e com itens separados por *separator*.

series_sort()

class calibre.utils.formatter_functions.**BuiltinSeriesSort**

series_sort() – devolve o valor de ordenação da série

user_categories()

class calibre.utils.formatter_functions.**BuiltinUserCategories**

user_categories() – obter uma lista de categorias do utilizador que contenham este livro, separados por vírgulas. Esta função funciona apenas na GUI. Se quiser usar esses valores nos modelos exportar-para-disco ou enviar-para-dispositivo então deve criar uma «Coluna obtida a partir de outras colunas» personalizada: use a função no modelo da coluna e use o valor dessa coluna nos seus modelos de exportar/enviar

virtual_libraries()

class calibre.utils.formatter_functions.**BuiltinVirtualLibraries**

virtual_libraries() – devolve uma lista de bibliotecas virtuais, separadas por vírgulas, que contenham este livro. Esta função aplica-se apenas à IGU. Se quiser usar estes valores nos modelos exportar-para-disco ou enviar-para--dispositivo então deve criar uma coluna personalizada «Coluna obtida a partir de outras colunas», usar a função no modelo dessa coluna, e usar o valor dessa coluna nos seus modelos de exportar/enviar

If-then-else

check_yes_no(nome_do_campo, é_indefinido, é_falso, é_verdadeiro)

class calibre.utils.formatter_functions.**BuiltinCheckYesNo**

check_yes_no(nome_do_campo, é_indefinido, é_falso, é_verdadeiro) – verifica o valor do campo sim/não definido pela chave de pesquisa nome_do_campo de acordo com os valores especificados pelos parâmetros, devolvendo «sim» se uma correspondência for encontrada, ou devolvendo vazio caso contrário. Defina os parâmetros é_indefinido, é_falso, ou é_verdadeiro como 1 (o número) para verificar essa condição; caso contrário, defina como 0. Exemplo: check_yes_no(«#bool», 1, 0, 1) devolve «sim» se o campo sim/não «#bool» estiver não definido (nem Verdadeiro, nem Falso) ou Verdadeiro. Mais de um parâmetro (é_indefinido, é_falso ou é_verdadeiro) podem ser definidos como 1. Esta função geralmente é utilizada pelas funções test() ou is_empty().

contains(val, pattern, text if match, text if not match)

class calibre.utils.formatter_functions.**BuiltinContains**

contains(val, pattern, text if match, text if not match) – verifica se o val contém correspondências para a expressão regular *pattern*. Devolve *text if match* se forem encontradas combinações, caso contrário devolve *text if no match*

field_exists(nome_campo)

class calibre.utils.formatter_functions.**BuiltinFieldExists**

field_exists(nome_campo) – verifica se um campo (coluna) com o nome "nome_campo" existe, devolvendo "1" se sim e "" se não.

ifempty(val, texto se vazio)

class calibre.utils.formatter_functions.**BuiltinIfempty**

ifempty(val, texto se vazio) – devolve val se val não estiver vazio, caso contrário devolve o texto definido para «texto se vazio»

test(val, texto se não vazio, texto se vazio)

class calibre.utils.formatter_functions.**BuiltinTest**

test(val, texto se não vazio, texto se vazio) – devolve *texto se não vazio* se o val não estiver vazio, caso contrário devolve "texto se vazio".

Iterating over values

first_non_empty(value, value, ...)

class calibre.utils.formatter_functions.**BuiltinFirstNonEmpty**

first_non_empty(value, value, …) – devolve o primeiro valor que não esteja vazio. Se todos os valores estiverem vazios, então é devolvida a string vazia. Pode ter quantos valores quiser.

lookup(val, [pattern, field,]+ else_field)

class calibre.utils.formatter_functions.**BuiltinLookup**

lookup(val, [pattern, field,]+ else_field) – como comando adicional, exceto que os argumentos são nomes de campos (metadados) e não texto. O valor do campo apropriado será obtido e usado. Note que, como as colunas compostas são campos, pode usar esta função num campo composto para usar o valor de algum outro campo composto. Isto é extremamente útil quando construir variáveis de caminhos para onde guardar

switch(val, [pattern, value,]+ else_value)

class calibre.utils.formatter_functions.**BuiltinSwitch**

switch(val, [pattern, value,]+ else_value) – para cada par *pattern, value*, verifica se *val* corresponde à expressão regular *pattern* e se assim for, devolve *value*. Se nenhum padrão for correspondido, então é devolvido *else_value*. Pode ter quantos pares *pattern, value* quiser

switch_if([test_expression, value_expression,]+ else_expression)

class calibre.utils.formatter_functions.**BuiltinSwitchIf**

switch_if([test_expression, value_expression,]+ else_expression) – for each «test_expression, value_expression» pair, checks if test_expression is True (non-empty) and if so returns the result of value_expression. If no test_expression is True then the result of else_expression is returned. You can have as many «test_expression, value_expression» pairs as you want.

List lookup

identifier_in_list(val, id, found_val, not_found_val)

class calibre.utils.formatter_functions.**BuiltinIdentifierInList**

identifier_in_list(val, id, found_val, not_found_val) – tratar val como uma lista de identificadores separados por vírgulas, comparando a string contra cada valor na lista. Um identificador tem o formato «identifier:value». O parâmetro id deve ser «id» ou «id:regexp». O primeiro caso corresponde, se houver qualquer identificador com esse id. O segundo caso corresponde se a expressão regular corresponde ao valor do identificador. Se houver uma correspondência, retorna found_val, caso contrário retorna not_found_val.

in_list(val, separador, [padrao, val_encontrado,]+ val_nao_encontrado)

class calibre.utils.formatter_functions.**BuiltinInList**

in_list(val, separador, [padrao, val_encontrado,]+ val_nao_encontrado) – tratando val como uma lista de itens separados por separador, caso o padrão corresponda a algum dos valores da lista, então devolve val_encontrado. Caso o padrão não corresponda a nenhum valor da lista, então devolve val_nao_encontrado. Os pares "padrao" e "val_encontrado" podem ser repetidos quantas vezes desejar. Os padrões são verificados por ordem. O val_encontrado para a primeira correspondência é devolvido. Variantes: in_list(), list_contains()

list_item(val, index, separator)

class calibre.utils.formatter_functions.**BuiltinListitem**

list_item(val, index, separator) – interpreta o valor como uma lista de itens separados por *separator*, devolvendo o item *index*. O primeiro item é o número zero. O último item pode ser obtido usando *list_item(-1,separator)*. Se o item não estiver na lista, então é devolvido o valor vazio. O separador tem o mesmo significado que na função de contagem.

select(key)

class calibre.utils.formatter_functions.**BuiltinSelect**

select(key) – interpreta o campo como uma lista de itens separados por vírgulas, com os itens na forma «id:value». Encontra o emparelhamento do ID igual a KEY, e devolve o valor correspondente. Devolve um valor vazio se nenhuma correspondência for encontrada.

str_in_list(val, separador, [string, val_encontrado,]+ val_nao_encontrado)

class calibre.utils.formatter_functions.**BuiltinStrInList**

str_in_list(val, separador, [string, val_encontrado,]+ val_nao_encontrado) – tratando "val" como uma lista de itens separados por "separador", caso a string corresponda a algum dos valores da lista, então devolve val_encontrado. Caso a string não corresponda a nenhum dos valores da lista, então devolve val_nao_encontrado. A comparação é do tipo exata (não contém) e não diferencia maiúsculas/minúsculas. Os pares "string" and "val_encontrado" podem ser repetidos quantas vezes desejar. Os padrões são verificados por ordem. É devolvido o "val_encontrado" para a primeira correspondência.

List manipulation

count(val, separator)

class calibre.utils.formatter_functions.**BuiltinCount**

count(val, separator) – interpreta o valor como uma lista de itens separados por *separator*, devolvendo o número de itens na lista. A maioria das listas usam uma vírgula como separador, mas em autores usa-se um E comercial. Exemplos: {tags:count(,)}, {authors:count(&)}. Nome curto: count(), list_count()

list_count_matching(lista, padrao, separador)

class calibre.utils.formatter_functions.**BuiltinListCountMatching**

list_count_matching(lista, padrao, separador) – interpreta "lista" como uma lista de itens separados por "separador", devolvendo o número de itens na lista que correspondam à expressão regular "padrao". Variantes: list_count_matching(), count_matching()

list_difference(list1, list2, separator)

class calibre.utils.formatter_functions.**BuiltinListDifference**

list_difference(list1, list2, separator) – devolve uma lista feita ao remover da list1 qualquer item encontrado em list2, usando uma comparação não sensível a maiúsculas. Os itens em list1 e list2 são separados por separador, tal como estão os itens da lista devolvida.

list_equals(list1, sep1, list2, sep2, yes_val, no_val)

class calibre.utils.formatter_functions.**BuiltinListEquals**

list_equals(list1, sep1, list2, sep2, yes_val, no_val) – devolverá yes_val se list1 e list2 contiverem os mesmos itens, caso contrário devolverá no_val. Os itens são determinados dividindo cada lista usando o carácter separador apropriado (sep1 ou sep2). A ordem dos itens das listas não é relevante. A comparação não é sensível a maiúsculas.

list_intersection(list1, list2, separator)

class calibre.utils.formatter_functions.**BuiltinListIntersection**

list_intersection(list1, list2, separator) – devolve uma lista feita ao remover da list1 qualquer item não encontrado em list2, usando uma comparação não sensível a maiúsculas. Os itens em list1 e list2 são separados por separador, tal como estão os itens da lista devolvida.

list_join(with_separator, list1, separator1 [, list2, separator2]*)

class calibre.utils.formatter_functions.**BuiltinListJoin**

list_join(with_separator, list1, separator1 [, list2, separator2]*) – return a list made by joining the items in the source lists (list1, etc) using with_separator between the items in the result list. Items in each source list[123…] are separated by the associated separator[123…]. A list can contain zero values. It can be a field like publisher that is single-valued, effectively a one-item list. Duplicates are removed using a case-insensitive comparison. Items are returned in the order they appear in the source lists. If items on lists differ only in letter case then the last is used. All separators can be more than one character. Example:

program:
list_join("#@#", $authors, "&", $tags, ",")

Pode usar list_join nos resultados de chamadas anteriores a list_join da seguinte forma:
program:

a = list_join("#@#", $authors, "&", $tags, ","); b = list_join("#@#", a, "#@#", $#genre, ",", $#people, "&")

Pode usar expressões para gerar uma lista. Por exemplo, imagine que você queira encontrar itens de autores e #genero. Com o género alterado para a palavra "Género: " seguida da primeira letra do género, ou seja, o género "Ficção" torna-se "Género: F". A expressão abaixo fará isto:

program:
list_join("#@#", $authors, "&", list_re($#genre, ",", "^(.).*$", "Genre: 1"), ",")

list_re(src_list, separator, include_re, opt_replace)

class calibre.utils.formatter_functions.**BuiltinListRe**

list_re(src_list, separator, include_re, opt_replace) – Construir uma lista, primeiro pela separação de src_list em itens usando o carácter separador. Para cada item da lista, verificar se ele corresponde a include_re. Se isso acontecer, então adicioná-lo à lista a ser obtida. Se opt_replace não é a string vazia, então aplicar a substituição antes de adicionar o item à lista obtida.

list_re_group(src_list, separador, include_re, search_re [, group_template]+)

class calibre.utils.formatter_functions.**BuiltinListReGroup**

list_re_group(src_list, separador, include_re, search_re [, group_template]+) – Como list_re exceto que as substituições não são opcionais. Usa re_group(list_item, search_re, group_template, …) ao fazer as substituições na lista resultante.

list_remove_duplicates(lista, separador)

class calibre.utils.formatter_functions.**BuiltinListRemoveDuplicates**

list_remove_duplicates(lista, separador) – devolve uma lista construída pela remoção de itens duplicados na lista fonte. Se os itens diferem apenas em minúsculas/maiúsculas, o último deles é devolvido. Os itens na lista fonte são separados pelo separador, assim como os itens na lista obtida.

list_sort(list, direction, separator)

class calibre.utils.formatter_functions.**BuiltinListSort**

list_sort(list, direction, separator) – devolve uma lista ordenada usando uma ordenação que não diferencia maiúsculas. Se a direção é zero, a lista é ordenada de forma crescente, caso contrário será por ordem decrescente. Os itens da lista são separados por separador, tal como o são os itens na lista devolvida.

list_split(list_val, sep, id_prefix)

class calibre.utils.formatter_functions.**BuiltinListSplit**

list_split(list_val, sep, id_prefix) – separa list_val em valores separados usando "sep", e então atribui os valores a variáveis nomeadas "id_prefix_N", onde N é a posição do valor na lista. O primeiro item tem uma posição 0 (zero). A função devolve o último elemento da lista. Exemplo: split("um:dois:blah", ":", "var") é o equivalente a var_0 = "um"; var_1 = "dois"; var_2 = "blah".

list_union(list1, list2, separator)

class calibre.utils.formatter_functions.**BuiltinListUnion**

list_union(list1, list2, separator) – devolve uma lista feita pela fusão dos itens da list1 e list2, removendo itens duplicados e sendo usada uma comparação não sensível a maiúsculas. Se os itens diferirem em maiúsculas é usado o primeiro da list1. Os itens em list1 e list2 são separados pelo separador, tal como estão os itens da lista devolvida. Nomes alternativos: list_union(), merge_lists()

range(start, stop, step, limit)

class calibre.utils.formatter_functions.**BuiltinRange**

range(start, stop, step, limit) – returns a list of numbers generated by looping over the range specified by the parameters start, stop, and step, with a maximum length of limit. The first value produced is "start". Subsequent values next_v are current_v+step. The loop continues while next_v < stop assuming step is positive, otherwise while next_v > stop. An empty list is produced if start fails the test: start>=stop if step is positive. The limit sets the maximum length of the list and has a default of 1000. The parameters start, step, and limit are optional. Calling range() with one argument specifies stop. Two arguments specify start and stop. Three arguments specify start, stop, and step. Four arguments specify start, stop, step and limit. Examples: range(5) -> "0,1,2,3,4". range(0,5) -> "0,1,2,3,4". range(-1,5) -> "-1,0,1,2,3,4". range(1,5) -> "1,2,3,4". range(1,5,2) -> "1,3". range(1,5,2,5) -> "1,3". range(1,5,2,1) -> error(limit exceeded).

subitems(val, start_index, end_index)

class calibre.utils.formatter_functions.**BuiltinSubitems**

subitems(val, start_index, end_index) – Esta função é usada para quebrar as listas de itens, tais como géneros. Ela interpreta o valor como uma lista de itens, onde cada item é uma lista separada por período separados por vírgula. Retorna uma nova lista feita pela primeira encontrando todos os itens separados por pontos, em seguida, para cada um desses itens extrair os componentes *start_index* para *end_index*, em seguida, combinando os resultados de volta juntos. O primeiro componente numa lista separada por período tem um índice de zero. Se o índice é negativo, então conta do fim da lista. Conforme um caso especial, um end_index de zero é considerado como o comprimento da lista. Exemplo usando o modo de tema básico e assumindo um valor de #genre «A.B.C»: {#genre:subitems(0,1)} retorna «A». {#genre:subitems(0,2)} retorna «A.B». {#genre:subitems(1,0)} retorna «B.C». Assumindo um valor de #genre «A.B.C, D.E.F», {#genre:subitems(0,1)} retorna «A, D». {#genre:subitems(0,2)} retorna «A.B, D.E»

sublist(val, start_index, end_index, separator)

class calibre.utils.formatter_functions.**BuiltinSublist**

sublist(val, start_index, end_index, separator) – interpretar o valor como uma lista de itens separados por *separator*, devolvendo uma nova lista criada a partir de *start_index* ao item *end_index*. O primeiro item é o número zero. Se o índice for negativo, então será contado a partir do fim da lista. Sendo um caso especial, um end_index de zero é considerado como o comprimento da lista. Exemplos, usando o modo de modelo básico, assumindo que a coluna de etiquetas (que é separada por vírgulas) contém «A, B, C»: {tags:sublist(0,1,\,)} devolve «A»; {tags:sublist(-1,0,\,)} devolve»C»; {tags:sublist(0,-1,\,)} retorna «A, B».

Other

arguments(id[=expression] [, id[=expression]]*)

class calibre.utils.formatter_functions.**BuiltinArguments**

arguments(id[=expression] [, id[=expression]]*) – Used in a stored template to retrieve the arguments passed in the call. It both declares and initializes local variables, effectively parameters. The variables are positional; they get the value of the parameter given in the call in the same position. If the corresponding parameter is not provided in the call then arguments assigns that variable the provided default value. If there is no default value then the variable is set to the empty string.

assign(id, val)

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

assign(id, val) – atribui VAL a ID, e depois devolve VAL. O ID deve ser um identificador e não uma expressão. Esta função pode, muitas vezes, ser substituída pelo operador =

globals(id[=expressão] [, id[=expressão]]*)

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

globals(id[=expressão] [, id[=expressão]]*) – Recupera «variáveis globais» que podem ser passadas para o formatador. Tanto declara como inicializa variáveis locais com os nomes das variáveis globais passadas. Se a variável correspondente não for fornecida nas variáveis globais passadas, então atribui à variável o valor predefinido fornecido. Caso não haja valor predefinido, então a variável será definida como uma cadeia de carateres vazia.

print(a[, b]*)

class calibre.utils.formatter_functions.**BuiltinPrint**

print(a[, b]*) – imprime os argumentos para a saída predefinida. A menos que inicie o Calibre a partir da linha de comandos (calibre-debug -g), a saída irá para um buraco negro.

Recursion

eval(template)

class calibre.utils.formatter_functions.**BuiltinEval**

eval(template) – avalia o tema, passando as variáveis locais (those "assign'ed to) em vez dos metadados do livro. Isto permite usar o processador de tema para a construção de resultados complexos de variáveis locais. Como os caracteres { and } são especiais, deve usar o carácter [[for the { character and]] for the }; então eles serão convertidos automaticamente. Note também que prefixos e sufixos (the |*prefix*|*suffix* syntax) não podem ser usados no argumento para esta função ao usar o Modo de programa por modelos.

template(x)

class calibre.utils.formatter_functions.**BuiltinTemplate**

template(x) – analisa x como um modelo. A analise é feita em seu próprio contexto, o que significa que as variáveis não são compartilhadas entre a chamador e a análise do modelo. Como os caracteres { and } são especiais, deve usar o carácter [[for the { character and]] for the }. Eles serão convertidos automaticamente. Por exemplo, template("[[title_sort]]") irá analisar o tema {title_sort} e devolver o seu valor. Note também que prefixos e sufixos (the *|prefix|suffix* syntax) não podem ser usados no argumento para esta função ao usar o Modo de programa por modelos.

Relational

cmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinCmp**

cmp(x, y, lt, eq, gt) – compara x com y depois de os converter para números. Devolve lt se x < y. Devolve eq se x == y. Caso contrário devolve gt. Em muitos casos os operadores de comparação lexical (>#, <#, ==# etc) podem substituir esta função.

first_matching_cmp(val, cmp1, result1,]+, else_result)

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

first_matching_cmp(val, cmp1, result1,]+, else_result) – compara «val < cmpN» em sequência, devolvendo resultN para a primeira comparação com sucesso. Devolve else_result se não houver comparações com sucesso. Exemplo: first_matching_cmp(10,5,»small»,10,»middle»,15,»large»,»giant») devolve «large». O mesmo exemplo em que o primeiro valor seja 16 devolve «giant».

strcmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinStrcmp**

strcmp(x, y, lt, eq, gt) – faz uma comparação intensiva não diferenciadora de maiúsculas de x e y como strings. Devolve lt se x < y. Devolve eq se x == y. Caso contrário devolve gt. Em muitos casos os operadores de comparação lexical (>, <, == etc) podem substituir esta função.

strcmpcase(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinStrcmpcase**

strcmpcase(x, y, lt, eq, gt) – does a case-sensitive comparison of x and y as strings. Returns lt if x < y. Returns eq if x == y. Otherwise returns gt. Note: This is NOT the default behavior used by calibre, for example, in the lexical comparison operators (==, >, <, etc.). This function could cause unexpected results, preferably use strcmp() whenever possible.

String case changes

capitalize(val)

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

capitalize(val) – devolve o val maiusculizado

lowercase(val)

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

lowercase(val) – devolve o val em minúsculas

titlecase(val)

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

titlecase(val) – devolve o val em maiusculização de título

uppercase(val)

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

uppercase(val) – devolve o val em maiúsculas

String manipulation

character(character_name)

class calibre.utils.formatter_functions.**BuiltinCharacter**

character(character_name) – devolve o carácter nomeado por character_name. Por exemplo, character("newline") devolve o carácter de nova linha ("n"). Os nomes de caracteres suportados são "newline", "return", "tab", e "backslash".

re(val, pattern, replacement)

class calibre.utils.formatter_functions.**BuiltinRe**

re(val, pattern, replacement) – devolve o val depois de aplicar a expressão regular. Todas as instâncias de *pattern* são substituídas por *replacement*. Como em tudo no Calibre, estas são expressões regulares compatíveis com Python

re_group(val, pattern [, template_for_group]*)

class calibre.utils.formatter_functions.**BuiltinReGroup**

re_group(val, pattern [, template_for_group]*) – devolve uma string criada aplicando o padrão de expressão regular a val e substituindo cada instância correspondente com a string calculada ao substituir cada grupo correspondente pelo valor devolvido pelo modelo correspondente. O valor correspondente original para o grupo está disponível como $. No modo de programa por modelo, como para o tema e funções eval, você usa [[for { and]] for }. O exemplo seguinte, no modo de programação por modelo procura por séries com mais de uma palavra e a primeira palavra em maiúsculas: {series:"re_group($, «(S*)(.*)», «[[$:uppercase()]]», «[[$]]»)"}

shorten(val, left chars, middle text, right chars)

class calibre.utils.formatter_functions.**BuiltinShorten**

shorten(val, left chars, middle text, right chars) – Devolve uma versão abreviada de val, que consiste em caracteres *left chars* do início de val, seguido por *middle text*, seguido pelos caracteres *right chars* do fim da string. *Left chars* e *right chars* devem ser inteiros. Por exemplo, suponha que o título do livro é *Antigas leis inglesas nos tempos de Ivanhoe*, e quer que ele se ajuste a uma dimensão de, no máximo, 15 caracteres. Se usar {title:shorten(9,-,5)}, o resultado será *Antigas I-anhoe*. Se o comprimento do campo for menor que os caracteres à esquerda + caracteres à direita + comprimento de *middle text*, então o campo será usado intacto. Por exemplo, o título *A Cúpula* não pode ser alterado.

strcat(a [, b]*)

class calibre.utils.formatter_functions.**BuiltinStrcat**

strcat(a [, b]*) – pode receber qualquer número de argumentos. Devolve uma expressão formada pela concatenação de todos os argumentos.

strcat_max(max, string1 [, prefix2, string2]*)

class calibre.utils.formatter_functions.**BuiltinStrcatMax**

strcat_max(max, string1 [, prefix2, string2]*) – Devolve uma expressão formada pela concatenação dos argumentos. O valor devolvido é inicializado para string1. Pares *Prefix, string* são adicionados ao final do valor desde que o comprimento da expressão resultante seja inferior a *max*. String1 será devolvida mesmo se string1 for maior que max. Pode passar quantos pares *prefix, string* quiser.

strlen(a)

class calibre.utils.formatter_functions.**BuiltinStrlen**

strlen(a) – Devolve o comprimento do texto passando-o como um argumento.

substr(str, start, end)

class calibre.utils.formatter_functions.**BuiltinSubstr**

substr(str, start, end) – retorna start'th pelos caracteres end'th de str. O primeiro carácter str é o carácter zero'th. Se end é negativo, então isso indica que muitos caracteres são contados à direita. Se end é zero, então isso indica o último carácter . Por exemplo, substr("12345", 1, 0) retorna "2345", e substr("12345", 1, -1) retorna "234".

swap_around_articles(val, separator)

class calibre.utils.formatter_functions.**BuiltinSwapAroundArticles**

swap_around_articles(val, separator) – devolve o valor dos artigos movidos para o final. O valor pode ser uma lista, caso em que cada membro da lista será processado. Se o valor for uma lista, então você deve indicar o separador da lista de valores. Se não for indicado um separador, então o valor será tratado como sendo um valor único e não uma lista.

swap_around_comma(val)

class calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

swap_around_comma(val) – determinar um valor de forma «B, A», retorna «A B». Isto é muito útil para a conversão de nomes em formatos LN, FN para FN LN. Se não houver nenhuma vírgula, a função val retorna sem alteração

to_hex(val)

class calibre.utils.formatter_functions.**BuiltinToHex**

to_hex(val) – returns the string encoded in hex. This is useful when constructing calibre URLs.

transliterate(a)

class calibre.utils.formatter_functions.**BuiltinTransliterate**

transliterate(a) – Retorna uma string no alfabeto latino formado pelo som aproximado das palavras na sequência de origem. Por exemplo, se a origem for «Фёдор Михáйлович Достоевский» a função retorna «Fiodor Mikhailovich Dostoievskii».

Template database functions

book_count(query, use_vl)

class calibre.utils.formatter_functions.**BuiltinBookCount**

book_count(query, use_vl) – returns the count of books found by searching for query. If use_vl is 0 (zero) then virtual libraries are ignored. This function can be used only in the GUI.

book_values(column, query, sep, use_vl)

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

book_values(column, query, sep, use_vl) – returns a list of the values contained in the column «column», separated by «sep», in the books found by searching for «query». If use_vl is 0 (zero) then virtual libraries are ignored. This function can be used only in the GUI.

other

set_globals(id[=expression] [, id[=expression]]*)

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

set_globals(id[=expression] [, id[=expression]]*) – Sets «global variables» that can be passed into the formatter. The globals are given the name of the id passed in. The value of the id is used unless an expression is provided.

API of the Metadata objects

The python implementation of the template functions is passed in a Metadata object. Knowing it's API is useful if you want to define your own template functions.

class calibre.ebooks.metadata.book.base.**Metadata**(*title*, *authors=('Desconhecido',)*, *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()* (página 204) 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* (página 205).

Please keep the method based API of this class to a minimum. Every method becomes a reserved field name.

is_null(*field*)

Return True if the value of field is null in this object. "null" means it is unknown or evaluates to False. So a title of _("Unknown") is null or a language of "und" is null.

Be careful with numeric fields since this will return True for zero as well as None.

Also returns True if the field does not exist.

deepcopy(*class_generator=<function Metadata.<lambda>>*)

Do not use this method unless you know what you are doing, if you want to create a simple clone of this object, use `deepcopy_metadata()` instead. Class_generator must be a function that returns an instance of Metadata or a subclass of it.

get_identifiers()

Return a copy of the identifiers dictionary. The dict is small, and the penalty for using a reference where a copy is needed is large. Also, we don't want any manipulations of the returned dict to show up in the book.

set_identifiers(*identifiers*)

Set all identifiers. Note that if you previously set ISBN, calling this method will delete it.

set_identifier(*typ*, *val*)

If val is empty, deletes identifier of type typ

standard_field_keys()

return a list of all possible keys, even if this book doesn't have them

custom_field_keys()

return a list of the custom fields in this book

all_field_keys()

All field keys known by this instance, even if their value is None

metadata_for_field(*key*)

return metadata describing a standard or custom field.

all_non_none_fields()

Return a dictionary containing all non-None metadata fields, including the custom ones.

get_standard_metadata(*field*, *make_copy*)

return field metadata from the field if it is there. Otherwise return None. field is the key name, not the label. Return a copy if requested, just in case the user wants to change values in the dict.

get_all_standard_metadata(*make_copy*)

return a dict containing all the standard field metadata associated with the book.

get_all_user_metadata(*make_copy*)

return a dict containing all the custom field metadata associated with the book.

get_user_metadata(*field*, *make_copy*)

return field metadata from the object if it is there. Otherwise return None. field is the key name, not the label. Return a copy if requested, just in case the user wants to change values in the dict.

set_all_user_metadata(*metadata*)

store custom field metadata into the object. Field is the key name not the label

set_user_metadata(*field*, *metadata*)

store custom field metadata for one column into the object. Field is the key name not the label

remove_stale_user_metadata(*other_mi*)

Remove user metadata keys (custom column keys) if they don't exist in "other_mi", which must be a metadata object

template_to_attribute(*other*, *ops*)

Takes a list [(src,dest), (src,dest)], evaluates the template in the context of other, then copies the result to self[dest]. This is on a best-efforts basis. Some assignments can make no sense.

smart_update(*other*, *replace_metadata=False*)

Merge the information in *other* into self. In case of conflicts, the information in *other* takes precedence, unless the information in *other* is NULL.

format_field(*key*, *series_with_index=True*)

Returns the tuple (display_name, formatted_value)

to_html()

A HTML representation of this object.

calibre.ebooks.metadata.book.base.**STANDARD_METADATA_FIELDS**

The set of standard metadata fields.

```
'''
All fields must have a NULL value represented as None for simple types,
an empty list/dictionary for complex types and (None, None) for cover_data
'''

SOCIAL_METADATA_FIELDS = frozenset((
    'tags',             # Ordered list
    'rating',           # A floating point number between 0 and 10
    'comments',         # A simple HTML enabled string
    'series',           # A simple string
    'series_index',     # A floating point number
    # Of the form { scheme1:value1, scheme2:value2}
    # For example: {'isbn':'123456789', 'doi':'xxxx', ... }
    'identifiers',
))

'''
The list of names that convert to identifiers when in get and set.
'''

TOP_LEVEL_IDENTIFIERS = frozenset((
    'isbn',
))

PUBLICATION_METADATA_FIELDS = frozenset((
    'title',            # title must never be None. Should be _('Unknown')
    # Pseudo field that can be set, but if not set is auto generated
    # from title and languages
    'title_sort',
    'authors',          # Ordered list. Must never be None, can be [_('Unknown')]
    'author_sort_map',  # Map of sort strings for each author
    # Pseudo field that can be set, but if not set is auto generated
    # from authors and languages
    'author_sort',
    'book_producer',
    'timestamp',        # Dates and times must be timezone aware
    'pubdate',
    'last_modified',
    'rights',
    # So far only known publication type is periodical:calibre
    # 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',
))
```

(continues on next page)

(continuação da página anterior)

```
BOOK_STRUCTURE_FIELDS = frozenset((
    # These are used by code, Null values are None.
    'toc', 'spine', 'guide', 'manifest',
))

USER_METADATA_FIELDS = frozenset((
    # A dict of dicts similar to field_metadata. Each field description dict
    # also contains a value field with the key #value#.
    'user_metadata',
))

DEVICE_METADATA_FIELDS = frozenset((
    'device_collections',   # Ordered list of strings
    'lpath',                # Unicode, / separated
    'size',                 # In bytes
    'mime',                 # Mimetype of the book file being represented
))

CALIBRE_METADATA_FIELDS = frozenset((
    'application_id',   # An application id, currently set to the db_id.
    'db_id',            # the calibre primary key of the item.
    'formats',          # list of formats (extensions) for this book
    # a dict of user category names, where the value is a list of item names
    # from the book that are in that category
    'user_categories',
    # a dict of author to an associated hyperlink
    'author_link_map',
))

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(
```

(continues on next page)

(continuação da página anterior)

```
                            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 Tudo sobre utilização de expressões regulares no Calibre

Regular expressions are features used in many places in calibre to perform sophisticated manipulation of e-book content and metadata. This tutorial is a gentle introduction to getting you started with using regular expressions in calibre.

Conteúdos

- *Primeiro, uma palavra de advertência e encorajamento* (página 209)
- *Onde pode usar expressões regulares no Calibre?* (página 209)
- *O que afinal é uma expressão regular?* (página 209)
- *Importa-se de explicar?* (página 209)
- *Isso não parece tão mau assim. O que há a seguir?* (página 209)
- *Ah, claro! Isto está a começar a fazer sentido!* (página 210)
- *Well, these special characters are very neat and all, but what if I wanted to match a dot or a question mark?* (página 210)
- *Então, quais são os conjuntos mais úteis?* (página 211)
- *But if I had a few varying strings I wanted to match, things get complicated?* (página 211)
- *Você perdeu…* (página 212)
- *No início, você disse que havia uma maneira de fazer expressões regulares insensíveis?* (página 212)
- *Acho que estou começando a entender essas expressões regulares agora… como eu faço para usá-las no calibre?* (página 212)
  - *Conversões* (página 212)
  - *Adicionar livros* (página 213)
  - *Editar metadados em lote* (página 213)
- *Quick reference* (página 214)
- *Créditos* (página 218)

10.4.1 Primeiro, uma palavra de advertência e encorajamento

Isso, inevitavelmente, vai ser um pouco técnico, afinal, expressões regulares são uma ferramenta técnica para fazer coisas técnicas. Vou ter que usar alguns jargões e conceitos que podem parecer complicados ou enrolados. Vou tentar explicar esses conceitos o mais claro que eu puder, contudo eu realmente não poderei fazer isso sem usá-los. Dito isto, não desanime por qualquer jargão, eu tentarei explicar tudo novo. E embora expressões regulares em si podem parecer, um horror (ou, para ser mais prosaico, uma sequência aleatória de letras e sinais), eu prometo que elas não são tão complicadas. Mesmo aqueles que entendem expressões regulares muito bem terão dificuldades para ler as mais complexas, mas escrevê--las não é tão difícil, você constrói a expressão passo a passo. Então, dê um passo e siga-me.

10.4.2 Onde pode usar expressões regulares no Calibre?

There are a few places calibre uses regular expressions. There's the *Search & replace* in conversion options, metadata detection from filenames in the import settings and Search & replace when editing the metadata of books in bulk. The calibre book editor can also use regular expressions in its *Search and replace* feature. Finally, you can use regular expressions when searching the calibre book list and when searching inside the calibre E-book viewer.

10.4.3 O que afinal *é* uma expressão regular?

A regular expression is a way to describe sets of strings. A single regular expression can *match* a number of different strings. This is what makes regular expression so powerful – they are a concise way of describing a potentially large number of variations.

Nota: I'm using string here in the sense it is used in programming languages: a string of one or more characters, characters including actual characters, numbers, punctuation and so-called whitespace (linebreaks, tabulators etc.). Please note that generally, uppercase and lowercase characters are not considered the same, thus «a» being a different character from «A» and so forth. In calibre, regular expressions are case insensitive in the Search bar, but not in the conversion options. There's a way to make every regular expression case insensitive, but we'll discuss that later. It gets complicated because regular expressions allow for variations in the strings it matches, so one expression can match multiple strings, which is why people bother using them at all. More on that in a bit.

10.4.4 Importa-se de explicar?

Well, that's why we're here. First, this is the most important concept in regular expressions: *A string by itself is a regular expression that matches itself*. That is to say, if I wanted to match the string `"Hello, World!"` using a regular expression, the regular expression to use would be `Hello, World!`. And yes, it really is that simple. You'll notice, though, that this *only* matches the exact string `"Hello, World!"`, not e.g. `"Hello, wOrld!"` or `"hello, world!"` or any other such variation.

10.4.5 Isso não parece tão mau assim. O que há a seguir?

Next is the beginning of the really good stuff. Remember where I said that regular expressions can match multiple strings? This is where it gets a little more complicated. Say, as a somewhat more practical exercise, the e-book you wanted to convert had a nasty footer counting the pages, like «Page 5 of 423». Obviously the page number would rise from 1 to 423, thus you'd have to match 423 different strings, right? Wrong, actually: regular expressions allow you to define sets of characters that are matched: To define a set, you put all the characters you want to be in the set into square brackets. So, for example, the set `[abc]` would match either the character «a», «b» or «c». *Sets will always only match one of the characters in the set*. They «understand» character ranges, that is, if you wanted to match all the lower case characters, you'd use the set `[a-z]` for lower- and uppercase characters you'd use `[a-zA-Z]` and so on. Got the idea?

So, obviously, using the expression `Page [0-9] of 423` you'd be able to match the first 9 pages, thus reducing the expressions needed to three: The second expression `Page [0-9][0-9] of 423` would match all two-digit page numbers, and I'm sure you can guess what the third expression would look like. Yes, go ahead. Write it down.

10.4.6 Ah, claro! Isto está a começar a fazer sentido!

I was hoping you'd say that. But brace yourself, now it gets even better! We just saw that using sets, we could match one of several characters at once. But you can even repeat a character or set, reducing the number of expressions needed to handle the above page number example to one. Yes, ONE! Excited? You should be! It works like this: Some so-called special characters, «+», «?» and «*», *repeat the single element preceding them*. (Element means either a single character, a character set, an escape sequence or a group (we'll learn about those last two later)- in short, any single entity in a regular expression). These characters are called wildcards or quantifiers. To be more precise, «?» matches *0 or 1* of the preceding element, «*» matches *0 or more* of the preceding element and «+» matches *1 or more* of the preceding element. A few examples: The expression `a?` would match either «» (which is the empty string, not strictly useful in this case) or «a», the expression `a*` would match «», «a», «aa» or any number of a's in a row, and, finally, the expression `a+` would match «a», «aa» or any number of a's in a row (Note: it wouldn't match the empty string!). Same deal for sets: The expression `[0-9]+` would match *every integer number there is*! I know what you're thinking, and you're right: If you use that in the above case of matching page numbers, wouldn't that be the single one expression to match all the page numbers? Yes, the expression `Page [0-9]+ of 423` would match every page number in that book!

Nota: A note on these quantifiers: They generally try to match as much text as possible, so be careful when using them. This is called «greedy behaviour»- I'm sure you get why. It gets problematic when you, say, try to match a tag. Consider, for example, the string `"<p class="calibre2">Title here</p>"` and let's say you'd want to match the opening tag (the part between the first pair of angle brackets, a little more on tags later). You'd think that the expression `<p.*>` would match that tag, but actually, it matches the whole string! (The character «.» is another special character. It matches anything *except* linebreaks, so, basically, the expression `.*` would match any single line you can think of). Instead, try using `<p.*?>` which makes the quantifier `"*"` non-greedy. That expression would only match the first opening tag, as intended. There's actually another way to accomplish this: The expression `<p[^>]*>` will match that same opening tag- you'll see why after the next section. Just note that there quite frequently is more than one way to write a regular expression.

10.4.7 Well, these special characters are very neat and all, but what if I wanted to match a dot or a question mark?

You can of course do that: Just put a backslash in front of any special character and it is interpreted as the literal character, without any special meaning. This pair of a backslash followed by a single character is called an escape sequence, and the act of putting a backslash in front of a special character is called escaping that character. An escape sequence is interpreted as a single element. There are of course escape sequences that do more than just escaping special characters, for example `"\t"` means a tabulator. We'll get to some of the escape sequences later. Oh, and by the way, concerning those special characters: Consider any character we discuss in this introduction as having some function to be special and thus needing to be escaped if you want the literal character.

10.4.8 Então, quais são os conjuntos mais úteis?

Knew you'd ask. Some useful sets are `[0-9]` matching a single number, `[a-z]` matching a single lowercase letter, `[A-Z]` matching a single uppercase letter, `[a-zA-Z]` matching a single letter and `[a-zA-Z0-9]` matching a single letter or number. You can also use an escape sequence as shorthand:

\d
: é equivalente a `[0-9]`

\w
: é equivalente a `[a-zA-Z0-9_]`

\s
: é equivalente a um espaço em branco

Nota: «Whitespace» is a term for anything that won't be printed. These characters include space, tabulator, line feed, form feed, carriage return, non-breaking spaces, etc.

Nota: The upper and lower case sets may match both upper and lowercase if the setting to make searches case insensitive is enabled. Such settings are found, for instance in Preferences->Searching in calibre itself and on the Search panel in the calibre *E-book viewer* as well as the calibre *Edit book* tool.

As a last note on sets, you can also define a set as any character *but* those in the set. You do that by including the character `"^"` as the *very first character in the set*. Thus, `[^a]` would match any character excluding «a». That's called complementing the set. Those escape sequence shorthands we saw earlier can also be complemented: `"\D"` means any non-number character, thus being equivalent to `[^0-9]`. The other shorthands can be complemented by, you guessed it, using the respective uppercase letter instead of the lowercase one. So, going back to the example `<p[^>]*>` from the previous section, now you can see that the character set it's using tries to match any character except for a closing angle bracket.

10.4.9 But if I had a few varying strings I wanted to match, things get complicated?

Fear not, life still is good and easy. Consider this example: The book you're converting has «Title» written on every odd page and «Author» written on every even page. Looks great in print, right? But in e-books, it's annoying. You can group whole expressions in normal parentheses, and the character `"|"` will let you match *either* the expression to its right *or* the one to its left. Combine those and you're done. Too fast for you? Okay, first off, we group the expressions for odd and even pages, thus getting `(Title)(Author)` as our two needed expressions. Now we make things simpler by using the vertical bar (`"|"` is called the vertical bar character): If you use the expression `(Title|Author)` you'll either get a match for «Title» (on the odd pages) or you'd match «Author» (on the even pages). Well, wasn't that easy?

You can, of course, use the vertical bar without using grouping parentheses, as well. Remember when I said that quantifiers repeat the element preceding them? Well, the vertical bar works a little differently: The expression «Title|Author» will also match either the string «Title» or the string «Author», just as the above example using grouping. *The vertical bar selects between the entire expression preceding and following it.* So, if you wanted to match the strings «Calibre» and «calibre» and wanted to select only between the upper- and lowercase «c», you'd have to use the expression `(c|C)alibre`, where the grouping ensures that only the «c» will be selected. If you were to use `c|Calibre`, you'd get a match on the string «c» or on the string «Calibre», which isn't what we wanted. In short: If in doubt, use grouping together with the vertical bar.

10.4.10 Você perdeu...

… wait just a minute, there's one last, really neat thing you can do with groups. If you have a group that you previously matched, you can use references to that group later in the expression: Groups are numbered starting with 1, and you reference them by escaping the number of the group you want to reference, thus, the fifth group would be referenced as `\5`. So, if you searched for `([^ ]+) \1` in the string «Test Test», you'd match the whole string!

10.4.11 No início, você disse que havia uma maneira de fazer expressões regulares insensíveis?

Yes, I did, thanks for paying attention and reminding me. You can tell calibre how you want certain things handled by using something called flags. You include flags in your expression by using the special construct `(?flags go here)` where, obviously, you'd replace «flags go here» with the specific flags you want. For ignoring case, the flag is `i`, thus you include `(?i)` in your expression. Thus, `(?i)test` would match «Test», «tEst», «TEst» and any case variation you could think of.

Another useful flag lets the dot match any character at all, *including* the newline, the flag `s`. If you want to use multiple flags in an expression, just put them in the same statement: `(?is)` would ignore case and make the dot match all. It doesn't matter which flag you state first, `(?si)` would be equivalent to the above.

10.4.12 Acho que estou começando a entender essas expressões regulares agora... como eu faço para usá-las no calibre?

Conversões

Let's begin with the conversion settings, which is really neat. In the *Search & replace* part, you can input a regexp (short for regular expression) that describes the string that will be replaced during the conversion. The neat part is the wizard. Click on the wizard staff and you get a preview of what calibre «sees» during the conversion process. Scroll down to the string you want to remove, select and copy it, paste it into the regexp field on top of the window. If there are variable parts, like page numbers or so, use sets and quantifiers to cover those, and while you're at it, remember to escape special characters, if there are some. Hit the button labeled *Test* and calibre highlights the parts it would replace were you to use the regexp. Once you're satisfied, hit OK and convert. Be careful if your conversion source has tags like this example:

```
Maybe, but the cops feel like you do, Anita. What's one more dead vampire?
New laws don't change that. </p>
<p class="calibre4"> <b class="calibre2">Generated by ABC Amber LIT Conv
<a href="http://www.processtext.com/abclit.html" class="calibre3">erter,
http://www.processtext.com/abclit.html</a></b></p>
<p class="calibre4"> It had only been two years since Addison v. Clark.
The court case gave us a revised version of what life was
```

(shamelessly ripped out of this thread[90]). You'd have to remove some of the tags as well. In this example, I'd recommend beginning with the tag `<b class="calibre2">`, now you have to end with the corresponding closing tag (opening tags are `<tag>`, closing tags are `</tag>`), which is simply the next `</b>` in this case. (Refer to a good HTML manual or ask in the forum if you are unclear on this point). The opening tag can be described using `<b.*?>`, the closing tag using `</b>`, thus we could remove everything between those tags using `<b.*?>.*?</b>`. But using this expression would be a bad idea, because it removes everything enclosed by <b>- tags (which, by the way, render the enclosed text in bold print), and it's a fair bet that we'll remove portions of the book in this way. Instead, include the beginning of the enclosed string as well, making the regular expression `<b.*?>\s*Generated\s+by\s+ABC\s+Amber\s+LIT.*?</b>` The `\s` with quantifiers are included here instead of explicitly using the spaces as seen in the string to catch any variations of the string that might occur. Remember to check what calibre will remove to make sure you don't remove any portions

[90] https://www.mobileread.com/forums/showthread.php?t=75594"

you want to keep if you test a new expression. If you only check one occurrence, you might miss a mismatch somewhere else in the text. Also note that should you accidentally remove more or fewer tags than you actually wanted to, calibre tries to repair the damaged code after doing the removal.

Adicionar livros

Another thing you can use regular expressions for is to extract metadata from filenames. You can find this feature in the «Adding books» part of the settings. There's a special feature here: You can use field names for metadata fields, for example `(?P<title>)` would indicate that calibre uses this part of the string as book title. The allowed field names are listed in the windows, together with another nice test field. An example: Say you want to import a whole bunch of files named like `Classical Texts: The Divine Comedy by Dante Alighieri.mobi`. (Obviously, this is already in your library, since we all love classical italian poetry) or `Science Fiction epics: The Foundation Trilogy by Isaac Asimov.epub`. This is obviously a naming scheme that calibre won't extract any meaningful data out of - its standard expression for extracting metadata is `(?P<title>.+) - (?P<author>[^_]+)`. A regular expression that works here would be `[a-zA-Z]+: (?P<title>.+) by (?P<author>.+)`. Please note that, inside the group for the metadata field, you need to use expressions to describe what the field actually matches. And also note that, when using the test field calibre provides, you need to add the file extension to your testing filename, otherwise you won't get any matches at all, despite using a working expression.

Editar metadados em lote

The last part is regular expression *Search and replace* in metadata fields. You can access this by selecting multiple books in the library and using bulk metadata edit. Be very careful when using this last feature, as it can do **Very Bad Things** to your library! Doublecheck that your expressions do what you want them to using the test fields, and only mark the books you really want to change! In the regular expression search mode, you can search in one field, replace the text with something and even write the result into another field. A practical example: Say your library contained the books of Frank Herbert's Dune series, named after the fashion `Dune 1 - Dune`, `Dune 2 - Dune Messiah` and so on. Now you want to get `Dune` into the series field. You can do that by searching for `(.*?) \d+ - .*` in the title field and replacing it with `\1` in the series field. See what I did there? That's a reference to the first group you're replacing the series field with. Now that you have the series all set, you only need to do another search for `.*? -` in the title field and replace it with `""` (an empty string), again in the title field, and your metadata is all neat and tidy. Isn't that great? By the way, instead of replacing the entire field, you can also append or prepend to the field, so, if you *wanted* the book title to be prepended with series info, you could do that as well. As you by now have undoubtedly noticed, there's a checkbox labeled *Case sensitive*, so you won't have to use flags to select behaviour here.

Well, that just about concludes the very short introduction to regular expressions. Hopefully I'll have shown you enough to at least get you started and to enable you to continue learning by yourself- a good starting point would be the Python documentation for regexps[91].

One last word of warning, though: Regexps are powerful, but also really easy to get wrong. calibre provides really great testing possibilities to see if your expressions behave as you expect them to. Use them. Try not to shoot yourself in the foot. (God, I love that expression…). But should you, despite the warning, injure your foot (or any other body parts), try to learn from it.

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

10.4.13 Quick reference

Referência rápida para a sintaxe de expressões regulares

This checklist summarizes the most commonly used/hard to remember parts of the regexp engine available in most parts of calibre.

Conteúdos

- *Classes de caracteres* (página 214)
- *Shorthand character classes* (página 215)
- *Os quantificadores* (página 215)
- *Greed* (página 215)
- *Alternation* (página 215)
- *Exclusão* (página 216)
- *Âncoras* (página 216)
- *Grupos* (página 216)
- *Lookarounds* (página 217)
- *Recursion* (página 217)
- *Caracteres especiais* (página 218)
- *Meta-caracteres* (página 218)
- *Modos* (página 218)

Classes de caracteres

Character classes are useful to represent different groups of characters, succinctly.

Exemplos:

Representação	Classe
`[a-z]`	Letras minúsculas. Não inclui caracteres com acentos ou ligaturas
`[a-z0-9]`	Letras minúsculas de a a z ou números de 0 a 9
`[A-Za-z-]`	Uppercase or lowercase letters, or a dash. To include the dash in a class, you must put it at the beginning or at the end so as not to confuse it with the hyphen that specifies a range of characters
`[^0-9]`	Any character except a digit. The caret (^) placed at the beginning of the class excludes the characters of the class (complemented class)
`[[a-z]--[a`	The lowercase consonants. A class can be included in a class. The characters `--` exclude what follows them
`[\w--[\d_]]`	All letters (including foreign accented characters). Abbreviated classes can be used inside a class

Exemplo:

```
<[^<>]+> to select an HTML tag
```

Shorthand character classes

Representação	Classe
`\d`	Um dígito (o mesmo que `[0-9]`)
`\D`	Qualquer carácter não-alfanumérico (o mesmo que `[^0-9]`)
`\w`	Um carácter alfanumérico (`[a-zA-Z0-9]`) incluindo caracteres com acentos e ligaturas
`\W`	Qualquer carácter "não-palavra"
`\s`	Espaço, espaço impartível, tabulação, parágrafo de linha
`\S`	Any "non-whitespace" character
`.`	Any character except newline. Use the "dot all" checkbox or the `(?s)` regexp modifier to include the newline character.

Os quantificadores

Quantificador	Number of occurrences of the expression preceding the quantifier
`?`	0 ou 1 ocorrência da expressão. O mesmo que `{0,1}`
`+`	1 ou mais ocorrências da expressão. O mesmo que `{1,}`
`*`	0, 1 ou mais ocorrências da expressão. O mesmo que `{0,}`
`{n}`	Exatamente N ocorrências da expressão
`{min,max}`	Number of occurrences between the minimum and maximum values included
`{min,}`	Number of occurrences between the minimum value included and the infinite
`{,max}`	Number of occurrences between 0 and the maximum value included

Greed

By default, with quantifiers, the regular expression engine is greedy: it extends the selection as much as possible. This often causes surprises, at first. `?` follows a quantifier to make it lazy. Avoid putting two in the same expression, the result can be unpredictable.

Beware of nesting quantifiers, for example, the pattern `(a*)*`, as it exponentially increases processing time.

Alternation

The `|` character in a regular expression is a logical `OR`. It means that either the preceding or the following expression can match.

Exclusão

Método 1

```
pattern_to_exclude(*SKIP)(*FAIL)|pattern_to_select
```

Exemplo:

```
"Blabla"(*SKIP)(*FAIL)|Blabla
```

seleciona Blabla, nas entradas Blabla ou «Blabla ou Blabla», mas não em «Blabla».

Método 2

```
pattern_to_exclude\K|(pattern_to_select)
```

```
"Blabla"\K|(Blabla)
```

seleciona Blabla, nas entradas Blabla ou «Blabla ou Blabla», mas não em «Blabla».

Âncoras

An anchor is a way to match a logical location in a string, rather than a character. The most useful anchors for text processing are:

\b
: Designates a word boundary, i.e. a transition from space to non-space character. For example, you can use `\bsurd` to match `the surd` but not `absurd`.

^
: Matches the start of a line (in multi-line mode, which is the default)

$
: Matches the end of a line (in multi-line mode, which is the default)

\K
: Resets the start position of the selection to its position in the pattern. Some regexp engines (but not calibre) do not allow lookbehind of variable length, especially with quantifiers. When you can use `\K` with these engines, it also allows you to get rid of this limit by writing the equivalent of a positive lookbehind of variable length.

Grupos

(expressão)
: Capturing group, which stores the selection and can be recalled later in the *search* or *replace* patterns with `\n`, where `n` is the sequence number of the capturing group (starting at 1 in reading order)

(?:expression)
: Group that does not capture the selection

(?>expression)
: Atomic Group: As soon as the expression is satisfied, the regexp engine passes, and if the rest of the pattern fails, it will not backtrack to try other combinations with the expression. Atomic groups do not capture.

(?|expression)
: Branch reset group: the branches of the alternations included in the expression share the same group numbers

(?<name>expressão)
Group named "name". The selection can be recalled later in the *search* pattern by `(?P=name)` and in the *replace* by `\g<name>`. Two different groups can use the same name.

Lookarounds

Lookaround	Meaning
`?=`	Positive lookahead (to be placed after the selection)
`?!`	Negative lookahead (to be placed after the selection)
`?<=`	Positive lookbehind (to be placed before the selection)
`?<!`	Negative lookbehind (to be placed before the selection)

Lookaheads and lookbehinds do not consume characters, they are zero length and do not capture. They are atomic groups: as soon as the assertion is satisfied, the regexp engine passes, and if the rest of the pattern fails, it will not backtrack inside the lookaround to try other combinations.

When looking for multiple matches in a string, at the starting position of each match attempt, a lookbehind can inspect the characters before the current position. Therefore, on the string 123, the pattern `(?<=\d)\d` (a digit preceded by a digit) should, in theory, select 2 and 3. On the other hand, `\d\K\d` can only select 2, because the starting position after the first selection is immediately before 3, and there are not enough digits for a second match. Similarly, `\d(\d)` only captures 2. In calibre's regexp engine practice, the positive lookbehind behaves in the same way, and selects only 2, contrary to theory.

Groups can be placed inside lookarounds, but capture is rarely useful. Nevertheless, if it is useful, it will be necessary to be very careful in the use of a quantifier in a lookbehind: the greed associated with the absence of backtracking can give a surprising capture. For this reason, use `\K` rather than a positive lookbehind when you have a quantifier (or worse, several) in a capturing group of the positive lookbehind.

Example of negative lookahead:

```
(?![^<>{}]*[>}])
```

Placed at the end of the pattern prevents to select within a tag or a style embedded in the file.

Whenever possible, it is always better to «anchor» the lookarounds, to reduce the number of steps necessary to obtain the result.

Recursion

Representação	Meaning
`(?R)`	Recursion of the entire pattern
`(?1)`	Recursion of the only pattern of the numbered capturing group, here group 1

Recursion is calling oneself. This is useful for balanced queries, such as quoted strings, which can contain embedded quoted strings. Thus, if during the processing of a string between double quotation marks, we encounter the beginning of a new string between double quotation marks, well we know how to do, and we call ourselves. Then we have a pattern like:

```
start-pattern(?>atomic sub-pattern|(?R))*end-pattern
```

To select a string between double quotation marks without stopping on an embedded string:

```
“((?>[^“”]+|(?R))*[^“”]+)”
```

This template can also be used to modify pairs of tags that can be embedded, such as `<div>` tags.

Caracteres especiais

Representação	Carácter
`\t`	tabulação
`\n`	quebra de linha
`\x20`	espaço quebrável
`\xa0`	espaço não-quebrável

Meta-caracteres

Meta-characters are those that have a special meaning for the regexp engine. Of these, twelve must be preceded by an escape character, the backslash (`\`), to lose their special meaning and become a regular character again:

```
^ . [ ] $ ( ) * + ? | \
```

Seven other meta-characters do not need to be preceded by a backslash (but can be without any other consequence):

```
{ } ! < > = :
```

Special characters lose their status if they are used inside a class (between brackets `[]`). The closing bracket and the dash have a special status in a class. Outside the class, the dash is a simple literal, the closing bracket remains a meta-character.

The slash (/) and the number sign (or hash character) (#) are not meta-characters, they don't need to be escaped.

In some tools, like regex101.com with the Python engine, double quotes have the special status of separator, and must be escaped, or the options changed. This is not the case in the editor of calibre.

Modos

(?s)
: Causes the dot (`.`) to match newline characters as well

(?m)
: Makes the `^` and `$` anchors match the start and end of lines instead of the start and end of the entire string.

10.4.14 Créditos

Obrigado por ajudar com dicas, correções e tal:

- ldolse
- kovidgoyal
- chaley
- dwanthny

- kacir
- Starson17
- Orpheu

For more about regexps see The Python User Manual[92]. The actual regular expression library used by calibre is: regex[93] which supports several useful enhancements over the Python standard library one.

10.5 Criar os seus próprios plugins para aumentar as funcionalidades do Calibre

O Calibre possui um design muito modular. Quase todas as funcionalidades do Calibre existem na forma de plugins. Os plugins são usados para conversão, para descarregar notícias (embora sejam chamadas de receitas), para vários componentes da interface de utilizador, para se ligar a dispositivos diferentes, para processar ficheiros ao adicioná-los ao Calibre e assim por diante. Pode obter uma lista completa de todos os plugins embutidos no Calibre, aceda a *Preferências* → *Avançado* → *Plugins*.

Aqui vamos mostrar como criar seus próprios plugins e adicionar novas funcionalidades ao Calibre

Conteúdos

- *Anatomia de um plugin Calibre* (página 220)
- *Um plugin da Interface do utilizador* (página 221)
  - *__init__.py* (página 222)
  - *ui.py* (página 223)
  - *main.py* (página 224)
  - *Obtendo recursos do ficheiro ZIP do plugin* (página 227)
  - *Ativar a configuração de utilizador para o seu plugin* (página 227)
- *Editar plugins de livro* (página 229)
  - *main.py* (página 230)
- *Adicionar traduções ao seu plugin* (página 232)
- *A API de plugin* (página 233)
- *Depurar plugins* (página 233)
- *Mais exemplos de plugins* (página 234)
- *Partilhar os seus plugins com outros* (página 234)

Nota: Isto só se aplica a versões Calibre >= 0.8.60

[92] https://docs.python.org/library/re.html
[93] https://bitbucket.org/mrabarnett/mrab-regex/src/hg/

10.5.1 Anatomia de um plugin Calibre

Um plugin de calibre é muito simples, é apenas um arquivo ZIP que contém algum código Python e outros recursos, como arquivos de imagem necessários ao plugin. Sem mais delongas, vamos ver um exemplo básico.

Suponha que você tenha uma instalação de calibre que esteja usando para publicar automaticamente vários documentos eletrônicos nos formatos EPUB e MOBI. Você deseja que todos os arquivos gerados pelo calibre tenham o editor definido como «Olá, mundo», veja como fazê-lo. Crie um arquivo chamado: file: *__init __. Py* (este é um nome especial e sempre deve ser usado para o arquivo principal do seu plugin) e insira o seguinte código Python nele:

```
from calibre.customize import FileTypePlugin

class HelloWorld(FileTypePlugin):

    name                = 'Hello World Plugin' # Name of the plugin
    description         = 'Set the publisher to Hello World for all new conversions'
    supported_platforms = ['windows', 'osx', 'linux'] # Platforms this plugin will␣
↪run on
    author              = 'Acme Inc.' # The author of this plugin
    version             = (1, 0, 0)   # The version number of this plugin
    file_types          = {'epub', 'mobi'} # The file types that this plugin will be␣
↪applied to
    on_postprocess      = True # Run this plugin after conversion is complete
    minimum_calibre_version = (0, 7, 53)

    def run(self, path_to_ebook):
        from calibre.ebooks.metadata.meta import get_metadata, set_metadata
        with open(path_to_ebook, 'r+b') as file:
            ext  = os.path.splitext(path_to_ebook)[-1][1:].lower()
            mi = get_metadata(file, ext)
            mi.publisher = 'Hello World'
            set_metadata(file, mi, ext)
        return path_to_ebook
```

That's all. To add this code to calibre as a plugin, simply run the following in the folder in which you created `__init__.py`:

```
calibre-customize -b .
```

Nota: No macOS, as ferramentas de linha de comando estão dentro do pacote do Calibre. Por exemplo, se instalou o Calibre em `/Applications` as ferramentas de linha de comando estão em `/Applications/calibre.app/Contents/MacOS/`.

You can download the Hello World plugin from helloworld_plugin.zip[94].

Toda vez que você usar o calibre para converter um livro, o método plugin: meth: *run* será chamado e o livro convertido terá seu editor definido como» Hello World «. Este é um plugin trivial, vamos para um exemplo mais complexo que realmente adiciona um componente à interface do usuário.

[94] https://calibre-ebook.com/downloads/helloworld_plugin.zip

10.5.2 Um plugin da Interface do utilizador

Este plugin será espalhado por alguns arquivos (para manter o código limpo). Ele mostrará como obter recursos (imagens ou arquivos de dados) do arquivo ZIP do plug-in, permitir que os usuários configurem seu plug-in, como criar elementos na interface do usuário do calibre e como acessar e consultar o banco de dados de livros no calibre.

You can download this plugin from interface_demo_plugin.zip[95]

A primeira coisa a observar é que esse arquivo ZIP contém muito mais arquivos, explicados abaixo, e preste atenção `plugin-import-name-interface_demo.txt`.

plugin-import-name-interface_demo.txt
: Um ficheiro de texto vazio usado para permitir a magia do plugin de multi-ficheiros. O ficheiro tem de estar presente em todas os plugins que usam mais do que um ficheiro .py. Deve estar vazio e o seu nome tem de estar na forma: `plugin-import-name-**some_name**.txt`. A presença deste ficheiro permite-lhe importar código dos ficheiros .py presentes dentro do ficheiro ZIP, usando uma declaração como no exemplo:

```
from calibre_plugins.some_name.some_module import some_object
```

The prefix `calibre_plugins` must always be present. `some_name` comes from the filename of the empty text file. `some_module` refers to `some_module.py` file inside the ZIP file. Note that this importing is just as powerful as regular Python imports. You can create packages and subpackages of .py modules inside the ZIP file, just like you would normally (by defining __init__.py in each sub--folder), and everything should «just work».

O nome que você usa para `` some_name`` entra em um namespace global compartilhado por todos os plugins, ** para torná-lo o mais exclusivo possível **. Mas lembre-se de que deve ser um identificador Python válido (apenas alfabetos, números e sublinhado)

__init__.py
: Tal como anteriormente, o ficheiro que define a classe do plugin

main.py
: Este arquivo contém o código real que faz algo útil

ui.py
: Esse arquivo define a parte de interface do plugin

images/icon.png
: O ícone para este plugin

about.txt
: Um ficheiro de texto com a informação sobre o plugin

traduções
: Uma pasta contendo arquivos `.mo` com as traduções da interface do usuário do seu plugin para diferentes idiomas. Veja abaixo os detalhes.

Agora vamos ver o código

[95] https://calibre-ebook.com/downloads/interface_demo_plugin.zip

__init__.py

Primeiro, o obrigatório `__init__.py` para definir os metadados do plugin:

```
from calibre.customize import InterfaceActionBase

class InterfacePluginDemo(InterfaceActionBase):
    '''
    This class is a simple wrapper that provides information about the actual
    plugin class. The actual interface plugin class is called InterfacePlugin
    and is defined in the ui.py file, as specified in the actual_plugin field
    below.

    The reason for having two classes is that it allows the command line
    calibre utilities to run without needing to load the GUI libraries.
    '''
    name                = 'Interface Plugin Demo'
    description         = 'An advanced plugin demo'
    supported_platforms = ['windows', 'osx', 'linux']
    author              = 'Kovid Goyal'
    version             = (1, 0, 0)
    minimum_calibre_version = (0, 7, 53)

    #: This field defines the GUI plugin class that contains all the code
    #: that actually does something. Its format is module_path:class_name
    #: The specified class must be defined in the specified module.
    actual_plugin       = 'calibre_plugins.interface_demo.ui:InterfacePlugin'

    def is_customizable(self):
        '''
        This method must return True to enable customization via
        Preferences->Plugins
        '''
        return True

    def config_widget(self):
        '''
        Implement this method and :meth:`save_settings` in your plugin to
        use a custom configuration dialog.

        This method, if implemented, must return a QWidget. The widget can have
        an optional method validate() that takes no arguments and is called
        immediately after the user clicks OK. Changes are applied if and only
        if the method returns True.

        If for some reason you cannot perform the configuration at this time,
        return a tuple of two strings (message, details), these will be
        displayed as a warning dialog to the user and the process will be
        aborted.

        The base class implementation of this method raises NotImplementedError
        so by default no user configuration is possible.
        '''
        # 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
```

(continues on next page)

(continuação da página anterior)

```
        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()
```

O único recurso digno de nota é o campo `actual_plugin`. Como o *Calibre* tem linha de comando e interfaces *GUI*, plug-ins *GUI* como este não devem carregar nenhuma biblioteca *GUI* em `__init__.py`. O campo `actual_plugin` faz isso para você, informando ao *Calibre* que o plugin real deve ser encontrado em outro arquivo dentro do seu arquivo ZIP, que só será carregado em um contexto GUI.

Lembre-se de que, para que isso funcione, você deve ter um arquivo `plugin-import-name-some_name.txt` no arquivo ZIP do plugin, conforme discutido acima.

Também existem alguns métodos para habilitar a configuração do plugin pelo usuário. Eles são discutidos abaixo.

ui.py

Agora vamos dar uma olhada em `ui.py` que define o plugin GUI real. O código-fonte é muito comentado e deve ser autoexplicativo:

```
from calibre.gui2.actions import InterfaceAction
from calibre_plugins.interface_demo.main import DemoDialog


class InterfacePlugin(InterfaceAction):

    name = 'Interface Plugin Demo'

    # Declare the main action associated with this plugin
    # The keyboard shortcut can be None if you dont want to use a keyboard
    # shortcut. Remember that currently calibre has no central management for
    # keyboard shortcuts, so try to use an unusual/unused shortcut.
    action_spec = ('Interface Plugin Demo', None,
            '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.
```

(continues on next page)

(continuação da página anterior)

```
        #
        # 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

A lógica real para implementar a caixa de diálogo das *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
```

(continues on next page)

(continuação da página anterior)

```
        self.l = QVBoxLayout()
        self.setLayout(self.l)

        self.label = QLabel(prefs['hello_world_msg'])
        self.l.addWidget(self.label)

        self.setWindowTitle('Interface Plugin Demo')
        self.setWindowIcon(icon)

        self.about_button = QPushButton('About', self)
        self.about_button.clicked.connect(self.about)
        self.l.addWidget(self.about_button)

        self.marked_button = QPushButton(
            'Show books with only one format in the calibre GUI', self)
        self.marked_button.clicked.connect(self.marked)
        self.l.addWidget(self.marked_button)

        self.view_button = QPushButton(
            'View the most recently added book', self)
        self.view_button.clicked.connect(self.view)
        self.l.addWidget(self.view_button)

        self.update_metadata_button = QPushButton(
            'Update metadata in a book\'s files', self)
        self.update_metadata_button.clicked.connect(self.update_metadata)
        self.l.addWidget(self.update_metadata_button)

        self.conf_button = QPushButton(
                'Configure this plugin', self)
        self.conf_button.clicked.connect(self.config)
        self.l.addWidget(self.conf_button)

        self.resize(self.sizeHint())

    def about(self):
        # Get the about text from a file inside the plugin zip file
        # The get_resources function is a builtin function defined for all your
        # plugin code. It loads files from the plugin zip file. It returns
        # the bytes from the specified file.
        #
        # Note that if you are loading more than one file, for performance, you
        # should pass a list of names to get_resources. In this case,
        # get_resources will return a dictionary mapping names to bytes. Names that
        # are not found in the zip file will not be in the returned dictionary.
        text = get_resources('about.txt')
        QMessageBox.about(self, 'About the Interface Plugin Demo',
                text.decode('utf-8'))

    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
```

(continues on next page)

(continuação da página anterior)

```
        # db object
        self.db.set_marked_ids(matched_ids)

        # Tell the GUI to search for all marked records
        self.gui.search.setEditText('marked:true')
        self.gui.search.do_search()

    def view(self):
        ''' View the most recently added book '''
        most_recent = most_recent_id = None
        db = self.db.new_api
        for book_id, timestamp in db.all_field_for('timestamp', db.all_book_ids()).
↪items():
            if most_recent is None or timestamp > most_recent:
                most_recent = timestamp
                most_recent_id = book_id

        if most_recent_id is not None:
            # Get a reference to the View plugin
            view_plugin = self.gui.iactions['View']
            # Ask the view plugin to launch the viewer for row_number
            view_plugin._view_calibre_books([most_recent_id])

    def update_metadata(self):
        '''
        Set the metadata in the files in the selected book's record to
        match the current metadata in the database.
        '''
        from calibre.ebooks.metadata.meta import set_metadata
        from calibre.gui2 import error_dialog, info_dialog

        # Get currently selected books
        rows = self.gui.library_view.selectionModel().selectedRows()
        if not rows or len(rows) == 0:
            return error_dialog(self.gui, 'Cannot update metadata',
                                'No books selected', show=True)
        # Map the rows to book ids
        ids = list(map(self.gui.library_view.model().id, rows))
        db = self.db.new_api
        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 dont use add_format_with_hooks as the hooks were
```

(continues on next page)

(continuação da página anterior)

```
                # already run when the file was first added to calibre.
                db.add_format(book_id, fmt, ffile, run_hooks=False)

        info_dialog(self, 'Updated files',
                'Updated the metadata in the files of %d book(s)'%len(ids),
                show=True)

    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

Obtendo recursos do ficheiro ZIP do plugin

o sistema de carregamento do plugin do calibre define algumas funções built-in que permitem obter convenientemente os arquivos do arquivo ZIP do plugin.

get_resources(name_or_list_of_names)
: This function should be called with a list of paths to files inside the ZIP file. For example to access the file `icon.png` in the folder images in the ZIP file, you would use: `images/icon.png`. Always use a forward slash as the path separator, even on Windows. When you pass in a single name, the function will return the raw bytes of that file or None if the name was not found in the ZIP file. If you pass in more than one name then it returns a dictionary mapping the names to bytes. If a name is not found, it will not be present in the returned dictionary.

get_icons(name_or_list_of_names, plugin_name="")
: A wrapper for get_resources() that creates QIcon objects from the raw bytes returned by get_resources. If a name is not found in the ZIP file the corresponding QIcon will be null. In order to support icon theme-ing, pass in the human friendly name of your plugin as `plugin_name`. If the user is using an icon theme with icons for your plugin, they will be loaded preferentially.

Ativar a configuração de utilizador para o seu plugin

Para permitir que os usuários configurem seu plug-in, você deve definir três métodos em sua classe base de plug-in, **is_customizable()**, **config_widget()** e **save_settings()** conforme mostrado abaixo:

```
    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.

```

(continues on next page)

(continuação da página anterior)

```
        If for some reason you cannot perform the configuration at this time,
        return a tuple of two strings (message, details), these will be
        displayed as a warning dialog to the user and the process will be
        aborted.

        The base class implementation of this method raises NotImplementedError
        so by default no user configuration is possible.
        '''
        # It is important to put this import statement here rather than at the
        # top of the module as importing the config class will also cause the
        # GUI libraries to be loaded, which we do not want when using calibre
        # from the command line
        from calibre_plugins.interface_demo.config import ConfigWidget
        return ConfigWidget()
```

```
    def save_settings(self, config_widget):
        '''
        Save the settings specified by the user with config_widget.

        :param config_widget: The widget returned by :meth:`config_widget`.
        '''
        config_widget.save_settings()

        # Apply the changes
        ac = self.actual_plugin_
        if ac is not None:
            ac.apply_settings()
```

calibre tem muitas maneiras diferentes de armazenar dados de configuração (um legado de sua longa história). A maneira recomendada é usar a classe **JSONConfig**, que armazena suas informações de configuração em um arquivo `.json`.

O código para gerenciar os dados de configuração no plug-in demo está em `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 dont accidentally clobber a calibre config file
prefs = JSONConfig('plugins/interface_demo')

# Set defaults
prefs.defaults['hello_world_msg'] = 'Hello, World!'


class ConfigWidget(QWidget):

    def __init__(self):
        QWidget.__init__(self)
        self.l = QHBoxLayout()
        self.setLayout(self.l)

        self.label = QLabel('Hello world &message:')
        self.l.addWidget(self.label)
```

(continues on next page)

(continuação da página anterior)

```
        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()
```

O objeto `prefs` agora está disponível em todo o código do plugin por um simples:

```
from calibre_plugins.interface_demo.config import prefs
```

É possível ver o objeto `prefs` sendo usado em `main.py`:

```
    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

10.5.3 Editar plugins de livro

Now let's change gears for a bit and look at creating a plugin to add tools to the calibre book editor. The plugin is available here: editor_demo_plugin.zip[96].

The first step, as for all plugins is to create the import name empty txt file, as described *above* (página 221). We shall name the file `plugin-import-name-editor_plugin_demo.txt`.

Now we create the mandatory `__init__.py` file that contains metadata about the plugin – its name, author, version, etc.

```
from calibre.customize import EditBookToolPlugin


class DemoPlugin(EditBookToolPlugin):

    name = 'Edit Book plugin demo'
    version = (1, 0, 0)
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'osx', 'linux']
    description = 'A demonstration of the plugin interface for the ebook editor'
    minimum_calibre_version = (1, 46, 0)
```

A single editor plugin can provide multiple tools each tool corresponds to a single button in the toolbar and entry in the *Plugins* menu in the editor. These can have sub-menus in case the tool has multiple related actions.

The tools must all be defined in the file `main.py` in your plugin. Every tool is a class that inherits from the *`calibre.gui2.tweak_book.plugin.Tool`* (página 360) class. Let's look at `main.py` from the demo plugin, the source code is heavily commented and should be self-explanatory. Read the API documents of the *`calibre.gui2.tweak_book.plugin.Tool`* (página 360) class for more details.

[96] https://calibre-ebook.com/downloads/editor_demo_plugin.zip

main.py

Here we will see the definition of a single tool that will multiply all font sizes in the book by a number provided by the user. This tool demonstrates various important concepts that you will need in developing your own plugins, so you should read the (heavily commented) source code carefully.

```
import re
from qt.core import QAction, QInputDialog
from css_parser.css import CSSRule

# The base class that all tools must inherit from
from calibre.gui2.tweak_book.plugin import Tool

from calibre import force_unicode
from calibre.gui2 import error_dialog
from calibre.ebooks.oeb.polish.container import OEB_DOCS, OEB_STYLES, serialize


class DemoTool(Tool):

    #: Set this to a unique name it will be used as a key
    name = 'demo-tool'

    #: If True the user can choose to place this tool in the plugins toolbar
    allowed_in_toolbar = True

    #: If True the user can choose to place this tool in the plugins menu
    allowed_in_menu = True

    def create_action(self, for_toolbar=True):
        # Create an action, this will be added to the plugins toolbar and
        # the plugins menu
        ac = QAction(get_icons('images/icon.png'), 'Magnify fonts', self.gui)  # noqa
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
            # register it for the action created for the menu, not the toolbar,
            # to avoid a double trigger
            self.register_shortcut(ac, 'magnify-fonts-tool', default_keys=(
↪'Ctrl+Shift+Alt+D',))
        ac.triggered.connect(self.ask_user)
        return ac

    def ask_user(self):
        # Ask the user for a factor by which to multiply all font sizes
        factor, ok = QInputDialog.getDouble(
            self.gui, 'Enter a magnification factor', 'Allow font sizes in the book␣
↪will be multiplied by the specified factor',
            value=2, min=0.1, max=4
        )
        if ok:
            # Ensure any in progress editing the user is doing is present in the␣
↪container
            self.boss.commit_all_editors_to_container()
            try:
                self.magnify_fonts(factor)
            except Exception:
                # Something bad happened report the error to the user
```

(continues on next page)

(continuação da página anterior)

```
                import traceback
                error_dialog(self.gui, _('Failed to magnify fonts'), _(
                    'Failed to magnify fonts, click "Show details" for more info'),
                    det_msg=traceback.format_exc(), show=True)
                # Revert to the saved restore point
                self.boss.revert_requested(self.boss.global_undo.previous_container)
            else:
                # Show the user what changes we have made, allowing her to
                # revert them if necessary
                self.boss.show_current_diff()
                # Update the editor UI to take into account all the changes we
                # have made
                self.boss.apply_container_update_to_gui()

    def magnify_fonts(self, factor):
        # Magnify all font sizes defined in the book by the specified factor
        # First we create a restore point so that the user can undo all changes
        # we make.
        self.boss.add_savepoint('Before: Magnify fonts')

        container = self.current_container  # The book being edited as a container␣
↪object

        # Iterate over all style declarations in the book, this means css
        # stylesheets, <style> tags and style="" attributes
        for name, media_type in container.mime_map.items():
            if media_type in OEB_STYLES:
                # A stylesheet. Parsed stylesheets are css_parser CSSStylesheet
                # objects.
                self.magnify_stylesheet(container.parsed(name), factor)
                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):
```

(continues on next page)

(continuação da página anterior)

```
        # Magnify all fonts in the specified stylesheet by the specified
        # factor.
        for rule in sheet.cssRules.rulesOfType(CSSRule.STYLE_RULE):
            self.magnify_declaration(rule.style, factor)

    def magnify_declaration(self, style, factor):
        # Magnify all fonts in the specified style declaration by the specified
        # factor
        val = style.getPropertyValue('font-size')
        if not val:
            return
        # see if the font-size contains a number
        num = re.search(r'[0-9.]+', val)
        if num is not None:
            num = num.group()
            val = val.replace(num, '%f' % (float(num) * factor))
            style.setProperty('font-size', val)
        # We should also be dealing with the font shorthand property and
        # font sizes specified as non numbers, but those are left as exercises
        # for the reader
```

Let's break down `main.py`. We see that it defines a single tool, named *Magnify fonts*. This tool will ask the user for a number and multiply all font sizes in the book by that number.

The first important thing is the tool name which you must set to some relatively unique string as it will be used as the key for this tool.

The next important entry point is the *calibre.gui2.tweak_book.plugin.Tool.create_action()* (página 361). This method creates the QAction objects that appear in the plugins toolbar and plugin menu. It also, optionally, assigns a keyboard shortcut that the user can customize. The triggered signal from the QAction is connected to the ask_user() method that asks the user for the font size multiplier, and then runs the magnification code.

The magnification code is well commented and fairly simple. The main things to note are that you get a reference to the editor window as `self.gui` and the editor *Boss* as `self.boss`. The *Boss* is the object that controls the editor user interface. It has many useful methods, that are documented in the *calibre.gui2.tweak_book.boss.Boss* (página 362) class.

Finally, there is `self.current_container` which is a reference to the book being edited as a *calibre.ebooks.oeb.polish.container.Container* (página 353) object. This represents the book as a collection of its constituent HTML/CSS/image files and has convenience methods for doing many useful things. The container object and various useful utility functions that can be reused in your plugin code are documented in *Documentação da API para as ferramentas de edição de ebooks* (página 352).

10.5.4 Adicionar traduções ao seu plugin

You can have all the user interface strings in your plugin translated and displayed in whatever language is set for the main calibre user interface.

The first step is to go through your plugin's source code and mark all user visible strings as translatable, by surrounding them in _(). For example:

```
action_spec = (_('My plugin'), None, _('My plugin is cool'), None)
```

Then use some program to generate .po files from your plugin source code. There should be one .po file for every language you want to translate into. For example: de.po for German, fr.po for French and so on. You can use the Poedit[97] program

[97] https://poedit.net/

for this.

Send these .po files to your translators. Once you get them back, compile them into .mo files. You can again use Poedit for that, or just do:

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

Put the .mo files into the `translations` folder in your plugin.

The last step is to simply call the function *load_translations()* at the top of your plugin's .py files. For performance reasons you should only call this function in those .py files that actually have translatable strings. So in a typical User Interface plugin you would call it at the top of `ui.py` but not `__init__.py`.

You can test the translations of your plugins by changing the user interface language in calibre under *Preferences → Interface → Look & feel* or by running calibre with the `CALIBRE_OVERRIDE_LANG` environment variable set. For example:

```
CALIBRE_OVERRIDE_LANG=de
```

Replace `de` with the language code of the language you want to test.

10.5.5 A API de plugin

As you may have noticed above, a plugin in calibre is a class. There are different classes for the different types of plugins in calibre. Details on each class, including the base class of all plugins can be found in *Documentação de API para plug-ins* (página 250).

Your plugin is almost certainly going to use code from calibre. To learn how to find various bits of functionality in the calibre code base, read the section on the calibre *Visual de código* (página 338).

10.5.6 Depurar plugins

A primeira e mais importante etapa é executar o Calibre em modo de depuração. Pode fazer isso na linha de comando com:

```
calibre-debug -g
```

Or from within calibre by right-clicking the *Preferences* button or using the `Ctrl+Shift+R` keyboard shortcut.

Ao executar a partir da linha de comando, a saída de depuração será impressa na consola; ao executar a partir do *Calibre*, a saída irá para um ficheiro``.txt``.

É possível inserir declarações `print()` em qualquer lugar do código do plugin, elas serão geradas no modo de depuração. Lembre-se, isso é *Python*, você realmente não precisa de nada além de declarações `print()` para depurar ;) Desenvolvi todo o *Calibre* usando apenas esta técnica de depuração.

Você pode testar rapidamente as alterações em seu plug-in usando a seguinte linha de comando:

```
calibre-debug -s; calibre-customize -b /path/to/your/plugin/folder; calibre
```

Isso desligará o calibre em execução, esperará até que o desligamento seja concluído e, em seguida, atualizará o calibre do plug-in e o reiniciará.

10.5.7 Mais exemplos de plugins

You can find a list of many sophisticated calibre plugins here[98].

10.5.8 Partilhar os seus plugins com outros

Se você gostaria de compartilhar os plug-ins que criou com outros usuários do calibre, poste seu plug-in em um novo tópico no fórum de plug-ins do calibre[99].

10.6 Integrar fórmulas matemáticas em livros digitais

O visualizador de livros digitais Calibre tem a capacidade de reproduzir formulas que estejam presentes em livros digitais (ficheiros EPUB e HTML). É possível inserir formulas diretamente utilizando programas como TeX, MathML ou AsciiMath. O visualizador de livros digitais Calibre usa o afamado MathJax[100] library para este fim. Isto é um tutorial rápido de como criar livros digitais que contenham formulas matemáticas para que funcionem bem com o visualizador de livros digitais Calibre.

10.6.1 Um simples ficheiro HTML com formulas matemáticas

Pode escrever matemática embebida num ficheiro HTML simples e o visualizador de ebooks do Calibre fará a respetiva renderização adequadamente composta como matemática. No exemplo abaixo, usaremos a notação do TeX para conteúdo matemático. Verá que pode usar comandos TeX normais, com a pequena particularidade de que os caracteres E comercial, menor que e maior que, têm de ser escritos como & < e > respetivamente.

O primeiro passo é dizer ao Calibre que conterá conteúdo matemático. Isto é conseguido ao adicionar o trecho de código seguinte à secção <head> do ficheiro HTML:

```
<script type="text/x-mathjax-config"></script>
```

Isto é, agora pode compor texto matemático tal como o faria num ficheiro .tex. Por exemplo, aqui estão as equações de Lorentz:

```
<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>
```

Este trecho assemelha-se à seguinte captura de ecrã no visualizador do Calibre.

O ficheiro HTML completo, com mais equações e texto matemático embebido, é reproduzido abaixo. Pode converter este ficheiro HTML para EPUB no Calibre obtendo como resultado um ebook que pode facilmente distribuir a outras pessoas.

[98] https://www.mobileread.com/forums/showthread.php?t=118764
[99] https://www.mobileread.com/forums/forumdisplay.php?f=237
[100] https://www.mathjax.org

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Figura1: *Equações Lorenz*

```
<!DOCTYPE html>
<html>
<!-- Copyright (c) 2012 Design Science, Inc. -->
<head>
<title>Math Test Page</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<!-- This script tag is needed to make calibre's ebook-viewer recpgnize that this␣
↪file needs math typesetting -->
<script type="text/x-mathjax-config">
    // This line adds numbers to all equations automatically, unless explicitly␣
↪suppressed.
    MathJax.tex = {tags: 'all'};
</script>

<style>
h1 {text-align:center}
h2 {
  font-weight: bold;
  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>\[
```

(continues on next page)

(continuação da página anterior)

```
\left( \sum_{k=1}^n a_k b_k \right)^{\!\!2} \leq
 \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
\]</p>

<h2>A Cross Product Formula</h2>

<p>\[
  \mathbf{V}_1 \times \mathbf{V}_2 =
   \begin{vmatrix}
    \mathbf{i} &amp; \mathbf{j} &amp; \mathbf{k} \\
    \frac{\partial X}{\partial u} &amp; \frac{\partial Y}{\partial u} &amp; 0 \\
    \frac{\partial X}{\partial v} &amp; \frac{\partial Y}{\partial v} &amp; 0 \\
   \end{vmatrix}
\]</p>

<h2>The probability of getting \(k\) heads when flipping \(n\) coins is:</h2>

<p>\[P(E) = {n \choose k} p^k (1-p)^{ n-k} \]</p>

<h2>An Identity of Ramanujan</h2>

<p>\[
   \frac{1}{(\sqrt{\phi \sqrt{5}}-\phi) e^{\frac25 \pi}} =
     1+\frac{e^{-2\pi}} {1+\frac{e^{-4\pi}} {1+\frac{e^{-6\pi}}
      {1+\frac{e^{-8\pi}} {1+\ldots} } } }
\]</p>

<h2>A Rogers-Ramanujan Identity</h2>

<p>\[
  1 +  \frac{q^2}{(1-q)}+\frac{q^6}{(1-q)(1-q^2)}+\cdots =
    \prod_{j=0}^{\infty}\frac{1}{(1-q^{5j+2})(1-q^{5j+3})},
     \quad\quad \text{for $|q|&lt;1$}.
\]</p>

<h2>Maxwell's Equations</h2>

<p>
\begin{align}
  \nabla \times \vec{\mathbf{B}} -\, \frac1c\, \frac{\partial\vec{\mathbf{E}}}{\
↪partial t} &amp; = \frac{4\pi}{c}\vec{\mathbf{j}} \\
  \nabla \cdot \vec{\mathbf{E}} &amp; = 4 \pi \rho \\
  \nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial\vec{\mathbf{B}}}{\
↪partial t} &amp; = \vec{\mathbf{0}} \\
  \nabla \cdot \vec{\mathbf{B}} &amp; = 0
\end{align}
</p>

<h2>In-line 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>
```

(continues on next page)

(continuação da página anterior)

```
<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>
```

10.6.2 Mais informação

Uma vez que o visualizador de ebooks do calibre usa a biblioteca MathJax para renderizar texto matemático, o melhor lugar para conseguir mais informação sobre como incluir texto matemático em ebooks e conseguir ajuda é o sítio web do MathJax[101].

10.7 Criar catálogos de AZW3 • EPUB • MOBI

A funcionalidade Criar catálogo do Calibre permite-lhe criar um catálogo da sua biblioteca numa variedade de formatos. Este ficheiro de ajuda descreve as opções de catálogo para quando gerar um catálogo nos formatos AZW3, EPUB e MOBI.

- *Selecionar livros para o catálogo* (página 237)
- *Secções incluídas* (página 238)
- *Prefixos* (página 239)
- *Livros excluídos* (página 239)
- *Géneros excluídos* (página 240)
- *Outras opções* (página 240)
- *Capas personalizadas de catálogo* (página 241)
- *Recursos adicionais de ajuda* (página 241)

10.7.1 Selecionar livros para o catálogo

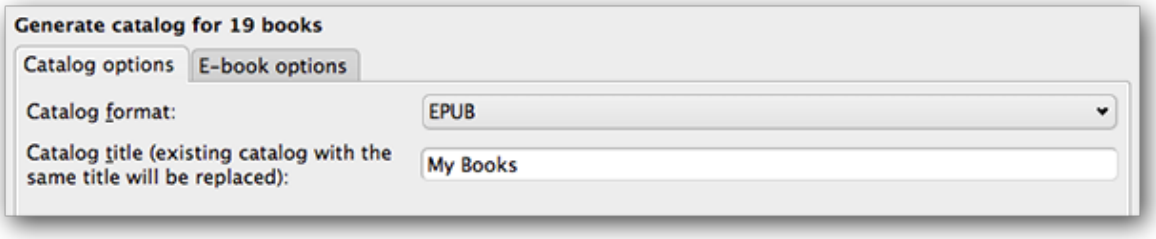
Se pretende ter *toda* a sua biblioteca catalogada, remova todos os critérios de pesquisa ou de filtragem na janela principal. Com um único livro selecionado, todos os livros da sua biblioteca serão candidatos para inclusão no catálogo gerado. Os livros podem ser excluídos individualmente através de critérios vários; veja abaixo a secção *Géneros excluídos* (página 240) para mais informação.

Se quiser um catálogo de apenas uma *parte* da sua biblioteca, tem duas opções:

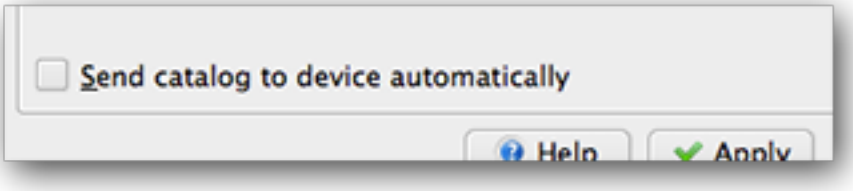
- Cria uma seleção múltipla de livros que pretende ver no catálogo. Com mais do que um livro selecionado na janela principal do Calibre, apenas os livros selecionados serão colocados no catálogo.
- Use o campo de pesquisa ou o navegador de conteúdos para filtrar os livros exibidos. Apenas os livros exibidos serão catalogados.

[101] https://www.mathjax.org

Para iniciar a criação do catálogo, selecione o item de menu *Converter livros* > *Criar um catálogo dos livros na sua biblioteca Calibre*. Pode também adicionar um botão *Criar catálogo* a uma barra de ferramentas em *Preferências* > *Interface* > *Barras e Menus* para um acesso mais fácil ao diálogo de criação de catálogos.

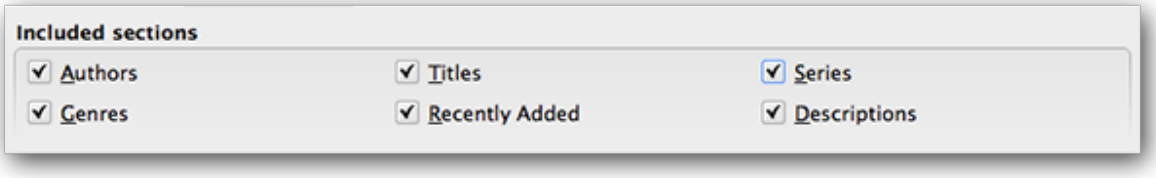


Em *Opções de catálogo*, selecione **AZW3, EPUB ou MOBI** como formato de catálogo. No campo *Título do catálogo*, indique um nome que será usado para o catálogo gerado. Se já existir um catálogo com o mesmo nome e formato será substituído pelo novo catálogo gerado.



Ativar *Enviar catálogo para dispositivo automaticamente* descarregará o catálogo, logo que a sua criação seja concluída, para um dispositivo ligado.

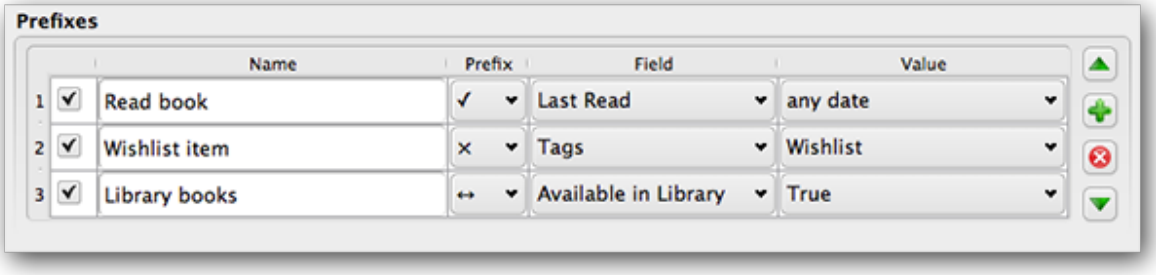
10.7.2 Secções incluídas



Secções ativadas com uma marca serão incluídas no catálogo gerado:

- *Autores* - todos os livros, ordenados por autor, apresentados em formato de lista. Livros sem série associada são listados após livros que estejam associados a uma.
- *Títulos* - todos os livros, ordenados por título, apresentados em formato de lista.
- *Series* - todos os livros que façam parte de uma série, ordenados por série, apresentados em formato de lista.
- *Géneros* - géneros individuais apresentados numa lista, ordenados por Autor e Série.
- *Recentemente adicionados* - todos os livros, ordenados por ordem cronológica inversa. Inclui livros adicionados nos últimos 30 dias, seguida de uma listagem dos livros adicionados por mês.
- *Descrições* - página de descrição detalhada para cada livro, incluindo uma miniatura da capa e comentários. Ordenada por autor, com livros não pertencentes a séries listados antes de livros pertencentes a séries.

10.7.3 Prefixos



Regras prefixas permitem-lhe adicionar um prefixo a listagens de livros quando são verificados certos critérios. Por exemplo, pode querer marcar livros que já leu com um visto, ou livros na sua lista de desejos com um X.

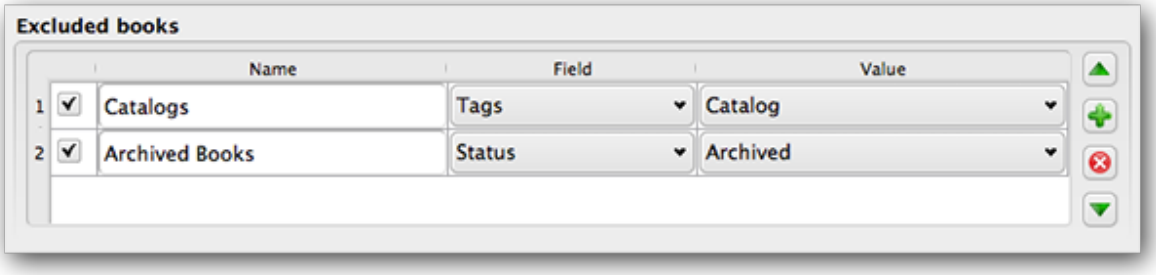
O visto na primeira coluna ativa a regra. *Nome* é o nome de uma regra à sua escolha. *Campo* pode ser *Etiquetas* ou uma coluna personalizada da sua biblioteca. *Valor* é o conteúdo do *Campo* que se pretende fazer coincidir. Quando uma regra prefixa for satisfeita, o livro será marcado com o *Prefixo* selecionado.

Três regras de prefixo foram especificadas no exemplo acima:

1. *Ler livro* especifica que um livro que contenha uma data na coluna personalizada chamada *Lido pela última vez em* será prefixado com um símbolo de visto.
2. O item *Lista de desejos* especifica que qualquer livro com a etiqueta *Lista de desejos* será prefixado com um símbolo X.
3. Livros de *Biblioteca* especifica que qualquer livro com um valor de Verdadeiro (ou Sim) na coluna personalizada *Disponível na Biblioteca* será prefixado com um símbolo de dupla seta.

A primeira regra que resulte numa correspondência fornece o prefixo. Regras desativadas ou incompletas são ignoradas.

10.7.4 Livros excluídos



As regras de exclusão permitem-lhe especificar livros que não serão catalogados.

O visto na primeira coluna ativa a regra. *Nome* é o nome de uma regra à sua escolha. *Campo* pode ser *Etiquetas* ou uma coluna personalizada da sua biblioteca. *Valor* é o conteúdo do *Campo* que se pretende fazer coincidir. Quando uma regra de exclusão for satisfeita, o livro será excluído do catálogo gerado.

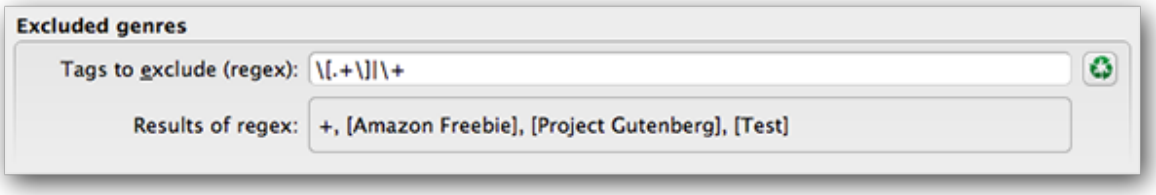
Duas regras de exclusão foram especificadas no exemplo acima:

1. A regra *Catálogos* especifica que qualquer livro com a etiqueta *Catálogos* será excluído do catálogo gerado.

2. A regra de livros *Arquivado* especifica que qualquer livro com um valor de *Arquivado* na coluna personalizada *Estado* será excluído do catálogo gerado.

Todas as regras são avaliadas para cada livro. Regras desativadas ou incompletas são ignoradas.

10.7.5 Géneros excluídos



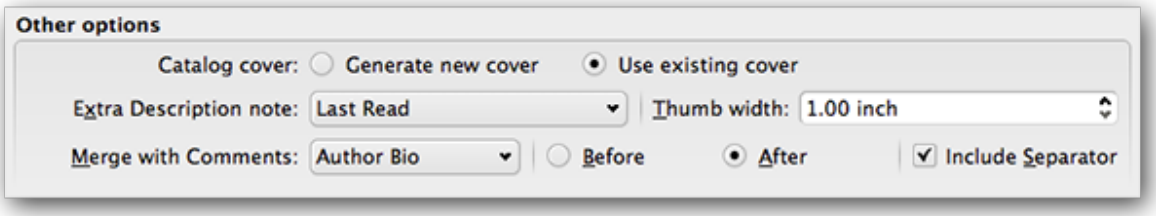
Quando é gerado o catálogo são usadas etiquetas da sua base de dados como géneros. Por exemplo, pode usar as etiquetas «Ficção» e «Não Ficção». Estas etiquetas tornam-se géneros no catálogo gerado, com os livros listados sob a sua respetiva lista de géneros baseada nas etiquetas a eles atribuídas. Um livro será listado em todas as secções de géneros para as quais possua uma etiqueta correspondente.

Pode usar determinadas etiquetas para outros propósitos, talvez um + para indicar um livro já lido, ou uma etiqueta entre colchetes como «[Oferta Amazon]» para indicar a origem do livro. A expressão regular *Géneros excluídos* permite-lhe especificar etiquetas que não pretenda usar como géneros no catálogo gerado. O valor predefinido da expressão regular de exclusão `\[.+\]\+` exclui todas as etiquetas com a forma «[etiqueta]», assim como exclui «+», a etiqueta predefinido para livros já lidos, de serem utilizadas como género no catálogo gerado.

Pode também usar o nome exato de uma etiqueta numa Regex. Por exemplo, `[Amazon Freebie]` ou `[Projecto Gutenberg]`. Se pretender listar múltiplas etiquetas exatas para exclusão, coloque o carácter «barra vertical» (pipe) entre eles: `[Amazon Freebie]|[Projecto Gutenberg]`

Resultados de expressão regular mostra-lhe quais as etiquetas que serão excluídas quando o catálogo for construído, baseado nas etiquetas da sua base de dados e da expressão regular que introduzir. Os resultados são atualizados sempre que modificar a expressão regular.

10.7.6 Outras opções



Capa de catálogo especifica se deva gerar uma nova capa ou usar uma capa já existente. É possível criar uma capa personalizada para os seus catálogos - ver *Capas personalizadas de catálogo* (página 241) para mais informações. Se criou uma capa personalizada que pretende reutilizar, selecione *Usar capa existente*; Caso contrário, selecione *Gerar nova capa*.

Nota de descrição adicional especifica o conteúdo de uma coluna personalizada a ser inserida na página de descrição, junto à miniatura da capa. Por exemplo, pode querer mostrar a data em que leu um livro pela última vez usando uma coluna personalizada *Última leitura*. Para usos avançados da funcionalidade de nota descritiva, veja este tópico no fórum Calibre <https://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395>`_.

Largura da miniatura especifica a preferência de largura para as miniaturas de capa incluídas nas páginas de descrição. As miniaturas são mantidas em cache para melhorar a performance. Para testar diferentes larguras, tente gerar um catálogo com apenas alguns livros até que determine a largura da sua preferência, e apenas então gere o catálogo completo. A primeira vez que um catálogo é gerado, com uma nova largura de miniatura, a performance será mais lenta, mas criações subsequentes do catálogo tirarão partido da cache de miniaturas.

Fundir com comentários especifica uma coluna personalizada cujo conteúdo será fundido em modo não-destrutivo com os metadados da coluna Comentários durante a geração do catálogo. Por exemplo, pode ter uma coluna personalizada *Biografia do autor* que pretende incluir nos metadados de comentários. Pode escolher inserir o conteúdo da coluna personalizada *antes ou depois* da secção de comentários, e opcionalmente, separar o conteúdo incluído com uma linha separadora horizontal. Os tipos de colunas personalizadas elegíveis incluem `texto, comentários, compósito`.

10.7.7 Capas personalizadas de catálogo

Com o plugin Gerar capas[102] instalado, pode criar capas personalizadas para o seu catálogo. Para instalar o plugin, vá a *Preferências* > *Avançadas* > *Plugins* > *Obter novos plugins*.

10.7.8 Recursos adicionais de ajuda

Para mais informação sobre a funcionalidade de catálogo do Calibre, veja o tópico do fórum MobileRead Creating Catalogs - Start here[103], onde pode encontrar informação sobre como personalizar modelos de catálogo e como submeter um registo de erros.

Para colocar questões ou discutir a funcionalidade do catálogo Calibre com outros utilizadores, visite o fórum do MobileRead Library Management[104].

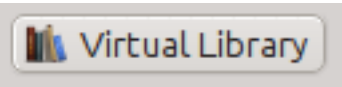
[102] https://www.mobileread.com/forums/showthread.php?t=124219
[103] https://www.mobileread.com/forums/showthread.php?t=118556
[104] https://www.mobileread.com/forums/forumdisplay.php?f=236

10.8 Bibliotecas virtuais

In calibre, a Virtual library is a way to tell calibre to open only a subset of a normal library. For example, you might want to only work with books by a certain author, or books having only a certain tag. Using Virtual libraries is the preferred way of partitioning your large book collection into smaller sub collections. It is superior to splitting up your library into multiple smaller libraries as, when you want to search through your entire collection, you can simply go back to the full library. There is no way to search through multiple separate libraries simultaneously in calibre.

A Virtual library is different from a simple search. A search will only restrict the list of books shown in the book list. A Virtual library does that, and in addition it also restricts the entries shown in the *Tag browser* to the left. The Tag browser will only show tags, authors, series, publishers, etc. that come from the books in the Virtual library. A Virtual library thus behaves as though the actual library contains only the restricted set of books.

10.8.1 Criação de bibliotecas virtuais



To use a Virtual library click the *Virtual library* button located to the left of the Search bar and select the *Create Virtual library* option. As a first example, let's create a Virtual library that shows us only the books by a particular author. Click the *Authors* link as shown in the image below and choose the author you want to use and click OK.

The Create Virtual library dialog has been filled in for you. Click OK and you will see that a new Virtual library has been created, and automatically switched to, that displays only the books by the selected author. As far as calibre is concerned, it is as if your library contains only the books by the selected author.

You can switch back to the full library at any time by once again clicking the *Virtual library* and selecting the entry named *<None>*.

Virtual libraries are based on *searches*. You can use any search as the basis of a Virtual library. The Virtual library will contain only the books matched by that search. First, type in the search you want to use in the Search bar or build a search using the *Tag browser*. When you are happy with the returned results, click the *Virtual library* button, choose *Create library* and enter a name for the new Virtual library. The Virtual library will then be created based on the search you just typed in. Searches are very powerful, for examples of the kinds of things you can do with them, see *A interface de pesquisa* (página 12).

Exemplos de bibliotecas virtuais úteis

- **Livros adicionados ao Calibre no último dia:**
  date:>1daysago
- **Livros adicionados ao Calibre no último mês:**
  data:>30diasatrás
- **Livros com 5 estrelas:**
  rating:5
- **Livros com uma classificação de pelo menos 4 estrelas:**
  rating:>=4
- **Livros sem avaliação:**
  rating:false
- **Periodicals downloaded by the Fetch News function in calibre::**
  tags:=News and author:=calibre
- **Livros sem etiquetas:**
  tags:false
- **Livros sem capas:**
  cover:false

10.8.2 Trabalho com bibliotecas virtuais

You can edit a previously created Virtual library or remove it, by clicking the *Virtual library* and choosing the appropriate action.

You can tell calibre that you always want to apply a particular Virtual library when the current library is opened, by going to *Preferences → Interface → Behavior*.

You can quickly use the current search as a temporary Virtual library by clicking the *Virtual library* button and choosing the **current search* entry.

You can display all available Virtual libraries as tabs above the book list. This is particularly handy if you like switching between Virtual libraries very often. Click the *Virtual library* button and select *Show Virtual libraries as tabs*. You can re-arrange the tabs by drag and drop and close ones you do not want to see. Closed tabs can be restored by right-clicking on the tab bar.

10.8.3 Usar bibliotecas virtuais nas pesquisas

You can search for books that are in a Virtual library using the `vl:` prefix. For example, `vl:Read` will find all the books in the *Read* Virtual library. The search `vl:Read and vl:"Science Fiction"` will find all the books that are in both the *Read* and *Science Fiction* Virtual libraries.

The value following `vl:` must be the name of a Virtual library. If the Virtual library name contains spaces then surround it with quotes.

One use for a Virtual library search is in the Content server. In *Preferences → Sharing over the net → Require username and password* you can limit the calibre libraries visible to a user. For each visible library you can specify a search expression to further limit which books are seen. Use `vl:"Virtual library name"` to limit the books to those in a Virtual library.

10.8.4 Usar restrições adicionais

You can further restrict the books shown in a Virtual library by using *Additional restrictions*. An additional restriction is saved search you previously created that can be applied to the current Virtual library to further restrict the books shown in a Virtual library. For example, say you have a Virtual library for books tagged as *Historical Fiction* and a saved search that shows you unread books, you can click the *Virtual Library* button and choose the *Additional restriction* option to show only unread Historical Fiction books. To learn about saved searches, see *A guardar pesquisas* (página 18).

CAPÍTULO 11

O esquema de URL calibre://

O calibre se registra como programa padrão para abrir URLs calibre://. Assim você pode usar isto para executar ações, como abrir livros, pesquisar por livros, etc. a partir de outros programas/documentos ou por linha de comando. Por exemplo: executar o seguinte na linha de comando:

```
calibre calibre://switch-library/Some_Library
```

Irá abrir o calibre com a biblioteca nomeada «Some Library». Os nomes das bibliotecas são o nome da pasta da biblioteca com espaços substituídos por sublinhados. O valor especial «_» significa a biblioteca atual. Os vários tipos de URLs estão documentados abaixo.

Você pode inserir estes links dentro de arquivos HTML ou documentos do Word ou similares e o sistema operacional irá automaticamente abrir o calibre para executar a ação especificada.

- *Trocar para uma biblioteca específica* (página 246)
- *Mostrar um livro específico no calibre* (página 246)
- *Open a specific book in the E-book viewer at a specific position* (página 246)
- *À procura de livros* (página 247)
- *Codificação Hex dos parâmetros da URL* (página 247)

11.1 Trocar para uma biblioteca específica

A sintaxe da URL é:

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

Os nomes das bibliotecas são o nome da pasta da biblioteca com espaços substituídos por sublinhado. O valor especial «_» significa a biblioteca atual. Você também pode usar :ref:`hex encoding <hex_encoding>`para nomes das bibliotecas, útil se os nomes das bibliotecas possuem caracteres especiais que, de alguma forma, exigiria codificação de URL. Os nomes de biblioteca codificados em hexadecimal se parecem com:

```
_hex_-AD23F4BC
```

Parte onde após o prefixo `` _hex_-`` é o nome da biblioteca codificado como UTF-8 e cada byte representado por dois caracteres hexadecimais.

11.2 Mostrar um livro específico no calibre

A sintaxe da URL é:

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

This will show the book with `book_id` (a number) in calibre. The ids for books can be seen in the calibre interface by hovering over the *Click to open* link in the *Book details* panel, it is the number in brackets at the end of the path to the book folder.

You can copy a link to the current book displayed in calibre by right clicking the *Book details* panel and choosing *Copy link to book*.

If a Virtual library is selected, calibre will use it when showing the book. If the book isn't found in that virtual library then the virtual library is cleared.

If you want to switch to a particular Virtual library when showing the book, use:

```
calibre://show-book/Library_Name/book_id?virtual_library=Library%20Name
or
calibre://show-book/Library_Name/book_id?encoded_virtual_library=hex_encoded_virtual_
↪library_name
```

replacing spaces in the Virtual library name by `%20`. If the book isn't in that virtual library then it is ignored.

11.3 Open a specific book in the E-book viewer at a specific position

A sintaxe da URL é:

```
calibre://view-book/Library_Name/book_id/book_format?open_at=location
```

Aqui, `` book_format`` é o formato do livro, por exemplo, *'EPUB`* ou `MOBI` e `location` é um local opcional dentro do livro. A maneira mais fácil de obter esses links é abrir um livro no visualizador, em seguida, nos controles do visualizador, selecione *Go to → Location* e lá será fornecido um link que você poderá copiar/colar em outro lugar.

11.4 À procura de livros

A sintaxe da URL é:

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

Aqui a consulta e qualquer *search expresion* (página 12) válida. Se a expressão de busca for complicada, :ref:èncode it as a hex string <hex_encoding>` e use `eq` em vez disso. Deixar de fora a consulta fará com que a pesquisa atual seja apagada.

By default, if a Virtual library is selected, calibre will clear it before doing the search to ensure all books are found. If you want to preserve the Virtual library, use:

```
calibre://search/Library_Name?q=query&virtual_library=_
```

If you want to switch to a particular Virtual library, use:

```
calibre://search/Library_Name?virtual_library=Library%20Name
or
calibre://search/Library_Name?encoded_virtual_library=hex_encoded_virtual_library_name
```

replacing spaces in the Virtual library name by `%20`.

If you perform a search in calibre and want to generate a link for it you can do so by right clicking the search bar and choosing *Copy search as URL*.

11.5 Codificação Hex dos parâmetros da URL

A codificação hex dos parâmetros da URL é feita, primeiro codificando os parâmetros como bytes UTF-8, e então substituindo cada byte por dois caracteres hexadecimais representando cada byte. Por exemplo: a string `abc` é os bytes `0x61 0x62 e 0x63` em UTF-8, assim a versão codificada é a string: `616263`.

CAPÍTULO 12

Personalizar Calibre

calibre has a highly modular design. Various parts of it can be customized. Here, you will learn:

- how to use environment variables and *tweaks* to customize calibre's behavior,
- how to specify your own static resources like icons and templates to override the defaults
- how to use *plugins* to add functionality to calibre.
- how to share icon themes and plugins with other calibre users.
- to see how to create *recipes* to add new sources of online content to calibre visit the Section *Adicionar o seu sítio de notícias favorito* (página 29).

Nota: Note that although icon themes and plugins are indexed and downloadable via calibre's builtin updater, they are not part of calibre, and their canonical locations for support and source code are on the Mobileread forums[105] in their support threads.

- *Variáveis de ambiente* (página 282)
- *Ajustes* (página 283)
- *Sobreposição a ícones, temas, etc.* (página 293)
- *Criar o seu próprio tema de ícones para o Calibre* (página 294)
- *Personalizar Calibre com plugins* (página 294)

[105] https://www.mobileread.com/forums/forumdisplay.php?f=166

12.1 Documentação de API para plug-ins

Define várias classes abstratas que podem ser subclassificadas para criar plugins poderosos. As classes úteis são:

- *Plugin* (página 250)
- *FileTypePlugin* (página 252)
- *Plugins de metadados* (página 254)
- *Plugins de catálogo* (página 255)
- *Plugins de descarregamento de metadados* (página 256)
- *Plugins de conversões* (página 259)
- *Drivers de dispositivos* (página 262)
- *User interface actions* (página 276)
- *Preferências de plugins* (página 279)

12.1.1 Plugin

class calibre.customize.**Plugin**(*plugin_path*)

Um plugin do Calibre. Membros úteis incluem:

- `self.installation_type`: Stores how the plugin was installed.
- **`self.plugin_path`: Stores path to the ZIP file that contains** este plugin ou nenhum se for um plugin integrado
- **`self.site_customization`: Guarda um texto de personalização inserido** pelo utilizador.

Métodos que devem ser sobrescritos nas sub classes:

- *`initialize()`* (página 251)
- *`customization_help()`* (página 252)

Métodos úteis:

- *`temporary_file()`* (página 252)
- `__enter__()`
- *`load_resources()`* (página 251)

`supported_platforms = []`

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx" , "linux"]»

`name = 'Trivial Plugin'`

The name of this plugin. You must set it something other than Trivial Plugin for it to work.

`version = (1, 0, 0)`

The version of this plugin as a 3-tuple (major, minor, revision)

`description = 'Não faz absolutamente nada'`

Uma curta descrição do que este plugin faz

author = 'Desconhecido'

O autor deste plugin

priority = 1

When more than one plugin exists for a filetype, the plugins are run in order of decreasing priority. Plugins with higher priority will be run first. The highest possible priority is `sys.maxsize`. Default priority is 1.

minimum_calibre_version = (0, 4, 118)

The earliest version of calibre this plugin requires

installation_type = None

The way this plugin is installed

can_be_disabled = True

Se falso, o usuário não será capaz de desativar este plugin. Use com cuidado.

type = 'Base'

The type of this plugin. Used for categorizing plugins in the GUI

initialize()

Called once when calibre plugins are initialized. Plugins are re-initialized every time a new plugin is added. Also note that if the plugin is run in a worker process, such as for adding books, then the plugin will be initialized for every new worker process.

Perform any plugin specific initialization here, such as extracting resources from the plugin ZIP file. The path to the ZIP file is available as `self.plugin_path`.

Note that `self.site_customization` is **not** available at this point.

config_widget()

Implement this method and *save_settings()* (página 251) in your plugin to use a custom configuration dialog, rather then relying on the simple string based default customization.

This method, if implemented, must return a QWidget. The widget can have an optional method validate() that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns True.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

save_settings(*config_widget*)

Save the settings specified by the user with config_widget.

Parâmetros

config_widget – The widget returned by *config_widget()* (página 251).

do_user_config(*parent=None*)

This method shows a configuration dialog for this plugin. It returns True if the user clicks OK, False otherwise. The changes are automatically applied.

load_resources(*names*)

If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

Por exemplo, para carregar uma imagem

```
pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png'])['images/icon.png
↪'])
icon = QIcon(pixmap)
```

Parâmetros
names – List of paths to resources in the ZIP file using / as separator

Retorno
A dictionary of the form `{name: file_contents}`. Any names that were not found in the ZIP file will not be present in the dictionary.

customization_help(*gui=False*)

Return a string giving help on how to customize this plugin. By default raise a `NotImplementedError`, which indicates that the plugin does not require customization.

If you re-implement this method in your subclass, the user will be asked to enter a string as customization for this plugin. The customization string will be available as `self.site_customization`.

Site customization could be anything, for example, the path to a needed binary on the user's computer.

Parâmetros
gui – Se verdadeiro devolve ajuda em formato HTML; se não, devolve ajuda em formato de texto.

temporary_file(*suffix*)

Devolve um objeto tipo ficheiro que é um ficheiro temporário no sistema de ficheiros. Este ficheiro permanecerá disponível mesmo depois de ser fechado e apenas será removido ao se desligar o interpretador. Use o membro `name` do objeto devolvido para aceder ao caminho completo do ficheiro temporário assim criado.

Parâmetros
suffix – The suffix that the temporary file will have.

cli_main(*args*)

This method is the main entry point for your plugins command line interface. It is called when the user does: calibre-debug -r «Plugin Name». Any arguments passed are present in the args variable.

12.1.2 FileTypePlugin

class calibre.customize.**FileTypePlugin**(*plugin_path*)

Bases: *Plugin* (página 250)

Um plugin que está associado com um determinado conjunto de tipos de arquivo.

file_types = {}

Set of file types for which this plugin should be run. Use "*" for all file types. For example: `{'lit', 'mobi', 'prc'}`

on_import = False

Se Verdadeiro, este plugin é executado quando os livros são adicionados ao banco de dados

on_postimport = False

If True, this plugin is run after books are added to the database. In this case the postimport and postadd methods of the plugin are called.

on_postconvert = False

If True, this plugin is run after a book is converted. In this case the postconvert method of the plugin is called.

on_postdelete = False

If True, this plugin is run after a book file is deleted from the database. In this case the postdelete method of the plugin is called.

on_preprocess = False

Se Verdadeiro, este plugin é executado pouco antes de uma conversão

on_postprocess = False

Se Verdadeiro, este plugin é executado após a conversão no arquivo final produzido pelo plugin de saída de conversão.

type = 'Tipo de ficheiro'

The type of this plugin. Used for categorizing plugins in the GUI

run(*path_to_ebook*)

Run the plugin. Must be implemented in subclasses. It should perform whatever modifications are required on the e-book and return the absolute path to the modified e-book. If no modifications are needed, it should return the path to the original e-book. If an error is encountered it should raise an Exception. The default implementation simply return the path to the original e-book. Note that the path to the original file (before any file type plugins are run, is available as self.original_path_to_file).

The modified e-book file should be created with the `temporary_file()` method.

Parâmetros
path_to_ebook – Caminho absoluto para o ebook.

Retorno
Caminho absoluto para o ebook editado

postimport(*book_id*, *book_format*, *db*)

Called post import, i.e., after the book file has been added to the database. Note that this is different from *postadd()* (página 254) which is called when the book record is created for the first time. This method is called whenever a new file is added to a book record. It is useful for modifying the book record based on the contents of the newly added file.

Parâmetros

- **book_id** – ID da base de dados do livro adicionado.
- **book_format** – O tipo de arquivo do livro que foi adicionado.
- **db** – Library database.

postconvert(*book_id*, *book_format*, *db*)

Called post conversion, i.e., after the conversion output book file has been added to the database. Note that it is run after a conversion only, not after a book is added. It is useful for modifying the book record based on the contents of the newly added file.

Parâmetros

- **book_id** – ID da base de dados do livro adicionado.
- **book_format** – O tipo de arquivo do livro que foi adicionado.
- **db** – Library database.

postdelete(*book_id*, *book_format*, *db*)

Called post deletion, i.e., after the book file has been deleted from the database. Note that it is not run when a book record is deleted, only when one or more formats from the book are deleted. It is useful for modifying the book record based on the format of the deleted file.

Parâmetros

- **book_id** – ID da base de dados do livro adicionado.
- **book_format** – O tipo de arquivo do livro que foi adicionado.

- **db** – Library database.

postadd(*book_id*, *fmt_map*, *db*)

Called post add, i.e. after a book has been added to the db. Note that this is different from *postimport()* (página 253), which is called after a single book file has been added to a book. postadd() is called only when an entire book record with possibly more than one book file has been created for the first time. This is useful if you wish to modify the book record in the database when the book is first added to calibre.

Parâmetros

- **book_id** – ID da base de dados do livro adicionado.
- **fmt_map** – Map of file format to path from which the file format was added. Note that this might or might not point to an actual existing file, as sometimes files are added as streams. In which case it might be a dummy value or a non-existent path.
- **db** – Base de dados da biblioteca

12.1.3 Plugins de metadados

class calibre.customize.**MetadataReaderPlugin**(**args*, ***kwargs*)

Bases: *Plugin* (página 250)

A plugin that implements reading metadata from a set of file types.

file_types = {}

Set of file types for which this plugin should be run. For example: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx" , "linux"]»

version = (6, 14, 1)

The version of this plugin as a 3-tuple (major, minor, revision)

author = 'Kovid Goyal'

O autor deste plugin

type = 'Leitor de metadados'

The type of this plugin. Used for categorizing plugins in the GUI

get_metadata(*stream*, *type*)

Return metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data.

Parâmetros

type – The type of file. Guaranteed to be one of the entries in *file_types* (página 254).

Retorno

Um objeto `calibre.ebooks.metadata.book.Metadata`

class calibre.customize.**MetadataWriterPlugin**(**args*, ***kwargs*)

Bases: *Plugin* (página 250)

A plugin that implements reading metadata from a set of file types.

file_types = {}

Set of file types for which this plugin should be run. For example: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx" , "linux"]»

version = (6, 14, 1)

The version of this plugin as a 3-tuple (major, minor, revision)

author = 'Kovid Goyal'

O autor deste plugin

type = 'Editor de metadados'

The type of this plugin. Used for categorizing plugins in the GUI

set_metadata(*stream*, *mi*, *type*)

Set metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data.

Parâmetros

- **type** – The type of file. Guaranteed to be one of the entries in *file_types* (página 254).
- **mi** – Um objeto `calibre.ebooks.metadata.book.Metadata`

12.1.4 Plugins de catálogo

class calibre.customize.**CatalogPlugin**(*plugin_path*)

Bases: *Plugin* (página 250)

Um plugin que implementa um gerador de catálogos.

file_types = {}

Output file type for which this plugin should be run. For example: "epub" or "xml"

type = 'Gerador de catálogos'

The type of this plugin. Used for categorizing plugins in the GUI

cli_options = []

CLI parser options specific to this plugin, declared as *namedtuple Option*:

from collections import namedtuple Option = namedtuple("Option", "option, default, dest, help") cli_options = [Option("–catalog-title", default = "My Catalog", dest = "catalog_title", help = (_("Title of generated catalog. nDefault:") + « "» + "%default" + «"»))] cli_options parsed in calibre.db.cli.cmd_catalog:option_parser()

initialize()

If plugin is not a built-in, copy the plugin's .ui and .py files from the ZIP file to $TMPDIR. Tab will be dynamically generated and added to the Catalog Options dialog in calibre.gui2.dialogs.catalog.py:Catalog

run(*path_to_output*, *opts*, *db*, *ids*, *notification=None*)

Run the plugin. Must be implemented in subclasses. It should generate the catalog in the format specified in file_types, returning the absolute path to the generated catalog file. If an error is encountered it should raise an Exception.

The generated catalog file should be created with the `temporary_file()` method.

Parâmetros

- **path_to_output** – Absolute path to the generated catalog file.
- **opts** – A dictionary of keyword arguments
- **db** – A LibraryDatabase2 object

12.1.5 Plugins de descarregamento de metadados

class calibre.ebooks.metadata.sources.base.**Source**(**args*, ***kwargs*)

Bases: *Plugin* (página 250)

type = 'Fonte de metadados'

The type of this plugin. Used for categorizing plugins in the GUI

author = 'Kovid Goyal'

O autor deste plugin

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx" , "linux"]»

capabilities = frozenset({})

Set of capabilities supported by this plugin. Useful capabilities are: "identify", "cover"

touched_fields = frozenset({})

List of metadata fields that can potentially be download by this plugin during the identify phase

has_html_comments = False

Set this to True if your plugin returns HTML formatted comments

supports_gzip_transfer_encoding = False

Setting this to True means that the browser object will indicate that it supports gzip transfer encoding. This can speedup downloads but make sure that the source actually supports gzip transfer encoding correctly first

ignore_ssl_errors = False

Set this to True to ignore HTTPS certificate errors when connecting to this source.

cached_cover_url_is_reliable = True

Cached cover URLs can sometimes be unreliable (i.e. the download could fail or the returned image could be bogus). If that is often the case with this source, set to False

options = ()

A list of Option objects. They will be used to automatically construct the configuration widget for this plugin

config_help_message = None

A string that is displayed at the top of the config widget for this plugin

can_get_multiple_covers = False

If True this source can return multiple covers for a given query

auto_trim_covers = False

If set to True covers downloaded by this plugin are automatically trimmed.

prefer_results_with_isbn = True

If set to True, and this source returns multiple results for a query, some of which have ISBNs and some of which do not, the results without ISBNs will be ignored

is_configured()

Return False if your plugin needs to be configured before it can be used. For example, it might need a username/password/API key.

customization_help()

Return a string giving help on how to customize this plugin. By default raise a `NotImplementedError`, which indicates that the plugin does not require customization.

If you re-implement this method in your subclass, the user will be asked to enter a string as customization for this plugin. The customization string will be available as `self.site_customization`.

Site customization could be anything, for example, the path to a needed binary on the user's computer.

Parâmetros

gui – Se verdadeiro devolve ajuda em formato HTML; se não, devolve ajuda em formato de texto.

config_widget()

Implement this method and *save_settings()* (página 257) in your plugin to use a custom configuration dialog, rather then relying on the simple string based default customization.

This method, if implemented, must return a QWidget. The widget can have an optional method validate() that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns True.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

save_settings(*config_widget*)

Save the settings specified by the user with config_widget.

Parâmetros

config_widget – The widget returned by *config_widget()* (página 257).

get_author_tokens(*authors*, *only_first_author=True*)

Take a list of authors and return a list of tokens useful for an AND search query. This function tries to return tokens in first name middle names last name order, by assuming that if a comma is in the author name, the name is in lastname, other names form.

get_title_tokens(*title*, *strip_joiners=True*, *strip_subtitle=False*)

Take a title and return a list of tokens useful for an AND search query. Excludes connectives(optionally) and punctuation.

split_jobs(*jobs*, *num*)

Split a list of jobs into at most num groups, as evenly as possible

test_fields(*mi*)

Return the first field from self.touched_fields that is null on the mi object

clean_downloaded_metadata(*mi*)

Call this method in your plugin's identify method to normalize metadata before putting the Metadata object into result_queue. You can of course, use a custom algorithm suited to your metadata source.

get_book_url(*identifiers*)

Return a 3-tuple or None. The 3-tuple is of the form: (identifier_type, identifier_value, URL). The URL is the URL for the book identified by identifiers at this source. identifier_type, identifier_value specify the identifier corresponding to the URL. This URL must be browsable to by a human using a browser. It is meant to provide a clickable link for the user to easily visit the books page at this source. If no URL is found, return None. This method must be quick, and consistent, so only implement it if it is possible to construct the URL from a known scheme given identifiers.

get_book_url_name(*idtype*, *idval*, *url*)

Return a human readable name from the return value of get_book_url().

get_book_urls(*identifiers*)

Override this method if you would like to return multiple urls for this book. Return a list of 3-tuples. By default this method simply calls *get_book_url()* (página 257).

get_cached_cover_url(*identifiers*)

Return cached cover URL for the book identified by the identifiers dictionary or None if no such URL exists.

Note that this method must only return validated URLs, i.e. not URLS that could result in a generic cover image or a not found error.

id_from_url(*url*)

Parse a URL and return a tuple of the form: (identifier_type, identifier_value). If the URL does not match the pattern for the metadata source, return None.

identify_results_keygen(*title=None*, *authors=None*, *identifiers={}*)

Return a function that is used to generate a key that can sort Metadata objects by their relevance given a search query (title, authors, identifiers).

These keys are used to sort the results of a call to *identify()* (página 258).

For details on the default algorithm see *InternalMetadataCompareKeyGen* (página 259). Re-implement this function in your plugin if the default algorithm is not suitable.

identify(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*)

Identificar um livro pelo seu título/autor/isbn, etc.

If identifiers(s) are specified and no match is found and this metadata source does not store all related identifiers (for example, all ISBNs of a book), this method should retry with just the title and author (assuming they were specified).

If this metadata source also provides covers, the URL to the cover should be cached so that a subsequent call to the get covers API with the same ISBN/special identifier does not need to get the cover URL again. Use the caching API for this.

Cada objeto de metadados colocado na result_queue por este método tem de ter um atributo de *source_relevance* que consiste num valor inteiro que indica a ordem pela qual os resultados foram devolvidos pela fonte de metadados para esta consulta. Este valor inteiro será usado por `compare_identify_results()`. Se a ordem for irrelevante, defina-o para zero em cada resultado.

Verifique se todas as informações de mapeamento de capa/ISBN são armazenadas em cache antes que o objeto metadados seja colocado na fila de resultados (result_queue).

Parâmetros

- **log** – Um objeto de registo de atividade. Use-o para produzir informação de depuração/erros
- **result_queue** – Uma fila de resultados, onde os resultados devem ser colocados. Cada resultado é um objeto de metadados.
- **abort** – If abort.is_set() returns True, abort further processing and return as soon as possible
- **title** – O título do livro, que pode ser Nenhum
- **authors** – Uma lista de autores do livro, que pode ser Nenhum
- **identifiers** – Um dicionário de outros identificadores, sendo os mais comuns {"isbn":"1234…"}
- **timeout** – Tempo de espera em segundos, nenhum pedido à rede deve ficar em espera por mais tempo de que o tempo de espera.

Retorno
Nenhum se não ocorrerem erros, caso contrário a representação em Unicode do erro adequada para mostrar ao utilizador.

download_cover(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*, *get_best_cover=False*)

Descarregue uma capa e coloque-a na result_queue. Os parâmetros têm todos o mesmo significado que para *identify()* (página 258). Coloque (self, cover_data) na result_queue.

Este método deve, sempre que possível e por motivos de eficiência, usar URL de capa que esteja em cache. Quando não estejam presentes dados em cache, a maioria dos plugins simplesmente invoca o método identify e usa os respetivos resultados.

Se o parâmetro get_best_cover for True e este plugin puder obter múltiplas capas, deve apenas obter a «melhor» delas.

class calibre.ebooks.metadata.sources.base.**InternalMetadataCompareKeyGen**(*mi*, *source_plugin*, *title*, *authors*, *identifiers*)

Generate a sort key for comparison of the relevance of Metadata objects, given a search query. This is used only to compare results from the same metadata source, not across different sources.

The sort key ensures that an ascending order sort is a sort by order of decreasing relevance.

O algoritmo é:

- Prefere resultados que têm pelo menos um identificador o mesmo que para a consulta
- Prefer results with a cached cover URL
- Prefer results with all available fields filled in
- Prefer results with the same language as the current user interface language
- Prefer results that are an exact title match to the query
- Prefer results with longer comments (greater than 10% longer)
- **Use the relevance of the result as reported by the metadata source's search** motor

12.1.6 Plugins de conversões

class calibre.customize.conversion.**InputFormatPlugin**(**args*)

Bases: *Plugin* (página 250)

InputFormatPlugins are responsible for converting a document into HTML+OPF+CSS+etc. The results of the conversion *must* be encoded in UTF-8. The main action happens in *convert()* (página 260).

type = 'Formato de origem'

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

Se falso, o usuário não será capaz de desativar este plugin. Use com cuidado.

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx " , "linux"]»

file_types = {}

Conjunto de tipos de arquivos para os quais este plugin deve ser executado Por exemplo: «set(["azw", "mobi", "prc"])»

is_image_collection = False

If True, this input plugin generates a collection of images, one per HTML file. This can be set dynamically, in the convert method if the input files can be both image collections and non-image collections. If you set this to True, you must implement the get_images() method that returns a list of images.

core_usage = 1

Number of CPU cores used by this plugin. A value of -1 means that it uses all available cores

for_viewer = False

Se definido como Verdadeiro, o plug-in de entrada irá executar o processamento especial para fazer a sua saída adequado para visualização

output_encoding = 'utf-8'

The encoding that this input plugin creates files in. A value of None means that the encoding is undefined and must be detected individually

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Options shared by all Input format plugins. Do not override in sub-classes. Use *options* (página 260) instead. Every option must be an instance of `OptionRecommendation`.

options = {}

Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = {}

A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

get_images()

Return a list of absolute paths to the images, if this input plugin represents an image collection. The list of images is in the same order as the spine and the TOC.

convert(*stream*, *options*, *file_ext*, *log*, *accelerators*)

This method must be implemented in sub-classes. It must return the path to the created OPF file or an `OEBBook` instance. All output should be contained in the current folder. If this plugin creates files outside the current folder they must be deleted/marked for deletion before this method returns.

Parâmetros

- **stream** – A file like object that contains the input file.
- **options** – Options to customize the conversion process. Guaranteed to have attributes corresponding to all the options declared by this plugin. In addition, it will have a verbose attribute that takes integral values from zero upwards. Higher numbers mean be more verbose. Another useful attribute is `input_profile` that is an instance of `calibre.customize.profiles.InputProfile`.
- **file_ext** – The extension (without the .) of the input file. It is guaranteed to be one of the *file_types* supported by this plugin.
- **log** – A `calibre.utils.logging.Log` object. All output should use this object.

- **accelarators** – A dictionary of various information that the input plugin can get easily that would speed up the subsequent stages of the conversion.

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

Called to allow the input plugin to perform postprocessing after the book has been parsed.

specialize(*oeb*, *opts*, *log*, *output_fmt*)

Called to allow the input plugin to specialize the parsed book for a particular output format. Called after postprocess_book and before any transforms are performed on the parsed book.

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin input plugins for examples.

class calibre.customize.conversion.**OutputFormatPlugin**(**args*)

Bases: *Plugin* (página 250)

OutputFormatPlugins are responsible for converting an OEB document (OPF+HTML) into an output e-book.

The OEB document can be assumed to be encoded in UTF-8. The main action happens in *convert()* (página 261).

type = 'Formato de saída'

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

Se falso, o usuário não será capaz de desativar este plugin. Use com cuidado.

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx" , "linux"]»

file_type = None

The file type (extension without leading period) that this plugin outputs

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Options shared by all Input format plugins. Do not override in sub-classes. Use *options* (página 261) instead. Every option must be an instance of `OptionRecommendation`.

options = {}

Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = {}

A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

property description

str(object="") -> str str(bytes_or_buffer[, encoding[, errors]]) -> str

Crie um novo objeto string a partir do objeto fornecido. Se a codificação ou os erros forem especificados, o objeto deve expor um buffer de dados que será decodificado usando a codificação fornecida e o manipulador de erros. Caso contrário, retorna o resultado do `object.__str__()` (se definido) ou `repr(object)`. O padrão de codificação é `sys.getdefaultencoding()`. O padrão de erros é "strict".

convert(*oeb_book*, *output*, *input_plugin*, *opts*, *log*)

Render the contents of *oeb_book* (which is an instance of `calibre.ebooks.oeb.OEBBook`) to the file specified by output.

Parâmetros

- **output** – Either a file like object or a string. If it is a string it is the path to a folder that may or may not exist. The output plugin should write its output into that folder. If it is a file like object, the output plugin should write its output into the file.
- **input_plugin** – The input plugin that was used at the beginning of the conversion pipeline.
- **opts** – Conversion options. Guaranteed to have attributes corresponding to the OptionRecommendations of this plugin.
- **log** – The logger. Print debug/info messages etc. using this.

specialize_options (*log*, *opts*, *input_fmt*)

Can be used to change the values of conversion options, as used by the conversion pipeline.

specialize_css_for_output (*log*, *opts*, *item*, *stylizer*)

Can be used to make changes to the css during the CSS flattening process.

Parâmetros

- **item** – O item (ficheiro HTML) a ser processado
- **stylizer** – A Stylizer object containing the flattened styles for item. You can get the style for any element by stylizer.style(element).

gui_configuration_widget (*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin output plugins for examples.

12.1.7 Drivers de dispositivos

A classe base para todos os drivers de dispositivo é *DevicePlugin* (página 262). No entanto, se seu dispositivo se expõe como uma unidade *USBMS* para o sistema operacional, você deve usar a classe *USBMS*, pois ela implementa toda a lógica necessária para oferecer suporte a esses tipos de dispositivos.

class calibre.devices.interface.**DevicePlugin** (*plugin_path*)

Bases: *Plugin* (página 250)

Defines the interface that should be implemented by backends that communicate with an e-book reader.

type = 'Interface do dispositivo'

The type of this plugin. Used for categorizing plugins in the GUI

FORMATS = ['lrf', 'rtf', 'pdf', 'txt']

Lista ordenada de formatos suportados

VENDOR_ID = 0

VENDOR_ID can be either an integer, a list of integers or a dictionary If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

Um inteiro ou uma lista de inteiros

BCD = None

BCD can be either None to not distinguish between devices based on BCD, or it can be a list of the BCD numbers of all devices supported by this driver.

THUMBNAIL_HEIGHT = 68

Altura das miniaturas no dispositivo

THUMBNAIL_COMPRESSION_QUALITY = 75

Compression quality for thumbnails. Set this closer to 100 to have better quality thumbnails with fewer compression artifacts. Of course, the thumbnails get larger as well.

WANTS_UPDATED_THUMBNAILS = False

Set this to True if the device supports updating cover thumbnails during sync_booklists. Setting it to true will ask device.py to refresh the cover thumbnails during book matching

CAN_SET_METADATA = ['title', 'authors', 'collections']

Whether the metadata on books can be set via the GUI.

CAN_DO_DEVICE_DB_PLUGBOARD = False

Whether the device can handle device_db metadata plugboards

path_sep = '/'

Path separator for paths to books on device

icon = 'reader.png'

Ícone para este dispositivo

UserAnnotation

alias of `Annotation`

OPEN_FEEDBACK_MESSAGE = None

GUI displays this as a message if not None in the status bar. Useful if opening can take a long time

VIRTUAL_BOOK_EXTENSIONS = frozenset({})

Set of extensions that are «virtual books» on the device and therefore cannot be viewed/saved/added to library. For example: `frozenset(['kobo'])`

VIRTUAL_BOOK_EXTENSION_MESSAGE = None

Mensagem para mostrar ao usuário para extensões de livros virtuais.

NUKE_COMMENTS = None

Whether to nuke comments in the copy of the book sent to the device. If not None this should be short string that the comments will be replaced by.

MANAGES_DEVICE_PRESENCE = False

If True indicates that this driver completely manages device detection, ejecting and so forth. If you set this to True, you *must* implement the detect_managed_devices and debug_managed_device_detection methods. A driver with this set to true is responsible for detection of devices, managing a blacklist of devices, a list of ejected devices and so forth. calibre will periodically call the detect_managed_devices() method and if it returns a detected device, calibre will call open(). open() will be called every time a device is returned even if previous calls to open() failed, therefore the driver must maintain its own blacklist of failed devices. Similarly, when ejecting, calibre will call eject() and then assuming the next call to detect_managed_devices() returns None, it will call post_yank_cleanup().

SLOW_DRIVEINFO = False

If set the True, calibre will call the *get_driveinfo()* (página 266) method after the books lists have been loaded to get the driveinfo.

ASK_TO_ALLOW_CONNECT = False

If set to True, calibre will ask the user if they want to manage the device with calibre, the first time it is detected. If you set this to True you must implement *get_device_uid()* (página 268) and *ignore_connected_device()* (página 268) and *get_user_blacklisted_devices()* (página 268) and *set_user_blacklisted_devices()* (página 268)

user_feedback_after_callback = None

Set this to a dictionary of the form {"title":title, "msg":msg, "det_msg":detailed_msg} to have calibre popup a message to the user after some callbacks are run (currently only upload_books). Be careful to not spam the user with too many messages. This variable is checked after *every* callback, so only set it when you really need to.

classmethod get_open_popup_message()

GUI displays this as a non-modal popup. Should be an instance of OpenPopupMessage

is_usb_connected(*devices_on_system*, *debug=False*, *only_presence=False*)

Return True, device_info if a device handled by this plugin is currently connected.

Parâmetros

devices_on_system – Lista de dispositivos atualmente ligados

detect_managed_devices(*devices_on_system*, *force_refresh=False*)

Called only if MANAGES_DEVICE_PRESENCE is True.

Scan for devices that this driver can handle. Should return a device object if a device is found. This object will be passed to the open() method as the connected_device. If no device is found, return None. The returned object can be anything, calibre does not use it, it is only passed to open().

This method is called periodically by the GUI, so make sure it is not too resource intensive. Use a cache to avoid repeatedly scanning the system.

Parâmetros

- **devices_on_system** – Conjunto de dispositivos USB encontrados no sistema.
- **force_refresh** – If True and the driver uses a cache to prevent repeated scanning, the cache must be flushed.

debug_managed_device_detection(*devices_on_system*, *output*)

Called only if MANAGES_DEVICE_PRESENCE is True.

Should write information about the devices detected on the system to output, which is a file like object.

Should return True if a device was detected and successfully opened, otherwise False.

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Parâmetros

- **key** – A chave para desbloquear o dispositivo
- **log_packets** – If true the packet stream to/from the device is logged
- **report_progress** – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information
- **detected_device** – Device information from the device scanner

can_handle_windows(*usbdevice*, *debug=False*)

Optional method to perform further checks on a device to see if this driver is capable of handling it. If it is not it should return False. This method is only called after the vendor, product ids and the bcd have matched, so it can do some relatively time intensive checks. The default implementation returns True. This method is called only on Windows. See also *can_handle()* (página 265).

Note that for devices based on USBMS this method by default delegates to *can_handle()* (página 265). So you only need to override *can_handle()* (página 265) in your subclass of USBMS.

Parâmetros

usbdevice – A usbdevice as returned by `calibre.devices.winusb.scan_usb_devices()`

can_handle(*device_info*, *debug=False*)

Unix version of *can_handle_windows()* (página 264).

Parâmetros

device_info – Is a tuple of (vid, pid, bcd, manufacturer, product, serial number)

open(*connected_device*, *library_uuid*)

Perform any device specific initialization. Called after the device is detected but before any other functions that communicate with the device. For example: For devices that present themselves as USB Mass storage devices, this method would be responsible for mounting the device or if the device has been automounted, for finding out where it has been mounted. The method *calibre.devices.usbms.device.Device.open()* (página 273) has an implementation of this function that should serve as a good example for USB Mass storage devices.

This method can raise an OpenFeedback exception to display a message to the user.

Parâmetros

- **connected_device** – The device that we are trying to open. It is a tuple of (vendor id, product id, bcd, manufacturer name, product name, device serial number). However, some devices have no serial number and on Windows only the first three fields are present, the rest are None.
- **library_uuid** – The UUID of the current calibre library. Can be None if there is no library (for example when used from the command line).

eject()

Un-mount / eject the device from the OS. This does not check if there are pending GUI jobs that need to communicate with the device.

NOTE: That this method may not be called on the same thread as the rest of the device methods.

post_yank_cleanup()

Called if the user yanks the device without ejecting it first.

set_progress_reporter(*report_progress*)

Set a function to report progress information.

Parâmetros

report_progress – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information

get_device_information(*end_session=True*)

Ask device for device information. See L{DeviceInfoQuery}.

Retorno
(device name, device version, software version on device, MIME type) The tuple can optionally have a fifth element, which is a drive information dictionary. See usbms.driver for an example.

get_driveinfo()

Return the driveinfo dictionary. Usually called from get_device_information(), but if loading the driveinfo is slow for this driver, then it should set SLOW_DRIVEINFO. In this case, this method will be called by calibre after the book lists have been loaded. Note that it is not called on the device thread, so the driver should cache the drive info in the books() method and this function should return the cached data.

card_prefix(*end_session=True*)

Return a 2 element list of the prefix to paths on the cards. If no card is present None is set for the card's prefix. E.G. ("/place", "/place2") (None, "place2") ("place", None) (None, None)

total_space(*end_session=True*)

Get total space available on the mountpoints:

1. Memória principal
2. Cartão de memória A
3. Cartão de memória B

Retorno
A 3 element list with total space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return 0.

free_space(*end_session=True*)

Get free space available on the mountpoints:

1. Memória principal
2. Cartão A
3. Cartão B

Retorno
A 3 element list with free space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return -1.

books(*oncard=None*, *end_session=True*)

Devolve a lista de livros no dispositivo.

Parâmetros
oncard – If "carda" or "cardb" return a list of e-books on the specific storage card, otherwise return list of e-books in main memory of device. If a card is specified and no books are on the card return empty list.

Retorno
Uma lista de livros.

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)

Upload a list of books to the device. If a file already exists on the device, it should be replaced. This method should raise a `FreeSpaceError` if there is not enough free space on the device. The text of the FreeSpaceError must contain the word «card» if `on_card` is not None otherwise it must contain the word «memory».

Parâmetros

- **files** – Lista de caminhos
- **names** – A list of file names that the books should have once uploaded to the device. len(names) == len(files)
- **metadata** – If not None, it is a list of `Metadata` objects. The idea is to use the metadata to determine where on the device to put the book. len(metadata) == len(files). Apart from the regular cover (path to cover), there may also be a thumbnail attribute, which should be used in preference. The thumbnail attribute is of the form (width, height, cover_data as jpeg).

Retorno
A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (página 267).

classmethod add_books_to_metadata(*locations*, *metadata*, *booklists*)
Add locations to the booklists. This function must not communicate with the device.

Parâmetros

- **locations** – Result of a call to L{upload_books}
- **metadata** – List of `Metadata` objects, same as for *upload_books()* (página 266).
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard="cardb")`).

delete_books(*paths*, *end_session=True*)
Eliminar livros em caminhos no dispositivo

classmethod remove_books_from_metadata(*paths*, *booklists*)
Remove books from the metadata list. This function must not communicate with the device.

Parâmetros

- **paths** – caminhos para livros o dispositivo.
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard="cardb")`).

sync_booklists(*booklists*, *end_session=True*)
Atualizar metadados no dispositivo.

Parâmetros
booklists – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard="cardb")`).

get_file(*path*, *outfile*, *end_session=True*)
Read the file at `path` on the device and write it to outfile.

Parâmetros
outfile – file object like `sys.stdout` or the result of an *open()* (página 265) call.

classmethod config_widget()
Should return a QWidget. The QWidget contains the settings for the device interface

classmethod save_settings(*settings_widget*)
Should save settings to disk. Takes the widget created in *config_widget()* (página 267) and saves all settings to disk.

classmethod settings()
Should return an opts object. The opts object should have at least one attribute *format_map* which is an ordered list of formats for the device.

set_plugboards(*plugboards*, *pb_func*)

provide the driver the current set of plugboards and a function to select a specific plugboard. This method is called immediately before add_books and sync_booklists.

pb_func is a callable with the following signature::
def pb_func(device_name, format, plugboards)

You give it the current device name (either the class name or DEVICE_PLUGBOARD_NAME), the format you are interested in (a "real" format or "device_db"), and the plugboards (you were given those by set_plugboards, the same place you got this method).

Retorno
Nenhum ou um quadro de conexões individual

set_driveinfo_name(*location_code*, *name*)

Set the device name in the driveinfo file to "name". This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the get_device_information() method.

prepare_addable_books(*paths*)

Given a list of paths, returns another list of paths. These paths point to addable versions of the books.

If there is an error preparing a book, then instead of a path, the position in the returned list for that book should be a three tuple: (original_path, the exception instance, traceback)

startup()

Called when calibre is starting the device. Do any initialization required. Note that multiple instances of the class can be instantiated, and thus __init__ can be called multiple times, but only one instance will have this method called. This method is called on the device thread, not the GUI thread.

shutdown()

Called when calibre is shutting down, either for good or in preparation to restart. Do any cleanup required. This method is called on the device thread, not the GUI thread.

get_device_uid()

Must return a unique id for the currently connected device (this is called immediately after a successful call to open()). You must implement this method if you set ASK_TO_ALLOW_CONNECT = True

ignore_connected_device(*uid*)

Should ignore the device identified by uid (the result of a call to get_device_uid()) in the future. You must implement this method if you set ASK_TO_ALLOW_CONNECT = True. Note that this function is called immediately after open(), so if open() caches some state, the driver should reset that state.

get_user_blacklisted_devices()

Return map of device uid to friendly name for all devices that the user has asked to be ignored.

set_user_blacklisted_devices(*devices*)

Set the list of device uids that should be ignored by this driver.

specialize_global_preferences(*device_prefs*)

Implement this method if your device wants to override a particular preference. You must ensure that all call sites that want a preference that can be overridden use device_prefs["something"] instead of prefs["something"]. Your method should call device_prefs.set_overrides(pref=val, pref=val, …). Currently used for: metadata management (prefs["manage_device_metadata"])

set_library_info(*library_name*, *library_uuid*, *field_metadata*)

Implement this method if you want information about the current calibre library. This method is called at startup and when the calibre library changes while connected.

is_dynamically_controllable()

Called by the device manager when starting plugins. If this method returns a string, then a) it supports the device manager's dynamic control interface, and b) that name is to be used when talking to the plugin.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

start_plugin()

This method is called to start the plugin. The plugin should begin to accept device connections however it does that. If the plugin is already accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

stop_plugin()

This method is called to stop the plugin. The plugin should no longer accept connections, and should cleanup behind itself. It is likely that this method should call shutdown. If the plugin is already not accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

get_option(*opt_string*, *default=None*)

Return the value of the option indicated by opt_string. This method can be called when the plugin is not started. Return None if the option does not exist.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

set_option(*opt_string*, *opt_value*)

Set the value of the option indicated by opt_string. This method can be called when the plugin is not started.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

is_running()

Devolve verdadeiro se o plugin foi iniciado, caso contrário é falso

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

synchronize_with_db(*db*, *book_id*, *book_metadata*, *first_call*)

Called during book matching when a book on the device is matched with a book in calibre's db. The method is responsible for synchronizing data from the device to calibre's db (if needed).

The method must return a two-value tuple. The first value is a set of calibre book ids changed if calibre's database was changed or None if the database was not changed. If the first value is an empty set then the metadata for the book on the device is updated with calibre's metadata and given back to the device, but no GUI refresh of that book is done. This is useful when the calibre data is correct but must be sent to the device.

The second value is itself a 2-value tuple. The first value in the tuple specifies whether a book format should be sent to the device. The intent is to permit verifying that the book on the device is the same as the book in calibre. This value must be None if no book is to be sent, otherwise return the base file name on the device (a string like foobar.epub). Be sure to include the extension in the name. The device subsystem will construct a send_books job for all books with not- None returned values. Note: other than to later retrieve the extension, the name is ignored in cases where the device uses a template to generate the file name, which most do. The second value in the returned tuple indicated whether the format is future-dated. Return True if it is, otherwise return False. calibre will display a dialog to the user listing all future dated books.

Extremely important: this method is called on the GUI thread. It must be threadsafe with respect to the device manager's thread.

book_id: the calibre id for the book in the database. book_metadata: the Metadata object for the book coming from the device. first_call: True if this is the first call during a sync, False otherwise

class calibre.devices.interface.**BookList**(*oncard*, *prefix*, *settings*)

Bases: `list`

Uma lista de livros. Cada objeto de livro deve ter os campos

1. título
2. autores
3. size (tamanho do livro)
4. datetime (uma enupla de hora UTC)
5. path (caminho, no dispositivo, para o livro)
6. thumbnail (can be None) thumbnail is either a str/bytes object with the image data or it should have an attribute image_path that stores an absolute (platform native) path to the image
7. etiquetas (uma lista de entradas, pode estar vazia).

supports_collections()

Return True if the device supports collections for this book list.

add_book(*book*, *replace_metadata*)

Add the book to the booklist. Intent is to maintain any device-internal metadata. Return True if booklists must be sync'ed

remove_book(*book*)

Remove a book from the booklist. Correct any device metadata at the same time

get_collections(*collection_attributes*)

Return a dictionary of collections created from collection_attributes. Each entry in the dictionary is of the form collection name:[list of books]

The list of books is sorted by book title, except for collections created from series, in which case series_index is used.

Parâmetros

collection_attributes – A list of attributes of the Book object

Dispositivos USB de armazenamento de ficheiros

A classe base para tais dispositivos é *calibre.devices.usbms.driver.USBMS* (página 273). Essa classe, por sua vez, herda algumas de suas funcionalidades de suas bases, documentadas a seguir. Um driver básico baseado em *USBMS* típico se parece com isto:

```
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]
```

(continues on next page)

(continuação da página anterior)

```
    PRODUCT_ID  = [0xb004]
    BCD         = [0x224]

    THUMBNAIL_HEIGHT = 144

    EBOOK_DIR_MAIN = 'eBooks'
    SUPPORTS_SUB_DIRS = False

    def upload_cover(self, path, filename, metadata):
        coverdata = getattr(metadata, 'thumbnail', None)
        if coverdata and coverdata[2]:
            with open('%s.jpg' % os.path.join(path, filename), 'wb') as coverfile:
                coverfile.write(coverdata[2])
```

class calibre.devices.usbms.device.**Device**(*plugin_path*)

Bases: DeviceConfig, *DevicePlugin* (página 262)

This class provides logic common to all drivers for devices that export themselves as USB Mass Storage devices. Provides implementations for mounting/ejecting of USBMS devices on all platforms.

VENDOR_ID = 0

VENDOR_ID can be either an integer, a list of integers or a dictionary If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

Um inteiro ou uma lista de inteiros

BCD = None

BCD can be either None to not distinguish between devices based on BCD, or it can be a list of the BCD numbers of all devices supported by this driver.

WINDOWS_MAIN_MEM = None

String identifying the main memory of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_A_MEM = None

String identifying the first card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_B_MEM = None

String identifying the second card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

OSX_MAIN_MEM_VOL_PAT = None

Used by the new driver detection to disambiguate main memory from storage cards. Should be a regular expression that matches the main memory mount point assigned by macOS

BACKLOADING_ERROR_MESSAGE = None

MAX_PATH_LEN = 250

O comprimento máximo do caminhos criado no dispositivo

NEWS_IN_FOLDER = True

Colocar notícias na sua própria pasta

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Parâmetros

- **key** – A chave para desbloquear o dispositivo
- **log_packets** – If true the packet stream to/from the device is logged
- **report_progress** – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information
- **detected_device** – Device information from the device scanner

set_progress_reporter(*report_progress*)

Set a function to report progress information.

Parâmetros

report_progress – Function that is called with a % progress (number between 0 and 100) for various tasks. If it is called with -1 that means that the task does not have any progress information

card_prefix(*end_session=True*)

Return a 2 element list of the prefix to paths on the cards. If no card is present None is set for the card's prefix. E.G. ("/place", "/place2") (None, "place2") ("place", None) (None, None)

total_space(*end_session=True*)

Get total space available on the mountpoints:

1. Memória principal
2. Cartão de memória A
3. Cartão de memória B

Retorno

A 3 element list with total space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return 0.

free_space(*end_session=True*)

Get free space available on the mountpoints:

1. Memória principal
2. Cartão A
3. Cartão B

Retorno

A 3 element list with free space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return -1.

windows_sort_drives(*drives*)

Called to disambiguate main memory and storage card for devices that do not distinguish between them on the basis of *WINDOWS_CARD_NAME*. For example: The EB600

can_handle_windows(*usbdevice*, *debug=False*)

Optional method to perform further checks on a device to see if this driver is capable of handling it. If it is not it should return False. This method is only called after the vendor, product ids and the bcd have matched, so it can do some relatively time intensive checks. The default implementation returns True. This method is called only on Windows. See also `can_handle()`.

Note that for devices based on USBMS this method by default delegates to `can_handle()`. So you only need to override `can_handle()` in your subclass of USBMS.

Parâmetros

usbdevice – A usbdevice as returned by `calibre.devices.winusb.scan_usb_devices()`

open(*connected_device*, *library_uuid*)

Perform any device specific initialization. Called after the device is detected but before any other functions that communicate with the device. For example: For devices that present themselves as USB Mass storage devices, this method would be responsible for mounting the device or if the device has been automounted, for finding out where it has been mounted. The method *calibre.devices.usbms.device.Device.open()* (página 273) has an implementation of this function that should serve as a good example for USB Mass storage devices.

This method can raise an OpenFeedback exception to display a message to the user.

Parâmetros

- **connected_device** – The device that we are trying to open. It is a tuple of (vendor id, product id, bcd, manufacturer name, product name, device serial number). However, some devices have no serial number and on Windows only the first three fields are present, the rest are None.
- **library_uuid** – The UUID of the current calibre library. Can be None if there is no library (for example when used from the command line).

eject()

Un-mount / eject the device from the OS. This does not check if there are pending GUI jobs that need to communicate with the device.

NOTE: That this method may not be called on the same thread as the rest of the device methods.

post_yank_cleanup()

Called if the user yanks the device without ejecting it first.

sanitize_callback(*path*)

Callback to allow individual device drivers to override the path sanitization used by `create_upload_path()`.

filename_callback(*default*, *mi*)

Callback to allow drivers to change the default file name set by `create_upload_path()`.

sanitize_path_components(*components*)

Perform any device specific sanitization on the path components for files to be uploaded to the device

get_annotations(*path_map*)

Resolve path_map to annotation_map of files found on the device

add_annotation_to_library(*db*, *db_id*, *annotation*)

Adicionar uma nota à biblioteca Calibre

class calibre.devices.usbms.cli.**CLI**

class calibre.devices.usbms.driver.**USBMS**(*plugin_path*)

Bases: *CLI* (página 273), *Device* (página 271)

The base class for all USBMS devices. Implements the logic for sending/getting/updating metadata/caching metadata/etc.

description = 'Comunicar com um leitor de livros digitais.'

Uma curta descrição do que este plugin faz

author = 'John Schember'

O autor deste plugin

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx " , "linux "]»

booklist_class

alias of `BookList`

book_class

alias of `Book`

FORMATS = []

Lista ordenada de formatos suportados

CAN_SET_METADATA = []

Whether the metadata on books can be set via the GUI.

get_device_information(*end_session=True*)

Ask device for device information. See L{DeviceInfoQuery}.

Retorno

(device name, device version, software version on device, MIME type) The tuple can optionally have a fifth element, which is a drive information dictionary. See usbms.driver for an example.

set_driveinfo_name(*location_code*, *name*)

Set the device name in the driveinfo file to "name". This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the get_device_information() method.

books(*oncard=None*, *end_session=True*)

Devolve a lista de livros no dispositivo.

Parâmetros

oncard – If "carda" or "cardb" return a list of e-books on the specific storage card, otherwise return list of e-books in main memory of device. If a card is specified and no books are on the card return empty list.

Retorno

Uma lista de livros.

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)

Upload a list of books to the device. If a file already exists on the device, it should be replaced. This method should raise a `FreeSpaceError` if there is not enough free space on the device. The text of the FreeSpaceError must contain the word «card» if `on_card` is not None otherwise it must contain the word «memory».

Parâmetros

- **files** – Lista de caminhos
- **names** – A list of file names that the books should have once uploaded to the device. len(names) == len(files)
- **metadata** – If not None, it is a list of `Metadata` objects. The idea is to use the metadata to determine where on the device to put the book. len(metadata) == len(files). Apart from the regular cover (path to cover), there may also be a thumbnail attribute, which should be used in preference. The thumbnail attribute is of the form (width, height, cover_data as jpeg).

Retorno
A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (página 275).

upload_cover (*path*, *filename*, *metadata*, *filepath*)
Upload book cover to the device. Default implementation does nothing.

Parâmetros

- **path** – The full path to the folder where the associated book is located.
- **filename** – O nome do arquivo do livro sem a extensão.
- **metadata** – metadata belonging to the book. Use metadata.thumbnail for cover
- **filepath** – O caminho absoluto para o ficheiro de ebook

add_books_to_metadata (*locations*, *metadata*, *booklists*)
Add locations to the booklists. This function must not communicate with the device.

Parâmetros

- **locations** – Result of a call to L{upload_books}
- **metadata** – List of `Metadata` objects, same as for *upload_books()* (página 274).
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard="cardb")`).

delete_books (*paths*, *end_session=True*)
Eliminar livros em caminhos no dispositivo

remove_books_from_metadata (*paths*, *booklists*)
Remove books from the metadata list. This function must not communicate with the device.

Parâmetros

- **paths** – caminhos para livros o dispositivo.
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard="cardb")`).

sync_booklists (*booklists*, *end_session=True*)
Atualizar metadados no dispositivo.

Parâmetros
booklists – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard="cardb")`).

classmethod normalize_path (*path*)
Return path with platform native path separators

12.1.8 User interface actions

Se você estiver adicionando seu próprio plug-in em um arquivo ZIP, deve criar uma subclasse de `InterfaceAction-Base` e `InterfaceAction`. O método `load_actual_plugin()` de sua subclasse `InterfaceActionBase` deve retornar um objeto instanciado de sua subclasse `InterfaceBase`.

class `calibre.gui2.actions.`**InterfaceAction**(*parent*, *site_customization*)

Bases: `QObject`

A plugin representing an «action» that can be taken in the graphical user interface. All the items in the toolbar and context menus are implemented by these plugins.

Note that this class is the base class for these plugins, however, to integrate the plugin with calibre's plugin system, you have to make a wrapper class that references the actual plugin. See the `calibre.customize.builtins` module for examples.

If two *`InterfaceAction`* (página 276) objects have the same name, the one with higher priority takes precedence.

Sub-classes should implement the *`genesis()`* (página 278), *`library_changed()`* (página 278), *`location_selected()`* (página 278), *`shutting_down()`* (página 279), *`initialization_complete()`* (página 278) and *`tag_browser_context_action()`* (página 278) methods.

Once initialized, this plugin has access to the main calibre GUI via the `gui` member. You can access other plugins by name, for example:

```
self.gui.iactions['Save To Disk']
```

To access the actual plugin, use the `interface_action_base_plugin` attribute, this attribute only becomes available after the plugin has been initialized. Useful if you want to use methods from the plugin class like do_user_config().

The QAction specified by *`action_spec`* (página 276) is automatically create and made available as `self.qaction`.

name = 'Implement me'

The plugin name. If two plugins with the same name are present, the one with higher priority takes precedence.

priority = 1

The plugin priority. If two plugins with the same name are present, the one with higher priority takes precedence.

popup_type = 1

The menu popup type for when this plugin is added to a toolbar

auto_repeat = False

Whether this action should be auto repeated when its shortcut key is held down.

action_spec = ('text', 'icon', None, None)

Of the form: (text, icon_path, tooltip, keyboard shortcut). icon, tooltip and keyboard shortcut can be None. keyboard shortcut must be either a string, None or tuple of shortcuts. If None, a keyboard shortcut corresponding to the action is not registered. If you pass an empty tuple, then the shortcut is registered with no default key binding.

action_add_menu = False

Se True, será criado um menu e adicionado automaticamente a self.qaction

action_menu_clone_qaction = False

Se True, um clone de self.qaction é adicionado ao menu de self.qaction. Se pretender que o texto desta ação seja diferente do de self.qaction, modifique esta variável para o novo texto

dont_add_to = frozenset({})

Conjunto de locais aos quais esta ação não pode ser adicionada. Veja `all_locations` para uma lista de locais possíveis

dont_remove_from = frozenset({})

Conjunto de locais dos quais esta ação não pode ser removida. Ver `all_locations` para uma lista de locais possíveis

action_type = 'global'

Tipo de ação "corrente" significa que ações na vista "global" atual implicam uma ação que não se repercute na vista atual, mas sim no Calibre como um todo

accepts_drops = False

If True, then this InterfaceAction will have the opportunity to interact with drag and drop events. See the methods, *accept_enter_event()* (página 277), :meth`:accept_drag_move_event`, *drop_event()* (página 277) for details.

accept_enter_event(*event*, *mime_data*)

This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

accept_drag_move_event(*event*, *mime_data*)

This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

drop_event(*event*, *mime_data*)

This method should perform some useful action and return True iff this interface action is capable of handling the drop event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI. You should not perform blocking/long operations in this function. Instead emit a signal or use QTimer.singleShot and return quickly. See the builtin actions for examples.

create_menu_action(*menu*, *unique_name*, *text*, *icon=None*, *shortcut=None*, *description=None*, *triggered=None*, *shortcut_name=None*, *persist_shortcut=False*)

Convenience method to easily add actions to a QMenu. Returns the created QAction. This action has one extra attribute calibre_shortcut_unique_name which if not None refers to the unique name under which this action is registered with the keyboard manager.

Parâmetros

- **menu** – The QMenu the newly created action will be added to
- **unique_name** – A unique name for this action, this must be globally unique, so make it as descriptive as possible. If in doubt, add an UUID to it.
- **text** – O texto ou a ação.
- **icon** – Either a QIcon or a file name. The file name is passed to the QIcon.ic() builtin, so you do not need to pass the full path to the images folder.
- **shortcut** – A string, a list of strings, None or False. If False, no keyboard shortcut is registered for this action. If None, a keyboard shortcut with no default keybinding is registered. String and list of strings register a shortcut with default keybinding as specified.
- **description** – A descrição para esta ação. Usado para definir dicas de ferramentas.
- **triggered** – A callable which is connected to the triggered signal of the created action.

- **shortcut_name** – The text displayed to the user when customizing the keyboard shortcuts for this action. By default it is set to the value of `text`.
- **persist_shortcut** – Shortcuts for actions that don't always appear, or are library dependent, may disappear when other keyboard shortcuts are edited unless `persist_shortcut` is set True.

load_resources(*names*)

If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

Por exemplo, para carregar uma imagem

```
pixmap = QPixmap()
pixmap.loadFromData(tuple(self.load_resources(['images/icon.png']).
↪values())[0])
icon = QIcon(pixmap)
```

Parâmetros

names – List of paths to resources in the ZIP file using / as separator

Retorno

A dictionary of the form `{name : file_contents}`. Any names that were not found in the ZIP file will not be present in the dictionary.

genesis()

Setup this plugin. Only called once during initialization. self.gui is available. The action specified by *action_spec* (página 276) is available as `self.qaction`.

location_selected(*loc*)

Called whenever the book list being displayed in calibre changes. Currently values for loc are: `library, main, card and cardb`.

This method should enable/disable this action and its sub actions as appropriate for the location.

library_about_to_change(*olddb*, *db*)

Chamado toda vez que a biblioteca atual é alterada.

Parâmetros

- **olddb** – The LibraryDatabase corresponding to the previous library.
- **db** – The LibraryDatabase corresponding to the new library.

library_changed(*db*)

Chamado toda vez que a biblioteca atual é alterada.

Parâmetros

db – The LibraryDatabase corresponding to the current library.

gui_layout_complete()

Called once per action when the layout of the main GUI is completed. If your action needs to make changes to the layout, they should be done here, rather than in *initialization_complete()* (página 278).

initialization_complete()

Chamada uma vez por ação quando a inicialização do GUI principal é concluida.

tag_browser_context_action(*index*)

Called when displaying the context menu in the Tag browser. `index` is the QModelIndex that points to the Tag browser item that was right clicked. Test it for validity with index.valid() and get the underlying TagTreeItem object with index.data(Qt.ItemDataRole.UserRole). Any action objects yielded by this method will be added to the context menu.

shutting_down()

Called once per plugin when the main GUI is in the process of shutting down. Release any used resources, but try not to block the shutdown for long periods of time.

class calibre.customize.**InterfaceActionBase**(**args*, ***kwargs*)

Bases: *Plugin* (página 250)

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx" , "linux"]»

author = 'Kovid Goyal'

O autor deste plugin

type = 'Ações da interface de utilizador'

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

Se falso, o usuário não será capaz de desativar este plugin. Use com cuidado.

load_actual_plugin(*gui*)

This method must return the actual interface action plugin object.

12.1.9 Preferências de plugins

class calibre.customize.**PreferencesPlugin**(*plugin_path*)

Bases: *Plugin* (página 250)

A plugin representing a widget displayed in the Preferences dialog.

This plugin has only one important method *create_widget()* (página 280). The various fields of the plugin control how it is categorized in the UI.

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : « [" windows ", " osx" , "linux"]»

author = 'Kovid Goyal'

O autor deste plugin

type = 'Preferências'

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

Se falso, o usuário não será capaz de desativar este plugin. Use com cuidado.

config_widget = None

Import path to module that contains a class named ConfigWidget which implements the ConfigWidgetInterface. Used by *create_widget()* (página 280).

category_order = 100

Where in the list of categories the *category* (página 280) of this plugin should be.

name_order = 100

Where in the list of names in a category, the *gui_name* (página 280) of this plugin should be

category = None

The category this plugin should be in

gui_category = None

The category name displayed to the user for this plugin

gui_name = None

The name displayed to the user for this plugin

icon = None

The icon for this plugin, should be an absolute path

description = None

The description used for tooltips and the like

create_widget(*parent=None*)

Create and return the actual Qt widget used for setting this group of preferences. The widget must implement the *calibre.gui2.preferences.ConfigWidgetInterface* (página 280).

The default implementation uses *config_widget* (página 279) to instantiate the widget.

class calibre.gui2.preferences.**ConfigWidgetInterface**

This class defines the interface that all widgets displayed in the Preferences dialog must implement. See *ConfigWidgetBase* (página 281) for a base class that implements this interface and defines various convenience methods as well.

changed_signal = None

This signal must be emitted whenever the user changes a value in this widget

supports_restoring_to_defaults = True

Set to True iff the `restore_to_defaults()` method is implemented.

restore_defaults_desc = 'Repões as configurações para os valores de origem. Terá de clicar em Aplicar para guardar efetivamente as definições predefinidas.'

The tooltip for the «Restore defaults» button

restart_critical = False

If True the Preferences dialog will not allow the user to set any more preferences. Only has effect if *commit()* (página 280) returns True.

genesis(*gui*)

Called once before the widget is displayed, should perform any necessary setup.

Parâmetros

gui – The main calibre graphical user interface

initialize()

Should set all config values to their initial values (the values stored in the config files). A «return» statement is optional. Return False if the dialog is not to be shown.

restore_defaults()

Should set all config values to their defaults.

commit()

Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an `AbortCommit` exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

refresh_gui(*gui*)

Called once after this widget is committed. Responsible for causing the gui to reread any changed settings. Note that by default the GUI re-initializes various elements anyway, so most widgets won't need to use this method.

initial_tab_changed()

Called if the initially displayed tab is changed before the widget is shown, but after it is initialized.

class calibre.gui2.preferences.**ConfigWidgetBase**(*parent=None*)

Base class that contains code to easily add standard config widgets like checkboxes, combo boxes, text fields and so on. See the *register()* (página 281) method.

This class automatically handles change notification, resetting to default, translation between gui objects and config objects, etc. for registered settings.

If your config widget inherits from this class but includes setting that are not registered, you should override the *ConfigWidgetInterface* (página 280) methods and call the base class methods inside the overrides.

changed_signal

This signal must be emitted whenever the user changes a value in this widget

supports_restoring_to_defaults = True

Set to True iff the `restore_to_defaults()` method is implemented.

restart_critical = False

If True the Preferences dialog will not allow the user to set any more preferences. Only has effect if *commit()* (página 281) returns True.

register(*name*, *config_obj*, *gui_name=None*, *choices=None*, *restart_required=False*, *empty_string_is_None=True*, *setting=<class 'calibre.gui2.preferences.Setting'>*)

Register a setting.

Parâmetros

- **name** – The setting name
- **config** – The config object that reads/writes the setting
- **gui_name** – The name of the GUI object that presents an interface to change the setting. By default it is assumed to be `'opt_' + name`.
- **choices** – If this setting is a multiple choice (combobox) based setting, the list of choices. The list is a list of two element tuples of the form: `[(gui name, value), ...]`
- **setting** – The class responsible for managing this setting. The default class handles almost all cases, so this param is rarely used.

initialize()

Should set all config values to their initial values (the values stored in the config files). A «return» statement is optional. Return False if the dialog is not to be shown.

commit(**args*)

Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an `AbortCommit` exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

restore_defaults(**args*)

Should set all config values to their defaults.

12.2 Variáveis de ambiente

- `CALIBRE_CONFIG_DIRECTORY` - sets the folder where configuration files are stored/read.
- `CALIBRE_TEMP_DIR` - define a pasta temporária a usar pelo Calibre
- `CALIBRE_CACHE_DIRECTORY` - define a pasta que o Calibre usará para armazenar dados que persistem entre as sessões
- `CALIBRE_OVERRIDE_DATABASE_PATH` - permite-lhe especificar o caminho completo para o ficheiro metadata.db. Usar esta variável permite-lhe ter o ficheiro metadata.db numa localização diferente da pasta onde se encontra a biblioteca. É útil se a sua pasta da biblioteca se encontra numa drive partilhada em rede que não tenha suporte para acesso exclusivo a ficheiros.
- `CALIBRE_DEVELOP_FROM` - used to run from a calibre development environment. See *Criar um ambiente de desenvolvimento para o Calibre* (página 337).
- `CALIBRE_OVERRIDE_LANG` - used to force the language used by the interface (ISO 639 language code)
- `CALIBRE_TEST_TRANSLATION` - used to test a translation .po file (should be the path to the .po file)
- `CALIBRE_NO_NATIVE_FILEDIALOGS` - causes calibre to not use native file dialogs for selecting files/folders.
- `CALIBRE_NO_NATIVE_MENUBAR` - causes calibre to not create a native (global) menu on Ubuntu Unity and similar Linux desktop environments. The menu is instead placed inside the window, as is traditional.
- `CALIBRE_USE_SYSTEM_THEME` - by default, on Linux, calibre uses its own builtin Qt style. This is to avoid crashes and hangs caused by incompatibilities between the version of Qt calibre is built against and the system Qt. The downside is that calibre may not follow the system look and feel. If you set this environment variable on Linux, it will cause calibre to use the system theme – beware of crashes and hangs.
- `CALIBRE_SHOW_DEPRECATION_WARNINGS` - causes calibre to print deprecation warnings to stdout. Useful for calibre developers.
- `CALIBRE_NO_DEFAULT_PROGRAMS` - prevent calibre from automatically registering the filetypes it is capable of handling with Windows.
- `QT_QPA_PLATFORM` - On Linux set this to `wayland` to force calibre to use Wayland and `xcb` to force use of X11.
- `SYSFS_PATH` - A usar se o sysfs está montado noutro local da árvore de ficheiros que não em /sys
- `http_proxy`, `https_proxy` - used on Linux to specify an HTTP(S) proxy

Veja Como definir variáveis de ambiente em Windows[106]. Se usa macOS pode definir variáveis de ambiente criando `~/Library/Preferences/calibre/macos-env.txt` e inserindo variáveis de ambiente, uma em cada linha, por exemplo:

```
CALIBRE_DEVELOP_FROM=$HOME/calibre-src/src
CALIBRE_NO_NATIVE_FILEDIALOGS=1
CALIBRE_CONFIG_DIRECTORY=~/.config/calibre
```

[106] https://www.computerhope.com/issues/ch000549.htm

12.3 Ajustes

Ajustes são pequenas alterações que pode especificar para controlar vários aspetos do comportamento do Calibre. Pode alterá-los indo a Preferências->Avançadas->Ajustes. Os valores padrão para os ajustes são reproduzidos abaixo

```
#!/usr/bin/env python
# vim:fileencoding=UTF-8:ts=4:sw=4:sta:et:sts=4:ai
# License: GPLv3 Copyright: 2010, Kovid Goyal <kovid at kovidgoyal.net>


# Contains various tweaks that affect calibre behavior. Only edit this file if
# you know what you are doing. If you delete this file, it will be recreated from
# defaults.

#: Auto increment series index
# The algorithm used to assign a book added to an existing series a series number.
# New series numbers assigned using this tweak are always integer values, except
# if a constant non-integer is specified.
# Possible values are:
#   next - First available integer larger than the largest existing number
#   first_free - First available integer larger than 0
#   next_free - First available integer larger than the smallest existing number
#   last_free - First available integer smaller than the largest existing number.␣
↪Return largest existing + 1 if no free number is found
#   const - Assign the number 1 always
#   no_change - Do not change the series index
#   a number - Assign that number always. The number is not in quotes. Note that 0.0␣
↪can be used here.
# Examples:
#   series_index_auto_increment = 'next'
#   series_index_auto_increment = 'next_free'
#   series_index_auto_increment = 16.5
#
# Set the use_series_auto_increment_tweak_when_importing tweak to True to
# use the above values when importing/adding books. If this tweak is set to
# False (the default) then the series number will be set to 1 if it is not
# explicitly set during the import. If set to True, then the
# series index will be set according to the series_index_auto_increment setting.
# Note that the use_series_auto_increment_tweak_when_importing tweak is used
# only when a value is not provided during import. If the importing regular
# expression produces a value for series_index, or if you are reading metadata
# from books and the import plugin produces a value, then that value will
# be used irrespective of the setting of the tweak.
series_index_auto_increment = 'next'
use_series_auto_increment_tweak_when_importing = False

#: Add separator after completing an author name
# Set this if the completion separator should be appended to the end of the
# completed text to automatically begin a new completion operation for authors.
# It can be either True or False
authors_completer_append_separator = False

#: Author sort name algorithm
# The algorithm used to copy author to author_sort.
# Possible values are:
#  invert: use "fn ln" -> "ln, fn"
#  copy  : copy author to author_sort without modification
```

(continues on next page)

(continuação da página anterior)

```
#  comma : use 'copy' if there is a ',' in the name, otherwise use 'invert'
#  nocomma : "fn ln" -> "ln fn" (without the comma)
# When this tweak is changed, the author_sort values stored with each author
# must be recomputed by right-clicking on an author in the left-hand tags
# panel, selecting 'Manage authors', and pressing
# 'Recalculate all author sort values'.
#
# The author_name_suffixes are words that are ignored when they occur at the
# end of an author name. The case of the suffix is ignored and trailing
# periods are automatically handled.
#
# The same is true for author_name_prefixes.
#
# The author_name_copywords are a set of words which, if they occur in an
# author name, cause the automatically generated author sort string to be
# identical to the author's name. This means that the sort for a string like
# "Acme Inc." will be "Acme Inc." instead of "Inc., Acme".
#
# If author_use_surname_prefixes is enabled, any of the words in
# author_surname_prefixes will be treated as a prefix to the surname, if they
# occur before the surname. So for example, "John von Neumann" would be sorted
# as "von Neumann, John" and not "Neumann, John von".
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')
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'
```

(continues on next page)

(continuação da página anterior)

```
#: Control partitioning of Tag browser
# When partitioning the Tag browser, the format of the subcategory label is
# controlled by a template: categories_collapsed_name_template if sorting by
# name, categories_collapsed_rating_template if sorting by average rating, and
# categories_collapsed_popularity_template if sorting by popularity. There are
# two variables available to the template: first and last. The variable 'first'
# is the initial item in the subcategory, and the variable 'last' is the final
# item in the subcategory. Both variables are 'objects'; they each have multiple
# values that are obtained by using a suffix. For example, first.name for an
# author category will be the name of the author. The sub-values available are:
#  name: the printable name of the item
#  count: the number of books that references this item
#  avg_rating: the average rating of all the books referencing this item
#  sort: the sort value. For authors, this is the author_sort for that author
#  category: the category (e.g., authors, series) that the item is in.
# Note that the "r'" in front of the { is necessary if there are backslashes
# (\ characters) in the template. It doesn't hurt anything to leave it there
# even if there aren't any backslashes.
categories_collapsed_name_template = r'{first.sort:shorten(4,,0)} - {last.
↪sort:shorten(4,,0)}'
categories_collapsed_rating_template = r'{first.avg_rating:4.2f:ifempty(0)} - {last.
↪avg_rating:4.2f:ifempty(0)}'
categories_collapsed_popularity_template = r'{first.count:d} - {last.count:d}'

#: Specify columns to sort the booklist by on startup
# Provide a set of columns to be sorted on when calibre starts.
# The argument is None if saved sort history is to be used
# otherwise it is a list of column,order pairs. Column is the
# lookup/search name, found using the tooltip for the column
# Order is 0 for ascending, 1 for descending.
# For example, set it to [('authors',0),('title',0)] to sort by
# title within authors.
sort_columns_at_startup = None

#: Control how dates are displayed
# Format to be used for publication date and the timestamp (date).
#  A string controlling how the publication date is displayed in the GUI
#  d     the day as number without a leading zero (1 to 31)
#  dd    the day as number with a leading zero (01 to 31)
#  ddd   the abbreviated localized day name (e.g. 'Mon' to 'Sun').
#  dddd  the long localized day name (e.g. 'Monday' to 'Sunday').
#  M     the month as number without a leading zero (1-12)
#  MM    the month as number with a leading zero (01-12)
#  MMM   the abbreviated localized month name (e.g. 'Jan' to 'Dec').
#  MMMM  the long localized month name (e.g. 'January' to 'December').
#  yy    the year as two digit number (00-99)
#  yyyy  the year as four digit number
#  h     the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm) '
#  hh    the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm) '
#  m     the minutes without a leading 0 (0 to 59) '
#  mm    the minutes with a leading 0 (00 to 59) '
#  s     the seconds without a leading 0 (0 to 59) '
#  ss    the seconds with a leading 0 (00 to 59) '
#  ap    use a 12-hour clock instead of a 24-hour clock, with "ap" replaced by the␣
↪localized string for am or pm
#  AP    use a 12-hour clock instead of a 24-hour clock, with "AP" replaced by the␣
```

(continues on next page)

(continuação da página anterior)

```
↪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
# 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).
```

(continues on next page)

(continuação da página anterior)

```
per_language_title_sort_articles = {
        # English
        'eng'  : (r'A\s+', r'The\s+', r'An\s+'),
        # Esperanto
        'epo': (r'La\s+', r"L'", 'L´'),
        # Spanish
        'spa'  : (r'El\s+', r'La\s+', r'Lo\s+', r'Los\s+', r'Las\s+', r'Un\s+',
                  r'Una\s+', r'Unos\s+', r'Unas\s+'),
        # French
        'fra'  : (r'Le\s+', r'La\s+', r"L'", u'L´', u'L’', r'Les\s+', r'Un\s+', r'Une\
↪s+',
                  r'Des\s+', r'De\s+La\s+', r'De\s+', r"D'", r'D´', r'D’'),
        # Polish
        'pol': (),
        # Italian
        'ita': ('Lo\\s+', 'Il\\s+', "L'", 'L´', 'La\\s+', 'Gli\\s+',
                'I\\s+', 'Le\\s+', 'Uno\\s+', 'Un\\s+', 'Una\\s+', "Un'",
                'Un´', 'Dei\\s+', 'Degli\\s+', 'Delle\\s+', 'Del\\s+',
                'Della\\s+', 'Dello\\s+', "Dell'", 'Dell´'),
        # Portuguese
        'por'  : (r'A\s+', r'O\s+', r'Os\s+', r'As\s+', r'Um\s+', r'Uns\s+',
                  r'Uma\s+', r'Umas\s+', ),
        # Romanian
        'ron'  : (r'Un\s+', r'O\s+', r'Nişte\s+', ),
        # German
        'deu'  : (r'Der\s+', r'Die\s+', r'Das\s+', r'Den\s+', r'Ein\s+',
                  r'Eine\s+', r'Einen\s+', r'Dem\s+', r'Des\s+', r'Einem\s+',
                  r'Eines\s+'),
        # Dutch
        'nld'  : (r'De\s+', r'Het\s+', r'Een\s+', r"'n\s+", r"'s\s+", r'Ene\s+',
                  r'Ener\s+', r'Enes\s+', r'Den\s+', r'Der\s+', r'Des\s+',
                  r"'t\s+"),
        # Swedish
        'swe'  : (r'En\s+', r'Ett\s+', r'Det\s+', r'Den\s+', r'De\s+', ),
        # Turkish
        'tur'  : (r'Bir\s+', ),
        # Afrikaans
        'afr'  : (r"'n\s+", r'Die\s+', ),
        # Greek
        'ell'  : (r'O\s+', r'I\s+', r'To\s+', r'Ta\s+', r'Tus\s+', r'Tis\s+',
                  r"'Enas\s+", r"'Mia\s+", r"'Ena\s+", r"'Enan\s+", ),
        # Hungarian
        'hun'  : (r'A\s+', r'Az\s+', r'Egy\s+',),
}
default_language_for_title_sort = None
title_sort_articles=r'^(A|The|An)\s+'

#: Specify a folder calibre should connect to at startup
# Specify a folder that calibre should connect to at startup using
# connect_to_folder. This must be a full path to the folder. If the folder does
# not exist when calibre starts, it is ignored.
# Example for Windows:
#     auto_connect_to_folder = 'C:/Users/someone/Desktop/testlib'
# Example for other operating systems:
#     auto_connect_to_folder = '/home/dropbox/My Dropbox/someone/library'
auto_connect_to_folder = ''
```

(continues on next page)

(continuação da página anterior)

```
#: 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:|(|)}'
#
# 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
```

(continues on next page)

(continuação da página anterior)

```
# 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
# 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, edit_cell,
# edit_metadata. Selecting anything other than open_viewer or show_book_details
```

(continues on next page)

(continuação da página anterior)

```
# 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
# 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
```

(continues on next page)

(continuação da página anterior)

```
# When calibre does a conversion from the same format to the same format, for
# example, from EPUB to EPUB, the original file is saved, so that in case the
# conversion is poor, you can tweak the settings and run it again. By setting
# this to False you can prevent calibre from saving the original file.
# Similarly, by setting save_original_format_when_polishing to False you can
# prevent calibre from saving the original file when polishing.
save_original_format = True
save_original_format_when_polishing = True

#: Number of recently viewed books to show
# Right-clicking the "View" button shows a list of recently viewed books. Control
# how many should be shown, here.
gui_view_history_size = 15

#: Change the font size of the Book details panel in the interface
# Change the font size at which book details are rendered in the side panel and
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0

#: What format to default to when using the "Unpack book" feature
# The "Unpack book" feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
# to that format rather than your output format preference.
# 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'
```

(continues on next page)

(continuação da página anterior)

```
#: Sort the list of libraries alphabetically
# The list of libraries in the Copy to library and Quick switch menus are
# normally sorted by most used. However, if there are more than a certain
# number of such libraries, the sorting becomes alphabetic. You can set that
# number here. The default is ten libraries.
many_libraries = 10

#: Choose available output formats for conversion
# Restrict the list of available output formats in the conversion dialogs.
# For example, if you only want to convert to EPUB and AZW3, change this to
# restrict_output_formats = ['EPUB', 'AZW3']. The default value of None causes
# all available output formats to be present.
restrict_output_formats = None

#: Set the thumbnail image quality used by the Content server
# The quality of a thumbnail is largely controlled by the compression quality
# used when creating it. Set this to a larger number to improve the quality.
# Note that the thumbnails get much larger with larger compression quality
# numbers.
# The value can be between 50 and 99
content_server_thumbnail_compression_quality = 75

#: Image file types to treat as e-books when dropping onto the "Book details" panel
# Normally, if you drop any image file in a format known to calibre onto the
# "Book details" panel, it will be used to set the cover. If you want to store
# some image types as e-books instead, you can set this tweak.
# 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
```

(continues on next page)

(continuação da página anterior)

```
# 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
```

12.4 Sobreposição a ícones, temas, etc.

Nota: calibre has direct support for icon themes, there are several icon themes available for calibre, that you can use by going to *Preferences → Interface → Look & Feel → Change icon theme*. It is preferable to use icon themes over overriding individual icons.

O Calibre permite-lhe alterar os recursos estáticos, tais como ícones, JavaScript, e modelos para os metadados da sobrecapa, catálogos, etc. por versões personalizadas de que goste. Todos os recursos estáticos estão armazenados na subpasta de recursos da pasta de instalação do Calibre. No Windows, esta situa-se habitualmente em `C:\Program Files\Calibre2\app\resources`. No macOS, `/Applications/calibre.app/Contents/Resources/resources/`. No Linux, se usou o binário de instalação retirado do sítio Web do Calibre, será `/opt/calibre/resources`. Estes caminhos podem ser diferentes, dependendo do modo como decidiu instalar o Calibre.

You should not change the files in this resources folder, as your changes will get overwritten the next time you update calibre. Instead, go to *Preferences → Advanced → Miscellaneous* and click *Open calibre configuration folder*. In this configuration folder, create a sub-folder called resources and place the files you want to override in it. Place the files in the appropriate sub folders, for example place images in `resources/images`, etc. calibre will automatically use your custom file in preference to the built-in one the next time it is started.

For example, if you wanted to change the icon for the *Remove books* action, you would first look in the built-in resources folder and see that the relevant file is `resources/images/remove_books.png`. Assuming you have an alternate icon in PNG format called `my_remove_books.png` you would save it in the configuration folder as `resources/images/remove_books.png`. All the icons used by the calibre user interface are in `resources/images` and its sub-folders. Placing an override file here will have even higher priority than a custom icon theme.

12.5 Criar o seu próprio tema de ícones para o Calibre

If you have created a beautiful set of icons and wish to share them with other calibre users via calibre's builtin icon theme support, you can easily package up your icons into a theme. To do so, go to *Preferences → Miscellaneous → Create icon theme*, select the folder where you have put your icons. Then fill up the theme metadata and click OK. This will result in a ZIP file containing the theme icons. You can upload that to the calibre forum at Mobileread[107] and then I will make your theme available via calibre's builtin icon theme system. By default, the icon theme you just created will also be installed as the current theme in calibre. If you are testing your theme, remember to remove the images from the `resources/images` folder so that the icons from the theme are used.

As of calibre 6, you can have custom icons for light and dark mode. Simply create two versions of the icon and name the files with the suffix `-for-dark-theme` and `-for-light-theme`. For example, `modified-for-dark-theme.png` and `modified-for-light-theme.png`. Then calibre will automatically use the appropriate icon based on the current theme.

12.6 Personalizar Calibre com plugins

O Calibre possui um design muito modular. Quase todas as funcionalidades do Calibre existem na forma de plugins. Os plugins são usados para conversão, para descarregar notícias (embora sejam chamadas de receitas), para vários componentes da interface de utilizador, para se ligar a dispositivos diferentes, para processar ficheiros ao adicioná-los ao Calibre e assim por diante. Pode obter uma lista completa de todos os plugins embutidos no Calibre, aceda a *Preferências → Avançado → Plugins*.

Pode escrever os seus próprios plugins para personalizar e expandir o comportamento do Calibre. A arquitetura de plugins do Calibre é muito simples, como pode verificar no tutorial *Criar os seus próprios plugins para aumentar as funcionalidades do Calibre* (página 219).

Once you have written a plugin, you can upload that to the calibre plugins forum at Mobileread[108] and it will be made available via calibre's builtin plugin updater.

[107] https://www.mobileread.com/forums/forumdisplay.php?f=166
[108] https://www.mobileread.com/forums/forumdisplay.php?f=237

CAPÍTULO 13

Interface de linha de comandos

Nota: No macOS, as ferramentas de linha de comando estão dentro do pacote do Calibre. Por exemplo, se instalou o Calibre em `/Applications` as ferramentas de linha de comando estão em `/Applications/calibre.app/Contents/MacOS/`.

13.1 Comandos documentados

13.1.1 `calibre`

```
calibre [opções] [caminho_para_livro ou url do Calibre ...]
```

Inicia a interface gráfica principal do Calibre e, opcionalmente, adiciona à base de dados o livro existente em caminho_para_livro. Também pode especificar alguns URL do Calibre para realizar várias ações diferentes, e não apenas adicionar livros. Por exemplo:

calibre://view-book/biblioteca_teste/1842/epub

Abrirá, no visualizador de ebooks do Calibre, o livro com o iD 1842, no formato EPUB, da biblioteca «biblioteca_teste». Os nomes de biblioteca são os nomes das pastas das bibliotecas com os espaços substituídos por sobtraço. Uma descrição completa das várias ações baseadas em URL está disponível no Manual do utilizador.

Sempre que passar argumentos para o **calibre** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--detach
Se existir, desligue do terminal de controlo (apenas para Linux)

--help, -h
mostrar esta mensagem de ajuda e sair

--ignore-plugins
Ignorar plugins personalizados. Útil se instalou algum plugin que esteja a impedir o arranque do Calibre

--no-update-check
Não verificar por atualizações

--shutdown-running-calibre, -s
Causa a terminação de uma instância do Calibre, caso exista alguma em execução. Note que se existirem tarefas em execução estas serão silenciosamente canceladas pelo que deve usar esta funcionalidade com cuidado.

--start-in-tray
Iniciar minimizado na área de notificação.

--verbose, -v
Ignorado, não utilizar. Presente apenas por retrocompatibilidade

--version
mostrar o número da versão do programa e sair

--with-library
Usar a biblioteca localizada no caminho especificado.

13.1.2 calibre-customize

```
Opções calibre-customize
```

Personalize o Calibre carregando plugins externos.

Sempre que passar argumentos para o **calibre-customize** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--add-plugin, -a
Adicione um plugin especificando um caminho para o ficheiro Zip que o contenha.

--build-plugin, -b
Para programadores de plugins: caminho para a pasta onde está a criar o plugin. Este comando irá criar o zip automaticamente e atualizar o plugin no Calibre.

--customize-plugin
Personalizar o plugin. Especifique o nome e uma expressão identificadora, separados por uma vírgula.

--disable-plugin
Desativar o plugin mencionado

--enable-plugin

Ativar o plugin mencionado

--help, -h

mostrar esta mensagem de ajuda e sair

--list-plugins, -l

Listar todos os plugins instalados

--remove-plugin, -r

Remover um plugin personalizado pelo seu nome. Não tem qualquer efeito sobre os plugins integrados.

--version

mostrar o número da versão do programa e sair

13.1.3 calibre-debug

```
calibre-debug [opções]
```

Várias interfaces de linhas de comando úteis para depurar o Calibre. Executado sem opções, este comando inicia um interpretador de Python. Pode também executar a interface de utilizador principal, o visualizador de ebooks e o editor Calibre em modo depuração.

Contém também as interfaces para várias partes do Calibre que não têm ferramentas de linha de comando dedicadas, tais como subconjuntos de fontes, ferramenta de diferenças entre livros digitais e outras.

Pode ainda utilizar **calibre-debug** para executar scripts em modo independente. Para tal, corra o seguinte:

calibre-debug -e myscript.py -- --option1 --option2 file1 file2 ...

Tudo após -- será passado ao script como parâmetro. Também pode usar **calibre-debug** como uma linha Shebang em scripts, tal como:

#!/usr/bin/env -S **calibre-debug** -e -- –

Sempre que passar argumentos para o **calibre-debug** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--add-simple-plugin

Adiciona um plugin simples (isto é, um plugin composto apenas de um ficheiro .py) especificando o caminho para o ficheiro py que contém o código do plugin.

--command, -c

Executar código Python.

--debug-device-driver, -d

Depurar deteção de dispositivos

--default-programs

Retirar o Calibre do registo de programas predefinidos do Windows *--default-programs* (página 297) = (register|unregister)

--diff

Executar a ferramenta Diff do Calibre. Por exemplo: Calibre-debug *--diff* (página 297) -- file1 file2

--edit-book

Iniciar a ferramenta "Editar livro" em modo de depuração.

--exec-file, -e

Executar o código Python no ficheiro.

--explode-book, -x

Desconstruir o livro para a pasta especificada. Uso: -x file.epub output_dir Exporta o livro sob a forma de uma coleção de ficheiros HTML e metadados, que poderá editar utilizando ferramentas adequadas à edição de HTML. Aplicável a ficheiros no formato EPUB, AZW3, HTMLZ e DOCX.

--export-all-calibre-data

Export all calibre data (books/settings/plugins). Normally, you will be asked for the export folder and the libraries to export. You can also specify them as command line arguments to skip the questions. Use absolute paths for the export folder and libraries. The special keyword "all" can be used to export all libraries. Examples: calibre-debug *--export-all-calibre-data* (página 298) # for interactive use calibre-debug *--export-all-calibre-data* (página 298) /path/to/empty/export/folder /path/to/library/folder1 /path/to/library2 calibre-debug *--export-all-calibre-data* (página 298) /export/folder all # export all known libraries

--fix-multiprocessing

Para uso interno

--gui, -g

Executar a interface de utilizador em modo de depuração. A informação da depuração será escrita em stdout e stderr.

--gui-debug

Executar a interface de utilizador com uma consola de depuração, gravando registos na localização especificada. Apenas para uso interno, use a opção -g para executar a interface de utilizador em modo de depuração

--help, -h

mostrar esta mensagem de ajuda e sair

--implode-book, -i

Implodir um livro previamente desconstruído. Uso: -i output_dir file.epub Importa o livro a partir dos ficheiros que se encontram na output_dir que foram criados por uma invocação prévia de *--explode-book* (página 298). Assegure-se de que especifica o mesmo tipo de ficheiro que usou aquando da desconstrução (explosão).

--import-calibre-data

Importar dados previamente exportados do Calibre

--inspect-mobi, -m

Inspecionar ficheiro(s) MOBI nas localizações especificadas

--paths

Mostra as localizações de sistema necessárias para configurar o ambiente do Calibre

--run-plugin, -r

Executa um plugin que fornece uma interface de linha de comandos. Por exemplo: calibre-debug -r "Plugin name" -- file1 --option1 Tudo o que surge após -- será passado para o plugin como argumentos.

--run-test, -t

Executar o(s) teste(s) nomeado(s). Use o valor especial "all" para executar todos os testes. Caso o nome do teste comece com um ponto, será considerado como um nome de um módulo. Caso o nome do teste comece com @, será considerado como um nome de uma categoria.

--run-without-debug

Não executar com o sinalizador DEBUG definido

--shutdown-running-calibre, -s

Causa a terminação de uma instância do Calibre, caso exista alguma em execução. Note que se existirem tarefas em execução estas serão silenciosamente canceladas pelo que deve usar esta funcionalidade com cuidado.

--subset-font, -f

Subdividir o tipo de letra especificado. Use −− depois desta opção para passar a opção pelo programa de subdivisão de tipos de letra.

--test-build

Testar módulos binário na produção

--version

mostrar o número da versão do programa e sair

--viewer, -w

Executar o visualizador de livros digitais no modo de depuração

13.1.4 calibre-server

```
calibre-server [options] [path to library folder ...]
```

Inicie o servidor de conteúdos Calibre. O servidor de conteúdos do Calibre expõe a sua biblioteca Calibre na Internet. Você pode especificar o caminho para a pasta da biblioteca com os argumentos para **calibre-server**. Se não especificar qualquer caminho, serão utilizadas todas as bibliotecas que o programa principal Calibre conheça.

Sempre que passar argumentos para o **calibre-server** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--access-log

Caminho para o ficheiro de registo de acessos. Este registo contém informações sobre os clientes que se ligam e fazem pedidos ao servidor. Por predefinição, nenhum registo de acesso é efetuado.

--ajax-timeout

O tempo em segundos para aguardar por uma resposta do servidor ao efetuar pedidos.

--auth-mode

Escolher o tipo de autenticação a usar. Define o modo de autenticação HTTP usado pelo servidor. Definir como "basic" se usar o servidor atrás de um proxy SSL. Senão, deixe como "auto", que usará "basic" se o SSL estiver configurado, ou "digest" se não estiver configurado.

--auto-reload

Recarregar o servidor automaticamente quando houver mudanças no código fonte. Útil para desenvolvimento. Você também deve especificar um valor pequeno para o tempo limite de encerramento do servidor.

--ban-after

Número de erros de autenticação até ser banido. Número de erros de autenticação até que um endereço de IP seja banido

--ban-for

Banir endereços IP que apresentem erros de autenticação continuados. Banir temporariamente o acesso a endereços IP que tenham falhas de autenticação repetidas no número de minutos especificado. Útil para evitar tentativas de adivinhar senhas. Se definido como zero, nenhum banimento será efetuado.

--book-list-mode

Escolher o modo de listagem de livros predefinido. Define o modo de lista de livros predefinida que será utilizada para novos utilizadores. Utilizadores individuais podem sobrepor-se ao modo predefinido nas suas próprias configurações. O predefinido é utilizar uma grelha de capas.

--compress-min-size

Tamanho mínimo para que a resposta utilize compressão de dados (em bytes).

--custom-list-template

Caminho para um ficheiro JSON contendo um modelo para o modo de lista de livros personalizados. A maneira mais fácil de criar esse ficheiro de modelo é ir a Preferências-> Partilha na rede-> Modelo para lista de livros, criar o modelo e exportá-lo.

--daemonize

Executar processo em segundo plano como um Daemon (apenas para Linux).

--displayed-fields

Restringir exibição de campos de metadados definidos pelo utilizador. Lista separada por vírgulas de campos de metadados personalizados que serão exibidos pelo servidor de conteúdos nas saídas de /opds e /mobile. Se especificar esta opção, qualquer campo que não esteja na lista não será exibido. Por exemplo: my_rating,my_tags

--enable-allow-socket-preallocation, --disable-allow-socket-preallocation

Pré-alocação de encaixe, por exemplo, com ativação de encaixe systemd. Por predefinição, a opção esta ativada.

--enable-auth, --disable-auth

Autenticação por senha para aceder ao servidor. Por predefinição, o servidor não tem restrições, permitindo que qualquer pessoa lhe possa aceder. Pode restringir o acesso a utilizadores predefinidos com esta opção. Por predefinição, a opção esta desativada.

--enable-fallback-to-detected-interface,
--disable-fallback-to-detected-interface

Substituir pela interface detetada automaticamente. Se. por algum motivo, o servidor não conseguir associar a interface especificada na opção listen_on, ele tentará detetar uma interface que se ligue ao mundo exterior e associar-se a ela. Por predefinição, a opção esta ativada.

--enable-local-write, --disable-local-write

Permitir que ligações locais não autenticadas realizem alterações. Normalmente, se não ativar a autenticação o servidor operará no modo de apenas leitura, para não permitir que utilizadores anónimos façam alterações nas suas bibliotecas Calibre. Essa opção permite que qualquer pessoa que se ligue no mesmo computador em que o servidor está a ser executado possa fazer alterações. Isto é útil se deseja executar o servidor sem autenticação, mas ainda usar o calibredb para fazer alterações nas suas bibliotecas Calibre. Note que ativar esta opção significa que qualquer programa em execução no computador também pode fazer alterações nas suas bibliotecas Calibre. Por predefinição, a opção esta desativada.

--enable-log-not-found, --disable-log-not-found

Registar pedidos HTTP 404 (Não encontrado). Por predefinição, o servidor regista todos os pedidos HTTP relativos a recursos que não são encontrados. Isto pode gerar muitos registos indesejados se o seu servidor for alvo de Bots. Utilize esta opção para o desativar. Por predefinição, a opção esta ativada.

--enable-use-bonjour, --disable-use-bonjour

Divulgar sinal OPDS através de BonJour. Anuncia os sinais OPDS através do serviço BonJour, de modo que as

aplicações de leitura baseadas em OPDS possam detetar e ligar-se ao servidor automaticamente. Por predefinição, a opção esta ativada.

--enable-use-sendfile, --disable-use-sendfile

Cópia zero na transferência de ficheiros para melhor desempenho. Isto usará transferências zero-copy no Kernel ao enviar ficheiros pela rede, aumentado a performance. Entretanto, pode causar transferências corrompidas de ficheiros em alguns sistemas de ficheiros com problemas. Desative se verificar transferências de corrompidas. Por predefinição, a opção esta ativada.

--help, -h

mostrar esta mensagem de ajuda e sair

--ignored-fields

Campos de metadados personalizados a ignorar. Lista separada por vírgulas de campos de metadados personalizados que não serão exibidos pelo servidor de conteúdos nas saídas de /opds e /mobile. Por exemplo: my_rating,my_tags

--listen-on

A interface sob a qual escutar ligações. Por predefinição, irá escutar em todas as interfaces IPv4 disponíveis. Pode mudar isto para, por exemplo, `'127.0.0.1'` e assim escutar apenas as ligações da máquina local, ou alterar para `'::'` e escutar todas as ligações de entrada IPv6 e IPv4.

--log

Caminho para o ficheiro de registos do servidor. Este ficheiro contém informações e erros do servidor, não contém informações de acesso. Por predefinição é escrito no stdout.

--manage-users

Gere a base de dados de utilizadores com permissão para ligar a este servidor. Pode usar de modo automático ao adicionar um –. Ver calibre-server *--manage-users* (página 301) -- help para detalhes. Veja também a opção *--userdb* (página 302).

--max-header-line-size

Tamanho máximo de cabeçalho HTTP individual (em KB).

--max-job-time

Tempo máximo para processos de trabalho. Quantidade máxima de tempo permitida para a execução dos processos de trabalho (em minutos). Defina para zero para não ter limite.

--max-jobs

Número máximo de processos de trabalho. As aplicações de processos de trabalho são iniciadas conforme necessário e usadas em grandes trabalhos, como preparar um livro para visualização, adicionar livros, converter etc. Normalmente, o número máx. de tais processos é baseado no número de núcleos de CPU. Você pode controlar esse número nesta configuração.

--max-log-size

Tamanho máx. do ficheiro log. (em MB). Tamanho máximo dos ficheiros de diário gerado pelo servidor. Quando o registo diário se tornar maior que este tamanho, será feita uma rotação automática. Defina para zero para desativar rotação de diário.

--max-opds-items

Número máximo de livros no sinal OPDS. Número máximo de livros que o servidor fornece por cada sinal de aquisição de OPDS único.

--max-opds-ungrouped-items

Número máximo de itens não agrupados em sinais OPDS. Agrupar itens em categorias tais como autores/etiquetas pela primeira lera quando existir mais que este número de itens. Escolha 0 para desativar.

--max-request-body-size

Tamanho máximo para ficheiros a carregar para o servidor (em MB).

--num-per-page

Número de livros a mostrar numa só pagina. O número de livros a mostrar numa só pagina no navegador.

--pidfile

Escreva o processo com o PID para o ficheiro especificado

--port

A porta onde escutar ligações.

--search-the-net-urls

Caminho para um ficheiro JSON contendo URLs para a funcionalidade "Pesquisar na internet". A maneira mais fácil de criar esse ficheiro é ir a Preferências-> Partilhar na rede->Pesquisar internet no Calibre, criar os URL e exportá-los.

--shutdown-timeout

Número de segundos de espera para um encerrar limpo.

--ssl-certfile

Caminho para o ficheiro do certificado SSL.

--ssl-keyfile

Caminho para o ficheiro da chave privada SSL.

--timeout

O tempo em segundos após o qual uma ligação inativa será encerrada.

--trusted-ips

Permitir que ligações não autenticadas, mas de endereços IP específicos, realizem alterações. Normalmente, se não ativar a autenticação o servidor operará no modo de apenas leitura, para não permitir que utilizadores anónimos façam alterações nas suas bibliotecas Calibre. Essa opção permite que qualquer pessoa que se ligue a partir do endereço IP especificado possa fazer alterações. Deve ser uma lista separada por vírgulas de endereços ou especificações de redes. Isto é útil se deseja executar o servidor sem autenticação, mas ainda usar o calibredb para fazer alterações nas suas bibliotecas Calibre. Note que ativar esta opção significa que qualquer pessoa que se ligue a partir dos endereços de IP especificados pode fazer alterações nas suas bibliotecas Calibre.

--url-prefix

Prefixo a anteceder todos os URL. Útil se você deseja rodar este servidor por trás de um proxy reverso. Por exemplo, use /calibre como prefixo de URL

--userdb

Caminho para a base de dados de utilizadores a usar para autenticação. A base de dados é um ficheiro SQLite. Para o criar use *--manage-users* (página 301). Pode ler mais sobre como gerir utilizadores em: https://manual.calibre-ebook.com/pt/server.html#managing-user-accounts-from-the-command-line-only

--version

mostrar o número da versão do programa e sair

--worker-count

Número máximo de processos de trabalho para executar pedidos.

13.1.5 calibre-smtp

```
calibre-smtp [options] [from to text]
```

Enviar email usando o protocolo SMTP. **calibre-smtp** tem dois modos de operação. No modo de composição você especifica destinatário (from) e texto (text)e estes são usados para construir e enviar uma mensagem de email. No modo de filtro, **calibre-smtp** lê uma mensagem completa de email do STDIN e envia-o.

TEXT é o corpo da mensagem de email. Se o texto não for especificado, uma mensagem completa de email é lida de STDIN. FROM (De) é o endereço de email do remetente e TO (para) é o endereço de email do destinatário. Quando um email completo é lido de STDIN, de e para são usados apenas na negociação SMTP, sendo que os cabeçalhos das mensagens não são modificados.

Sempre que passar argumentos para o **calibre-smtp** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--fork, -f
: Criar um processo separado e entregar a mensagem nesse processo de segundo plano. Se usar esta opção, deve também usar *--outbox* (página 303) para lidar com as eventuais falhas de entrega.

--help, -h
: mostrar esta mensagem de ajuda e sair

--localhost, -l
: Nome do anfitrião local. Usado quando se efetua a ligação ao servidor de SMTP.

--outbox, -o
: Caminho para o diretório maildir onde são armazenadas as mensagens de email que falharam.

--timeout, -t
: Ultrapassado o tempo para a ligação

--verbose, -v
: Seja mais detalhado

--version
: mostrar o número da versão do programa e sair

COMPOSE MAIL

Opções para compor um email. Ignorado se o texto não for especificado

--attachment, -a
: Ficheiro para anexar ao email

--subject, -s
: Assunto do email

SMTP RELAY

Opções para usar um servidor de SMTP intermediário para enviar correio eletrónico. O Calibre tenta enviar o email diretamente a menos que a opção –relay seja especificada.

--cafile

Caminho para um ficheiro contendo certificados CA concatenados no formato PEM. Usado para verificar o certificado do servidor quando se usa TLS. Por predefinição é usado o certificado CA do sistema.

--dont-verify-server-certificate

Não efetuar a verificação do certificado do servidor quando se usa ligações com TLS. Tal costumava ser o comportamento por predefinição nas versões do Calibre anteriores à 3.27. Se estiver a usar um servidor intermediário que tenha um certificado auto-assinado ou inválido por qualquer outra razão, pode usar esta opção para obter o comportamento habitual das versões anteriores à 3.27

--encryption-method, -e

Método de encriptação a usar aquando da ligação ao intermediário. Escolhas possíveis são TLS, SSL e NONE. A predefinição é o uso de TLS. Escolher NONE é extremamente inseguro.

--password, -p

Senha para retransmissão

--port

Porta TCP para ligação ao servidor intermediário. Por predefinição é usada a porta 465 se o método de encriptação for SSL, ou a porta 25 nos restantes casos.

--relay, -r

Um servidor intermediário de SMTP usado para enviar correio eletrónico.

--username, -u

Nome de utilizador para retransmissão

13.1.6 calibredb

```
calibredb comando [options] [arguments]
```

calibredb é a interface de linha de comandos para a base de dados do Calibre. Possui vários subcomandos, documentados a seguir.

calibredb pode ser usado para manipular quer uma base de dados Calibre especificada por caminho quer um *Servidor de conteúdos* Calibre em execução em máquina local ou na Internet. Você pode iniciar um :guilabel: *Servidor de conteúdos* Calibre usando o programa **calibre-server** ou, no programa principal Calibre, clicar em *Ligar/Partilhar -> Iniciar servidor de conteúdos`*. Como o **calibredb** pode efetuar alterações nas suas bibliotecas Calibre, primeiro você deve configurar a autenticação no servidor. Existem duas maneiras de o fazer:

- Se planeia ligar-se apenas a um servidor a funcionar no mesmo computador, você pode simplesmente usar a opção `--enable-local-write` do servidor de conteúdos, para permitir que qualquer programa que seja executado no computador local, incluindo o calibredb, possa fazer alterações nos seus dados Calibre. Ao executar o servidor a partir do programa principal do Calibre, esta opção está em *Preferências → Partilhar na rede → Avançado*.
- Se deseja ativar o acesso pela internet então deve configurar contas de utilizador no servidor e usar as opções :option: *–username* e :option:` –password` para :command: *calibredb* para fornecer acesso.

Você pode configurar a autenticação de utilizadores para :command: *calibre-server* usando a opção `--enable-auth` e usando `--manage-users` para criar as contas de utilizador. Se estiver a executar o servidor a partir do programa Calibre principal, use *Preferências → Partilhar na rede → Exigir nome de utilizador / senha*.

Para ligar a um servidor de conteúdo em execução, passe a URL do servidor para a opção :option: *–with-library*. Veja a documentação dessa opção para detalhes e exemplos.

- *OPÇÕES GLOBAIS* (página 306)
- *list* (página 306)
- *add* (página 307)
  - *Adicionar a partir de pastas* (página 308)
- *remove* (página 309)
- *add_format* (página 309)
- *remove_format* (página 309)
- *show_metadata* (página 309)
- *set_metadata* (página 310)
- *export* (página 310)
- *catalog* (página 311)
  - *EPUB OPÇÕES* (página 312)
- *saved_searches* (página 313)
- *add_custom_column* (página 314)
- *custom_columns* (página 314)
- *remove_custom_column* (página 314)
- *set_custom* (página 315)
- *restore_database* (página 315)
- *check_library* (página 315)
- *list_categories* (página 316)
- *backup_metadata* (página 316)
- *clone* (página 316)
- *embed_metadata* (página 317)
- *search* (página 317)
- *fts_index* (página 317)
- *fts_search* (página 318)

OPÇÕES GLOBAIS

--help, -h

mostrar esta mensagem de ajuda e sair

--library-path, --with-library

Caminho para a biblioteca do Calibre. Por predefinição é usado o caminho armazenado nas definições. Pode também ligar-se a um servidor de conteúdos Calibre para executar ações em bibliotecas remotas. Para tal, use um URL na forma: http://hostname:port/#id_da_biblioteca , por exemplo, http://localhost:8080/#mylibrary . id_da_biblioteca é o identificador da biblioteca à qual se pretende ligar no servidor de conteúdos. Pode usar o valor especial de id_da_bilbioteca - para obter uma lista de identificadores de bibliotecas disponíveis no servidor. Para detalhes sobre como configurar o acesso através de um servidor de conteúdos, veja https://manual.calibre-ebook.com/pt/generated/pt/calibredb.html.

--password

Senha para se ligar a um servidor de conteúdos Calibre. Para ler a senha a partir da entrada padrão, use o valor especial: <stdin>. Para ler a senha de um ficheiro use: <f:/path/to/file> (isto é: <f: followed by the full path to the file and a trailing >). Os parênteses angulares acima são obrigatórios, lembre-se de os escapar ou use aspas para a sua Shell.

--timeout

O tempo limite, em segundos, para se ligar a uma biblioteca Calibre pela rede. A predefinição é de dois minutos.

--username

Nome de utilizador para ligar ao servidor de conteúdos do Calibre

--version

mostrar o número da versão do programa e sair

list

```
calibredb list [options]
```

Lista de livros disponíveis na base de dados do Calibre.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--ascending

Ordenar os resultados por ordem crescente

--fields, -f

Os campos a apresentar ao listar os livros na base dados. Deve ser uma lista de campos separados por virgulas. Campos disponíveis: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid A predefinição é: title,authors. O campo especial "all" pode ser utilizado para selecionar todos os campos. Para além dos campos internos acima, os campos personalizados também estão disponíveis como *field_name, por exemplo, para o campo personalizado #rating, use o nome: *rating

--for-machine

Gerar ficheiro de saída no formato JSON que é mais adequado para análise de máquina. Isto fará com que a largura de linha e as opções de separador sejam ignorados.

--limit

O número máximo de resultados a mostrar. Predefinição: todos

--line-width, -w

A largura máxima de uma linha no ficheiro de destino. A predefinição será detetar o tamanho do ecrã.

--prefix

O prefixo para todos os caminhos para os ficheiros. A predefinição é usar o caminho absoluto para a pasta da biblioteca.

--search, -s

Filtrar resultados pela expressão de pesquisa. Sobre o formato da expressão de pesquisa, veja a documentação relacionada com as pesquisas no Manual do utilizador. A predefinição é não filtrar.

--separator

A expressão utilizada para separar os campos. A predefinição é um espaço.

--sort-by

O campo pelo qual ordenar os resultados. Pode especificar múltiplos campos separando-os por vírgulas. Campos disponíveis: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid Predefinição: id

--template

The template to run if "template" is in the field list. Default: None

--template_file, -t

Path to a file containing the template to run if "template" is in the field list. Default: None

--template_heading

Heading for the template column. Default: template. This option is ignored if the option *--for-machine* (página 306) is set

add

```
calibredb add [options] ficheiro1 ficheiro2 ficheiro3 ...
```

Adicionar os ficheiros especificados à base de dados. Também pode especificar pastas: ver abaixo as opções relativas às pastas.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--authors, -a

Definir o autor do(s) livro(s) adicionado(s)

--automerge, -m

Se forem encontrados livros com títulos e autores semelhantes, fundir automaticamente os formatos (ficheiros) a receber nos registos de livros existentes. Um valor de "ignore" (Ignorar) significa que os formatos duplicados são descartados. Um valor "overwrite" (Sobrepor) significa que os formatos duplicados na biblioteca são substituídos pelos ficheiros recém-adicionados. Um valor de "new_record" (Novo registo) significa que os formatos duplicados são colocados num novo registo de livro.

--cover, -c

Caminho para a capa utilizar no livro adicionado

--duplicates, -d

Adicionar livros à base de dados mesmo se estes já existem. A comparação é feita baseada nos títulos e autores dos livros. Note que a opção *--automerge* (página 307) tem precedência.

--empty, -e

Adicionar livro vazio (um registo sem qualquer formato)

--identifier, -I

Definir identificadores para este livro, por exemplo -I asin:XXX -I isbn:YYY

--isbn, -i

Definir o ISBN do(s) livro(s) adicionado(s)

--languages, -l

Uma lista de idiomas separada por vírgulas (é melhor utilizar códigos de idioma ISO639, apesar de alguns nomes de idiomas poderem ser reconhecidos)

--series, -s

Definir as séries do(s) livro(s) adicionado(s)

--series-index, -S

Definir o numero na série do(s) livro(s) adicionado(s)

--tags, -T

Definir as etiquetas do(s) livro(s) adicionado(s)

--title, -t

Definir o título do(s) livro(s) adicionado(s)

Adicionar a partir de pastas

Opções para controlar a inclusão de livros a partir de pastas. Por predefinição só serão adicionados ficheiros com extensões conhecidas de formatos de livros.

--add

Um padrão (glob) de nomes de ficheiros, em que os ficheiros que correspondam a este padrão serão adicionados quando se varrem pastas na pesquisa de ficheiros, mesmo que não sejam de um tipo de ficheiro conhecido como ebook. Pode ser especificado múltiplas vezes para múltiplos padrões.

--ignore

Um padrão (glob) de nomes de ficheiros, em que os ficheiros que correspondam a este padrão serão ignorados quando se varrem pastas na pesquisa de ficheiros. Pode ser especificado múltiplas vezes para múltiplos padrões. Exemplo: *.pdf ignorará todos os ficheiros pdf

--one-book-per-directory, -1

Assumir que cada pasta tem um único livro lógico e que todos os ficheiros nela contidos são formatos diferentes desse livro

--recurse, -r

Processar pastas recursivamente

remove

```
calibredb remover ids
```

Remover os livros, identificados por ID, da base de dados. Os ID devem estar numa lista de números de ID, separados por vírgulas. Você pode obter o numero de ID através do comando de pesquisa. Por exemplo, 23,34,57-85 (se especificar um intervalo, o último número do intervalo não será incluído).

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--permanent

Não usar a Reciclagem

add_format

```
calibredb add_format [options] ID ebook_file
```

Adicionar o livro em ebook_file para os formatos disponíveis para o livro lógico identificado por ID. Pode obter o ID usando o comando de pesquisa. Caso o formato já exista será substituído, a menos que esteja especificada a opção de não substituir.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--dont-replace

Não substituir o formato se o formato já existir

remove_format

```
calibredb remove_format [options] id fmt
```

Remover o formato fmt do livro lógico identificado pelo ID. Pode saber o ID usando o comando search. fmt deve ser uma extensão de ficheiro como LRF ou TXT ou EPUB. Se o livro lógico não tem fmt disponível, não fazer nada.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

show_metadata

```
calibredb show_metadata [options] id
```

Mostrar os metadados armazenados na base de dados do Calibre para o livro identificado pelo ID. O ID é um número de identificação do comando de pesquisa.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--as-opf

Imprimir os metadados no formulário OPF (XML)

set_metadata

```
calibredb set_metadata [options] book_id [/path/to/metadata.opf]
```

Definir os metadados armazenados na base de dados Calibre para o livro identificado por book_id do ficheiro OPF metadata.opf. book_id é um número de identificação do comando de pesquisa. Pode ter uma ideia do formato OPF utilizando a opção –as-opf alternando para o comando show_metadata. Também pode definir os metadados de campos individuais com a opção –field. Se usar a opção –field, não é necessário especificar um ficheiro OPF.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--field, -f

O campo a ser definido. O formato é field_name:value, por exemplo: *--field* (página 310) tags:tag1,tag2. Use *--list-fields* (página 310) para obter uma lista de todos os nomes de campo. Pode especificar esta opção várias vezes para definir vários campos. Nota: para idiomas deve usar os códigos de idioma ISO639 (por exemplo, en para Inglês, pt para Português e assim por diante). Para identificadores, a sintaxe é *--field* (página 310) identifiers:isbn:XXXX,doi:YYYYY. Para campos booleanos (sim/não) use verdadeiro e falso ou sim e não.

--list-fields, -l

Lista de nomes de campos de metadados que podem ser usadas com a opção *--field* (página 310)

export

```
calibredb exportar [options] ID
```

Exporta os livros especificados pelos ID (numa lista separada por virgulas) para o sistema de ficheiros. A operação de exportação guarda todos os formatos do livro, a sua capa e os metadados (num ficheiro OPF). Pode saber os números de ID através do comando de pesquisa.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--all

Exportar todos os livros da base de dados, ignorando a lista de ID.

--dont-asciiize

Converter com o Calibre todos os caracteres não ingleses para equivalentes ingleses nos nomes dos ficheiros. Isto é útil se estiver a guardar para um sistema de ficheiros antigo que não possua suporte total Unicode a nomes de ficheiros. Especificar este parâmetro irá desligar este comportamento.

--dont-save-cover

Normalmente o Calibre guarda a capa num ficheiro separado, junto aos ficheiros do livro em causa. Especificar este parâmetro irá desligar este comportamento.

--dont-update-metadata

Normalmente o Calibre atualiza os metadados nos ficheiros exportados a partir da biblioteca do Calibre. Torna a exportação de ficheiros mais lenta. Especificar este parâmetro irá desligar este comportamento.

--dont-write-opf

Normalmente o Calibre escreve os metadados num ficheiro OPF separado, junto aos ficheiros do livro em causa. Especificar este parâmetro irá desligar este comportamento.

--formats

Lista separada por vírgulas de de formatos a guardar para cada livro. A predefinição todos os formatos disponíveis são guardados.

--progress

Relatório de execução

--replace-whitespace

Substituir espaços por sobtraços

--single-dir

Exportar todos os livros para uma única pasta

--template

Modelo para controlar a estrutura do nome de ficheiro e pastas para os ficheiros exportados. A predefinição é "{author_sort}/{title}/{title} - {authors}", o que vai guardar os livros numa subpasta por autor com o nome dos ficheiros contendo título e autor. As variáveis disponíveis são: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt

O formato para exibir datas. %d - dia, %b - mês, %m - número do mês, %Y - ano. A predefinição é: %b, %Y

--to-dir

Exportar livros para a pasta especificada. O predefinido é .

--to-lowercase

Converter os caminhos para minúsculas.

catalog

```
calibredb catalog /path/to/destination.(csv|epub|mobi|xml...) [options]
```

Export a **catalog** in format specified by path/to/destination extension. Options control how entries are displayed in the generated **catalog** output. Note that different **catalog** formats support different sets of options. To see the different options, specify the name of the output file and then the –help option.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--ids, -i

Lista separada por vírgulas de ID de base de dados a catalogar. Se declarada, *--search* (página 311) será ignorada. Predefinição: all (todas)

--search, -s

Filtrar os resultados dos termos de pesquisa. Para saber mais sobre o formato dos termos da pesquisa, consulte a documentação relacionada com a pesquisa no manual de utilizador. Predefinição: sem filtragem

--verbose, -v

Mostrar informação de saída detalhada. Útil para depurar.

EPUB OPÇÕES

--catalog-title

Título do catálogo gerado utilizado como título nos metadados. Predefinição: 'My Books' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--cross-reference-authors

Criar referências cruzadas na secção de autores para livros com múltiplos autores. Predefinição: 'False' Aplica-se a: formatos de saída AZW3,EPUB, MOBI

--debug-pipeline

Guarda o resultado, na pasta especificada, dos diferentes estágios da cadeia de conversão. Será útil se não tiver a certeza onde é que está a ocorrer o erro do processo de conversão. Predefinição: 'None' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--exclude-genre

Expressão regular descrevendo etiquetas a excluir como géneros. Predefinição: '[.+]|^+$' exclui etiquetas com colchetes, por ex. '[Projeto Gutenberg]', e ainda '+', a etiqueta predefinida para os livros lidos. Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--exclusion-rules

Especifica as regras usadas para excluir livros do catálogo gerado. O modelo para uma regra de exclusão é ('<rule name>','Tags','<comma-separated list of tags>') ou ('<rule name>','<custom column>','<pattern>'). Por exemplo: (('Livros arquivados','#status','Arquivado'),) irá excluir um livro com um valor de 'Arquivado' na coluna personalizada 'status'. Quando estiverem definidas várias regras serão aplicadas todas as regras. Predefinição: "(('Catalogs','Tags','Catalog'),)" Aplica-se aos formatos de saída AZW3, EPUB, MOBI

--generate-authors

Incluir a secção 'Autores' no catálogo. Predefinição: 'False' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--generate-descriptions

Incluir a secção 'Descrição' no catálogo. Predefinição: 'False' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--generate-genres

Incluir a secção 'Géneros' no catálogo. Predefinição: 'False' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--generate-recently-added

Incluir a secção 'Recentemente adicionado' no catálogo. Predefinição: 'False' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--generate-series

Incluir a secção 'Séries' no catálogo. Predefinição: 'False' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--generate-titles

Incluir a secção 'Títulos' no catálogo. Predefinição: 'False' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--genre-source-field

Campo fonte para a secção 'Géneros'. Predefinição: 'Etiquetas' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--header-note-source-field

Campo personalizado que contém texto de notas para inserir no cabeçalho "Descrição". Predefinição: '' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--merge-comments-rule

#<custom field>:[before|after]:[True|False] especificando: <custom field> Campo personalizado contendo notas para fundir com os comentários [before|after] Colocação de notas em relação aos comentários [True|False] - Uma linha horizontal é inserida entre notas e comentários Predefinição: '::' Aplica-se a formatos de saída AZW3, EPUB, MOBI

--output-profile

Especifica um perfil de saída para um dispositivo. Em alguns casos, um perfil de saída é necessário para otimizar o catálogo para um dispositivo específico. Por exemplo, 'kindle' ou 'kindle_dx' cria um índice com secções e artigos. Predefinição: 'None' Aplica-se aos formatos de saída AZW3, EPUB, MOBI

--prefix-rules

Especifica as regras usadas para incluir prefixos que indiquem livros de leitura, itens de lista de desejos e outros prefixos especificados pelo utilizador. O modelo para uma regra de prefixo é ('<rule name>','<source field>','<pattern>','<prefix>'). Quando forem definidas várias regras será usada a primeira regra correspondente. Predefinição: "(('Read books','tags','+','✓'),('Wishlist item','tags','Wishlist','×'))" Aplica-se a formatos de saída AZW3, EPUB, MOBI

--preset

Utilizar um nome predefinido criado com a IGU do construtor de catálogo. Uma predefinição especifica todas as configurações para construir um catálogo. Predefinição: 'None' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--thumb-width

Dica sobre o tamanho (em pol.) para as capas de livros no catálogo. Faixa: 1.0 - 2.0 Predefinição: '1.0' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--use-existing-cover

Substituir uma capa existente quando gerar o catálogo. Predefinição: 'False' Aplica-se a: formatos de saída AZW3, EPUB, MOBI

saved_searches

```
calibredb saved_searches [opções] (list|add|remove)
```

Gerir as pesquisas guardadas armazenadas nesta base de dados. Se tentar adicionar uma consulta com um nome que já exista, este será substituído.

Sintaxe para adicionar:

calibredb **saved_searches** add search_name search_expression

Sintaxe para remover:

calibredb **saved_searches** remove search_name

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

add_custom_column

```
calibredb add_custom_column [options] label name datatype
```

Cria uma coluna personalizada. LABEL é o nome amigável da coluna para a máquina. Não pode conter espaços ou dois pontos. NAME é o nome amigável da coluna para humanos. DATATYPE pode ser um dos seguintes: bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--display

Um dicionário de opções para personalizar a forma como os dados nesta coluna serão interpretados. Esta é uma string JSON. Para colunas de enumeração, use *--display* (página 314)"{\ "enum_values\ ":[\ "val1\ ", \ "val2\ "]}" Existem muitas opções que podem ir para as opções de variáveis de exibição. As opções por tipo de coluna são: composite: composite_template, composite_sort, make_category,contains_html, use_decorations datetime: date_format enumeration: enum_values, enum_colors, use_decorations int, float: number_format text: is_names, use_decorations A melhor maneira de encontrar combinações válidas é criar uma coluna personalizada do tipo apropriado na GUI e em seguida olhar para o backup OPF de um livro (assegure-se que um novo OPF foi criado da coluna foi acrescentado). Vai ver o JSON da "exibição" para a nova coluna em OPF.

--is-multiple

Esta coluna armazena etiquetas como dados (ex. múltiplos valores separados por vírgulas). Só se aplica caso o tipo de dados seja de texto.

custom_columns

```
calibredb custom_columns [opções]
```

Lista as colunas personalizadas disponíveis. Mostra rótulos de colunas e os ID.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--details, -d

Mostrar detalhes para cada coluna.

remove_custom_column

```
calibredb remove_custom_column [opções] etiquetas
```

Remover o campo personalizado com o rótulo especificado. Pode ver quais os campos personalizados disponíveis com o comando custom_columns.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--force, -f

Não solicitar confirmação

set_custom

```
calibredb set_custom [opções] valor ID coluna
```

Define o conteúdo de um coluna personalizada para o livro identificado pelo ID. Pode obter uma lista dos ID usando o comando search. Pode obter uma lista de colunas personalizadas usando o comando custom_columns.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--append, -a

Se a coluna armazena valores múltiplos, adicione os valores especificados aos já existentes em vez de os substituir.

restore_database

```
calibredb restore_database [options]
```

Restaurar esta base de dados a partir dos metadados armazenados nos ficheiros OPF em cada pasta da biblioteca do Calibre. Isto é útil apenas se o seu ficheiro metadata.db tiver sido corrompido.

AVISO: Este comando regenera completamente a sua base de dados. Vai perder todas as pesquisas guardadas, categorias do utilizador, conectores, preferências de conversão por livro e receitas personalizadas. Os metadados restaurados serão tão exatos quanto os encontrados nos ficheiros OPF.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--really-do-it, -r

Executar efetivamente a recuperação. O comando não irá ser executado se esta opção não for especificada.

check_library

```
calibredb check_library [options]
```

Executar algumas verificações no sistema de ficheiros representando uma biblioteca. Os relatórios são invalid_titles, extra_titles, invalid_authors, extra_authors, missing_formats, extra_formats, extra_files, missing_covers, extra_covers, failed_folders

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--csv, -c

Saída em CSV

--ignore_extensions, -e

Lista separada por vírgulas de extensões a ignorar: Predefinição: todas

--ignore_names, -n

Lista separada por vírgulas de nomes a ignorar: Predefinição: todos

--report, -r

Lista separada por vírgulas de relatórios: Predefinição: todos

--vacuum-fts-db

Esvaziar a base de dados de pesquisa em texto integral. Dependendo do tamanho da base de dados isto pode ser muito lento e consumir muita memória.

list_categories

```
calibredb list_categories [options]
```

Produzir um relatório das informações da categoria na base de dados. A informação é equivalente à que é mostrada no navegador de conteúdos.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--categories, -r
: Lista separada por vírgulas de categorias de limitadores de pesquisa. Predefinição: tudo

--csv, -c
: Saída em CSV

--dialect
: O tipo de ficheiro CSV a criar. Opções: excel, excel-tab, unix

--item_count, -i
: Na saída apresentar apenas o número de itens numa categoria em vez da contagem por item dentro de cada categoria

--width, -w
: A largura máxima de uma linha no ficheiro de destino. A predefinição será detetar o tamanho do ecrã.

backup_metadata

```
calibredb backup_metadata [options]
```

Cópia de segurança dos metadados armazenados na base de dados em ficheiros OPF individuais em cada diretório de livros. Normalmente isto acontece automaticamente, mas pode executar este comando para forçar uma recriação dos ficheiros OPF, com a opção –all.

Note que normalmente não há necessidade de fazer isto, pois os ficheiros OPF são copiados automaticamente sempre que os metadados sejam alterados.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--all
: Por norma, este comando só funciona com livros que tenham ficheiros OPF antiquados. Esta opção faz com que funcione com todos os livros.

clone

```
calibredb clone path/to/new/library
```

Criar um **clone** da biblioteca atual. Isto cria uma nova biblioteca, vazia, mas que contém as mesmas colunas personalizadas, bibliotecas virtuais e outras configurações da biblioteca atual.

A biblioteca clonada não contém livros. Se quiser criar uma cópia completa, incluindo todos os livros, use as ferramentas do sistema operativo para copiar a pasta da biblioteca.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

embed_metadata

```
calibredb embed_metadata [opções] book_id
```

Atualiza os metadados nos ficheiros atuais dos livros existentes na biblioteca do Calibre a partir dos metadados guardados na base de dados do Calibre. Normalmente, os metadados são atualizados apenas quando se exportam ficheiros a partir do Calibre, donde este comando é útil quando pretende que os ficheiros sejam atualizados na pasta onde estão guardados. Note que diferentes formatos de ficheiro suportam diferentes quantidades de metadados. Pode usar o valor especial "all" para o parâmetro book_id de modo a atualizar os metadados em todos os livros. Pode também especificar múltiplos ID de livros separando-os com espaços ou usando gamas de ID, separados por hífenes. Por exemplo: calibredb **embed_metadata** 1 2 10-15 23

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--only-formats, -f

Atualizar metadados apenas em ficheiros de formato especifico. Especifique-o vários vezes para vários formatos. Por definição, todos os formatos são atualizados.

search

```
calibredb search [opções] expressão_de_pesquisa
```

Procura nesta biblioteca o termo de pesquisa especificado, devolvendo uma lista de ID de livros, separada por virgulas, que correspondem ao critério de pesquisa. O formato de saída é útil como fonte de dados para outros comandos que aceitem uma lista de ID como valor de entrada.

A expressão de pesquisa pode ser qualquer das possíveis construções da poderosa linguagem de queries de pesquisa do Calibre, como por exemplo: calibredb **search** author:asimov "title:»i robot»"

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--limit, -l

O número máximo de resultados a mostrar. A predefinição é todos os resultados.

fts_index

```
calibredb fts_index [options] enable/disable/status/reindex
```

Control the Full text search indexing process.

enable

Turns on FTS indexing for this library

disable

Turns off FTS indexing for this library

status

Shows the current indexing status

reindex

Can be used to re-index either particular books or the entire library. To re-index particular books specify the book ids as additional arguments after the reindex command. If no book ids are specified the entire library is re-indexed.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--indexing-speed

A velocidade da indexação. Escolha entre Rápida para indexação rápida que utilizará todos os recursos do seu computador ou, Lenta para indexação menos intensiva a nível dos recursos. Note que a velocidade é redefinida para lenta depois de cada invocação.

--wait-for-completion

Aguarde até que todos os livros sejam indexados. Periodicamente será mostrado o progresso da ação

fts_search

```
calibredb fts_search [opções] expressão de pesquisa
```

Executa uma pesquisa de texto integral em toda a biblioteca ou num seu subconjunto.

Sempre que passar argumentos para o calibredb que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

--do-not-match-on-related-words

Encontrar apenas palavras exatas, e não palavras relacionadas. Dessa forma, 'corrigir' não irá encontrar 'corrigindo'.

--include-snippets

Incluir trechos do texto à volta de cada correspondência. Note que isto torna a pesquisa muito mais lenta.

--indexing-threshold

A parte da biblioteca que deve ser indexada antes que a pesquisa seja permitida, tal como uma percentagem. Valor predefinido é 90

--match-end-marker

O marcador usado para indicar o final de uma palavra correspondente dentro de um trecho

--match-start-marker

O marcador usado para indicar o início de uma palavra correspondente dentro de um trecho

--output-format

O formato de saída dos resultados da pesquisa. Use "text" para texto simples ou "json" para saída JSON.

--restrict-to

Restringir os livros pesquisados, seja usando uma expressão de pesquisa ou por ID. Por exemplo: ids:1,2,3 para restringir por ID ou search:tag:foo para restringir a livros com a etiqueta 'foo'.

13.1.7 ebook-convert

```
ebook-convert ficheiro_entrada ficheiro_saída [opções]
```

Converte um formato de livro digital noutro formato

ficheiro_entrada (input_file) é a fonte e o ficheiro_saída (output_file) é o resultado final. Os dois devem ser especificados como os dois primeiros argumentos para o comando.

O formato do resultado final é calculado através da extensão de ficheiro do ficheiro_saída. O ficheiro_saída pode ter também o formato especial .EXT onde EXT é a extensão do ficheiro de saída. Neste caso, o nome do formato do ficheiro

de saída é derivado do nome do ficheiro de entrada. Note que o nome do ficheiro não pode começar com um hífen. Finalmente, se ficheiro_saída não tiver extensão, então será tratado como pasta e será gravado nessa pasta um «livro digital aberto» (OEB) composto por ficheiros HTML. Estes ficheiros são os ficheiros que normalmente seriam passados pelo plugin de saída.

Depois de especificar o ficheiro de entrada e o de saída poderá personalizar a conversão através da especificação de várias opções. As opções disponíveis dependem dos tipos de ficheiro de entrada e do de saída. Para obter ajuda neste tópico indique o ficheiro de entrada e o de saída seguido da opção -h .

Para obter a documentação completa do sistema de conversão veja *Conversão de livros digitais* (página 55)

Sempre que passar argumentos para o **ebook-convert** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

As opções e os valores das opções variam dependendo simultaneamente do formato de entrada e saída. Assim você deve sempre verificar com:

```
ebook-convert myfile.input_format myfile.output_format -h
```

Abaixo estão as opções que são comuns a todas as conversões, seguidas das opções especificas a cada formato de entrada e saída.

- *OPÇÕES DE ORIGEM* (página 320)
- *OPÇÕES DE DESTINO* (página 320)
- *VISUAL* (página 321)
- *PROCESSAMENTO HEURÍSTICO* (página 324)
- *PESQUISAR E SUBSTITUIR* (página 324)
- *DETEÇÃO DE ESTRUTURA* (página 325)
- *ÍNDICE* (página 326)
- *METADADOS* (página 326)
- *DEPURAR* (página 327)

--help, -h

mostrar esta mensagem de ajuda e sair

--input-profile

Especificar o perfil de origem. O perfil de origem indica ao sistema de conversão sobre como interpretar várias informações no documento de origem. Por exemplo, comprimentos dependentes da resolução (i. e. comprimento em píxeis). Escolhas disponíveis: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes

Listar nomes incorporados das receitas. Poderá criar um livro digital a partir das receitas criadas como em: ebook--convert "Nome da receita.receita" output.epub

--output-profile

Especifica o perfil de saída. O perfil de saída indica ao sistema de conversão como otimizar o documento criado para o dispositivo especificado (como, por exemplo, redimensionar imagens para as dimensões do tamanho do ecrã). Em alguns casos, um perfil de saída pode ser utilizado para otimizar a saída para um dispositivo em particular, mas isso raramente é necessário. As escolhas são:cybookg3, cybook_opus, default, generic_eink, generic_eink_hd, generic_eink_large, hanlinv3, hanlinv5, illiad, ipad, ipad3, irexdr1000, irexdr800, jetbook5, kindle, kindle_dx, kindle_fire, kindle_oasis, kindle_pw, kindle_pw3, kindle_scribe, kindle_voyage, kobo, msreader, mobipocket,

nook, nook_color, nook_hd_plus, pocketbook_inkpad3, pocketbook_lux, pocketbook_hd, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet

--version

mostrar o número da versão do programa e sair

OPÇÕES DE ORIGEM

Opções para controlar o processamento do ficheiro de origem mobi

--input-encoding

Especifica a codificação de caracteres para o documento de origem. Se definida, esta opção irá sobrepor-se a qualquer codificação declarada no documento. Particularmente útil em documentos que não declaram uma codificação ou que possuem declarações de codificação erradas

OPÇÕES DE DESTINO

Opções para controlar o processamento do ficheiro de destino epub

--dont-split-on-page-breaks

Desligar a separação nas quebras de página. Normalmente os ficheiros de origem são automaticamente separados, nas quebras de página, em dois ficheiros. Isto faz com que o livro de destino seja analisado mais rapidamente e ocupando menos recursos. No entanto a separação é um processo lento e se o seu ficheiro de origem tiver muitas quebras de página deve desligar a separação nas quebras de página.

--epub-flatten

Esta opção só é necessária se pretende utilizar EPUB com o FBReaderJ. Vai remover a estrutura de ficheiros do formato EPUB colocando todas as imagens no nível de topo.

--epub-inline-toc

Inserir um índice incorporado que irá aparecer como parte do conteúdo principal do livro.

--epub-toc-at-end

Colocar o índice incorporado no final do livro em vez de no início.

--epub-version

A versão do ficheiro EPUB a ser gerado. O EPUB 2 é o mais compatível, use apenas o EPUB 3 se achar que realmente precisa dele.

--extract-to

Extrai o conteúdo do ficheiro EPUB gerado para a pasta especificada. O conteúdo da pasta especificada vai ser primeiro apagado, portanto, tenha cuidado.

--flow-size

Dividir todos os ficheiros HTML maiores que este tamanho (em KB). Isto é necessário porque a maioria dos leitores EPUB não lida bem com ficheiros muito grandes. A predefinição de 260KB é o tamanho requerido por Adobe Digital Editions. Definir para 0 para desativar a divisão baseada no tamanho.

--no-default-epub-cover

Normalmente, se o ficheiro de origem não tem capa, e se não especificar uma, será gerada uma capa predefinida com o título, autores, etc. Esta opção desativa a criação dessa capa.

--no-svg-cover

Não utilizar SVG para a capa do livro. Utilize esta opção se o seu EPUB for utilizado num dispositivo que não suporte SVG, como o iPhone o JetBook Lite. Sem esta opção estes dispositivos mostrarão uma capa em branco.

--preserve-cover-aspect-ratio

Quando utilizar uma capa SVG, esta opção permite esticar a capa para a área do ecrã disponível mas preservando o aspeto (proporção largura/altura). Isto significa que podem surgir margens brancas nos lados ou no topo e fundo da imagem mas a imagem não será distorcida.

--pretty-print

Se especificado, o plugin de saída irá tentar criar um ficheiro o mais próximo possível da leitura humana. Poderá não produzir alterações em alguns plugins de saída.

--toc-title

Título para qualquer índice incorporado gerado.

VISUAL

Opções para controlar o visual e funcionamento do ficheiro de destino

--asciiize

Transliterar caracteres Unicode para uma representação em ASCII. Utilize com cuidado porque irá substituir os caracteres Unicode por ASCII. Por exemplo, irá substituir "Pelé" por "Pele". Note também que em certos casos onde existem representações múltiplas de um carácter (por exemplo caracteres partilhados pelo Chinês e Japonês) será utilizada a representação baseada no idioma da interface atual do Calibre.

--base-font-size

O tamanho base do tipo de letra em pts. Todos os tamanhos dos tipos de letra no livro produzido vão ser alterados proporcionalmente, baseados neste tamanho. Ao escolher um tamanho maior os tipos de letra no ficheiro de destino serão maiores e vice versa. Por predefinição, se o valor for zero, o tamanho base do tipo de letra é escolhido baseado no perfil de destino que escolheu.

--change-justification

Mudar justificação do texto. Um valor de "left" vai converter todo o texto justificado no original para texto com alinhamento à esquerda (ou seja, sem justificação). Um valor de "justify" converte todo o texto sem justificação para justificado. Um valor de "original" (a predefinição) não altera a justificação no ficheiro fonte. Tenha em conta que apenas alguns formatos de saída suportam a justificação de texto.

--disable-font-rescaling

Desativar a alteração proporcional do tamanho dos tipos de letra.

--embed-all-fonts

Incorporar todos os tipos de letra referenciada no documento de entrada, mas que ainda não foram incorporadas. Isto irá procurar no seu sistema por tipo de letra, e se as encontrar, irá incorporá-las. A incorporação só irá funcionar se o formato que está a converter suporta letras incorporadas, p.ex. EPUB, AZW3, DOCX ou PDF. Por favor, garanta que tem a licença adequada para incorporar os tipos de letra utilizadas neste documento.

--embed-font-family

Incorporar a família tipográfica especificada no livro. Isto especifica o tipo de letra base a ser utilizada pelo livro. Se o documento de entrada especificar os seus próprios tipos de letra, eles poderão sobrescrever o tipo de letra base. Pode usar a opção de filtragem de informação de estilos para remover os tipos de letra do documento de entrada. Note que a incorporação de tipos de letra só funciona com certos formatos de saída, principalmente com EPUB, AZW3 e DOCX.

--expand-css

Por predefinição, o Calibre vai usar o modo abreviado para várias propriedades CSS como margem, espaçamento, contorno, etc. Esta opção fará com que use a forma expandida. Note que o CSS é sempre expandido ao gerar ficheiros EPUB com o perfil de saída definido para um dos perfis do Nook, porque o Nook não permite CSS abreviado.

--extra-css

O caminho para a folha de estilos CSS ou CSS em bruto. Este CSS vai ser adicionado às regras de estilo do ficheiro de origem de modo a ser usado para se sobrepor a essas regras.

--filter-css

Uma lista separada por vírgulas de propriedades CSS que serão removidas de todas as regras de estilo do CSS. Isto é útil se a presença de informações de alguns estilos impede o estilo de ser anulado no seu dispositivo. Por exemplo: font-family,color,margin-left,margin-right

--font-size-mapping

Mapeamento dos nomes dos tipos de letra CSS para tamanhos em pts. Um exemplo de definição é 12,12,14,16,18,20,22,24. Estes são os mapeamentos para os tamanhos xx-small até xx-large, sendo o tamanho final o maior. O algoritmo de alteração proporcional dos tipos de letra usa estes tamanhos para alterar de forma inteligente o tamanho dos tipos de letra. A predefinição é usar um mapeamento baseado no perfil de destino que escolheu.

--insert-blank-line

Inserir uma linha em branco entre os parágrafos. Não funciona se o ficheiro de origem não usar parágrafos (etiquetas <p> ou).

--insert-blank-line-size

Definir a altura das linhas inseridas em branco (valores `'em'`). A altura das linhas entre parágrafos será duas vezes o tamanho definido aqui.

--keep-ligatures

Preserva as ligaturas presentes no documento de entrada. Uma ligatura é um formato especial para mostrar um par de caracteres tal como ff, fi, fl, etc. A maioria dos leitores não têm suporte para ligaturas nos seus tipos de letra predefinidos e não conseguem mostrá-las corretamente. Por predefinição, o Calibre irá transformar uma ligatura no seu par correspondente de caracteres normais. Esta opção irá preservar as ligaturas.

--line-height

A altura da linha em pontos. Controlar espaçamento entre linhas consecutivas de texto. Apenas aplicar elementos que não definam a sua própria altura de linha. Na maioria dos casos, é mais útil usar a opção de altura mínima de linha. Por predefinição não será feita qualquer manipulação da altura de linha.

--linearize-tables

Alguns documentos mal desenhados usam tabelas para controlar a disposição do texto na página. Quando convertidos, estes documentos apresentam texto que sai para fora da página bem outros problemas. Esta opção extrai o conteúdo das tabelas e apresenta-o de uma forma linear.

--margin-bottom

Definir a margem inferior em pts. A predefinição é 5.0. Definir este valor para 0 irá fazer com que nenhuma margem seja definida (as margens do documento original serão preservadas). Nota: formatos já com medidas da páginas pré-estabelecidas como PDF e DOCX já têm as sua próprias definições de margem determinadas.

--margin-left

Definir a margem esquerda em pts. A predefinição é 5.0. Definir este valor para 0 irá fazer com que nenhuma margem seja definida (as margens do documento original serão preservadas). Nota: formatos já com medidas da páginas pré-estabelecidas como PDF e DOCX já têm as sua próprias definições de margem determinadas e que têm precedência.

--margin-right

Definir a margem direita em pts. A predefinição é 5.0. Definir este valor para 0 irá fazer com que nenhuma margem seja definida (as margens do documento original serão preservadas). Nota: formatos já com medidas da páginas pré-estabelecidas como PDF e DOCX já têm as sua próprias definições de margem determinadas.

--margin-top

Definir a margem superior em pts. A predefinição é 5.0. Definir este valor para 0 irá fazer com que nenhuma margem seja definida (as margens do documento original serão preservadas). Nota: formatos já com medidas da páginas pré-estabelecidas como PDF e DOCX já têm as sua próprias definições de margem determinadas e que têm precedência.

--minimum-line-height

A altura mínima da linha, como percentagem calculada a partir do tamanho de letra do elemento. O Calibre irá assegurar-se que cada elemento tem um altura de linha de pelo menos o valor aqui definido, independentemente do que o documento ativo especificar. Defina com Zero para desativar isto. A predefinição é 120% . Use de preferência esta definição em vez da especificação direta da altura de linha, a não ser que saiba exatamente o que está a fazer. Por exemplo, pode obter texto com "espaço duplo" se definir isto com 240.

--remove-paragraph-spacing

Remover o espaçamento entre parágrafos. Também define uma indentação nos parágrafos de 1.5em. A remoção do espaçamento não funciona se o ficheiro de origem não usar parágrafos (etiquetas <p> ou).

--remove-paragraph-spacing-indent-size

Quando o Calibre remove linhas em branco entre parágrafos, insere automaticamente uma indentação de parágrafo para garantir que os parágrafos sejam facilmente distinguíveis. Esta opção controla o tamanho desse avanço (unidade de medida "em"). Se inserir um valor negativo, será utilizado o avanço especificado no documento, o que significa que o Calibre não modificará a indentação.

--smarten-punctuation

Converter aspas simples, hífenes e reticências para os seus equivalentes tipográficos. Para detalhes, ver https://daringfireball.net/projects/smartypants

--subset-embedded-fonts

Segmentar todos os tipos de letra incorporadas. Todos os tipos de letra incorporados serão reduzidas para conter apenas os símbolos utilizados no documento. Isto diminui o tamanho dos ficheiros de tipos de letra. È particularmente útil se estiver a incorporar um tipo de letra pesado e com muitos símbolos não utilizados.

--transform-css-rules

Caminho para um ficheiro que contém as regras para transformar os estilos CSS neste livro. O modo mais fácil de criar este ficheiro é usar o Assistente para criar regras na interface gráfica do Calibre. Para tal, aceda à secção "Aparência->Transformar estilos" do diálogo de conversão. Uma vez criadas as regras, pode usar o botão "Exportar" para as guardar num ficheiro.

--transform-html-rules

Caminho para um ficheiro que contém as regras para transformar o HTML neste livro. O modo mais fácil de criar este ficheiro é usar o Assistente para criar regras na interface gráfica do Calibre. Para tal, aceda à secção "Aparência->Transformar HTML" do diálogo de conversão. Uma vez criadas as regras, pode usar o botão "Exportar" para as guardar num ficheiro.

--unsmarten-punctuation

Converter citações elegantes, traços e reticências aos seus equivalentes simples.

PROCESSAMENTO HEURÍSTICO

Modifique o texto do documento e a sua estrutura usando padrões comuns. Desativada por predefinição. Usar –enable--heuristics para ativar. Ações individuais podem ser desativadas com as opções –disable-* .

--disable-dehyphenate

Analisa o documento por palavras que contenham hífenes. O documento será utilizado como um dicionário para determinar se os hífenes devem ser mantidos ou removidos.

--disable-delete-blank-paragraphs

Remover parágrafos vazios do documento quando estão entre outros parágrafos

--disable-fix-indents

Transforma a indentação criada a partir de espaços impartíveis múltiplos em indentações CSS.

--disable-format-scene-breaks

Marcadores de quebra de cena alinhados à esquerda são centrados. Substituir quebras de cenas suaves que usam várias linhas em branco por linhas horizontais.

--disable-italicize-common-cases

Pesquisar por palavras comuns e padrões que denotam itálicos e tornar esse texto itálico

--disable-markup-chapter-headings

Detetar cabeçalhos e subcabeçalhos de capítulos sem formatação. Trocá-los para etiquetas h2 e h3. Esta opção não irá criar um índice, mas pode ser utilizada juntamente com a estrutura de deteção para o criar.

--disable-renumber-headings

Pesquisar por ocorrências de etiquetas html <h1> ou <h2>. As etiquetas serão renumeradas para evitar divisões no meio de cabeçalhos de capítulos.

--disable-unwrap-lines

Unifique as linhas usando pontuação e outras pistas de formatação.

--enable-heuristics

Ativar o processamento heurístico. Esta opção deve ser ativada para que o processamento heurístico ocorra.

--html-unwrap-factor

Escala utilizada para determinar o tamanho em que uma linha deve ter as quebras eliminadas. Valores válidos são decimais entre 0 e 1. A predefinição é 0.4, um pouco abaixo do tamanho médio de uma linha. Se apenas algumas linhas do documento precisam de quebras eliminadas, este valor deve ser reduzido.

--replace-scene-breaks

Substitui quebras de lógica no texto especificado. Pelas definições, o texto do documento já escrito será usado.

PESQUISAR E SUBSTITUIR

Modificar o texto e estrutura do documento de acordo com padrões definidos pelo utilizador.

--search-replace

Caminho para um ficheiro que contém expressões de pesquisa e substituição comuns. O ficheiro deve conter linhas alternadas de expressão comum seguida do padrão de substituição (que pode ser uma linha vazia). As expressões regulares devem estar na sintaxe Python regex e o ficheiro deve estar codificado em UTF-8.

--sr1-replace

Substituição para substituir o texto encontrado com a sr1-search

--sr1-search

Padrão de pesquisa (expressão regular) a ser substituído por um sr1-replace

--sr2-replace

Substituição para substituir o texto encontrado com a sr2-search

--sr2-search

Padrão de pesquisa (expressão regular) a ser substituído pela sr2-replace

--sr3-replace

Substituição para substituir o texto encontrado com a sr3-replace

--sr3-search

Padrão de pesquisa (expressão regular) a ser substituído pela sr3-replace

DETEÇÃO DE ESTRUTURA

Controlar a deteção automática da estrutura do documento.

--chapter

Uma expressão XPath para apagar os títulos dos capítulos. A predefinição é considerar as etiquetas <h1> ou <h2> que contém as palavras "capítulo","livro","secção", "prólogo", "epilogo" ou "parte" como títulos de capítulos assim como quaisquer etiquetas que tenham class="chapter". A expressão usada deve avaliar uma lista de elementos. Para desativar a deteção de capítulos, use a expressão "/". Veja o tutorial do XPath no Manual do Utilizador do Calibre para mais ajudas no uso desta função.

--chapter-mark

Especificar como marcar os capítulos detetados. Um valor "quebra de página" vai inserir quebras de página antes dos capítulos. Um valor "regra" vai inserir uma linha antes dos capítulos. Um valor "nenhum" vai desativar a marcação de capítulos e um valor "ambos" irá usar tanto quebras de página como linhas para marcar os capítulos.

--disable-remove-fake-margins

Alguns documentos especificam tamanhos de margem direita e esquerda individualmente para cada parágrafo. O Calibre tentará detetar e remover estas margens. Alguma vezes, isso pode causar a remoção indevida de margens. Neste caso, pode desativar a remoção.

--insert-metadata

Inserir os metadados no início do livro. Isto é útil se o seu leitor não suporta apresentar/pesquisar os metadados diretamente.

--page-breaks-before

Uma expressão XPath. Serão inseridas quebras de página antes dos elementos especificados. Para desativar use a expressão: /

--prefer-metadata-cover

Usar a capa detetada no ficheiro de origem em vez da capa especificada.

--remove-first-image

Remover a primeira imagem do livro especificado. Útil se o documento especificado possui uma imagem de capa não reconhecida como tal. Se tal existir, e no caso desta opção não ter sido especificada, quando definir uma capa no Calibre o documento gerado ficará com duas imagens de capa.

--start-reading-at

Uma expressão XPath para detetar a localização no documento a partir do qual se deve iniciar a leitura. Alguns programas de leitura de ebooks (como por exemplo o Kindle) usam esta localização como posição para abertura do livro. Veja o tutorial do XPath no Manual do Utilizador do Calibre para mais ajuda no uso desta função.

ÍNDICE

Controlar a geração automática de um índice. Por predefinição, se o ficheiro de origem tiver um índice, este é utilizado em vez do gerado automaticamente.

--duplicate-links-in-toc

Quando criar um índice a partir de hiperligações no documento de entrada, permitir entradas duplicadas. Por ex: permitir mais de uma entrada com o mesmo texto, desde que apontem para localizações diferentes.

--level1-toc

Expressão XPath que especifica todas as etiquetas que deverão ser adicionadas ao primeiro nível do índice. Se isto for especificado, terá precedência em relação a outras formas de deteção automática. Consulte o tutorial do XPath no Manual do utilizador do Calibre para exemplos.

--level2-toc

Expressão XPath que especifica todas as etiquetas que deverão ser adicionadas no segundo nível do índice . Cada entrada será adicionada abaixo da entrada anterior do nível um. Consulte o tutorial do XPath no Manual do Utilizador do Calibre para exemplos.

--level3-toc

Expressão XPath que especifica todas as etiquetas que deverão ser adicionadas no Índice do terceiro nível. Cada entrada será adicionada abaixo da entrada anterior no nível dois. Consulte o tutorial do XPath no Manual do Utilizador do Calibre para exemplos.

--max-toc-links

Número máximo de atalhos a inserir no índice. Definir para 0 para desativar. A predefinição é: 50. Os atalhos só são adicionados ao índice se forem detetados menos que o limite de capítulos.

--no-chapters-in-toc

Não adicionar ao Índice os capítulos detetados automaticamente.

--toc-filter

Remove as entradas do Índice cujos títulos correspondem à expressão regular especificada. As entradas correspondentes e as suas dependentes são removidas.

--toc-threshold

Se forem detetados menos capítulos do que este número, os atalhos serão adicionados ao índice. A predefinição é: 6

--use-auto-toc

Normalmente, se o ficheiro de origem já tem um Índice este é utilizado em vez do gerado automaticamente. Com esta opção o gerado automaticamente é sempre utilizado.

METADADOS

Opções para definir os metadados no ficheiro de saída

--author-sort

Expressão a ser usada quando ordenar por autor.

--authors

Definir os autores. Múltiplos autores devem ser separados por um e comercial (ampersand).

--book-producer

Definir o produtor do livro.

--comments

Definir a descrição do livro.

--cover

Define a capa para o ficheiro de imagem especificado ou de um URL

--isbn

Definir ISBN do livro.

--language

Definir idioma.

--pubdate

Definir a data de publicação (esta é assumida no fuso horário local, a menos que um fuso horário seja explicitamente especificado)

--publisher

Definir a editora do livro.

--rating

Definir a avaliação. Deve ser um algarismo entre 1 e 5.

--read-metadata-from-opf, --from-opf, -m

Ler os metadados do ficheiro OPF especificado. Os metadados lidos deste ficheiro vão sobrepor-se aos metadados no ficheiro de origem.

--series

Definir a série a que este livro pertence.

--series-index

Definir o índice do livro nesta série.

--tags

Definir as etiquetas do livro. Deve ser uma lista separada por vírgulas.

--timestamp

Definir a marca temporal do livro (algo já em desuso)

--title

Definir o título.

--title-sort

A versão do título a ser usada para a alfabetação.

DEPURAR

Opções para ajudar com a depuração da conversão

--debug-pipeline, -d

Guarda o resultado, na pasta especificada, dos diferentes estágios da cadeia de conversão. Será útil se não tem a certeza em que etapa do processo de conversão é que está a ocorrer o erro.

--verbose, -v

Nível de verbosidade. Especifique múltiplas vezes para aumentar a verbosidade. Se especificar duas vezes obterá uma verbosidade completa, uma vez terá uma verbosidade média e zero vezes uma verbosidade mínima.

13.1.8 ebook-edit

```
ebook-edit [opts] [path_to_ebook] [name_of_file_inside_book ...]
```

Iniciar a ferramenta Editor de livros do Calibre. Você também pode especificar os nomes de ficheiros dentro do livro que serão abertos automaticamente para edição.

Sempre que passar argumentos para o **ebook-edit** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--detach
: Se existir, desligue do terminal de controlo (apenas para Linux)

--help, -h
: mostrar esta mensagem de ajuda e sair

--select-text
: O texto a selecionar no livro quando for aberto para edição

--version
: mostrar o número da versão do programa e sair

13.1.9 ebook-meta

```
ebook-meta ebook_file [opções]
```

Ler/Escrever metadados a partir de/para ficheiros de ebook.

Formatos suportados para leitura de metadados: azw, azw1, azw3, azw4, cb7, cbc, cbr, cbz, chm, docx, epub, fb2, fbz, html, htmlz, imp, lit, lrf, lrx, mobi, odt, oebzip, opf, pdb, pdf, pml, pmlz, pobi, prc, rar, rb, rtf, snb, tpz, txt, txtz, updb, zip

Formatos suportados para escrita de metadados: azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

Diferentes tipos de ficheiros suportam diferentes tipos de metadados. Se tentar definir um metadado num tipo de ficheiro que não o suporte, o metadado será ignorado silenciosamente.

Sempre que passar argumentos para o **ebook-meta** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--author-sort
: Expressão a ser usada quando ordenar pelo autor. Se não especificada, e o(s) autor(es) são especificados, então será gerada automaticamente a partir do(s) autor(es).

--authors, -a
: Definir os autores. Múltiplos autores devem ser separados por um E comercial (ampersand). Os nomes dos autores devem ter a ordem Nome Apelido.

--book-producer, -k

Definir o produtor do livro.

--category

Definir a categoria do livro.

--comments, -c

Definir a descrição do livro.

--cover

Definir a capa do ficheiro especificado.

--date, -d

Definir a data de publicação.

--from-opf

Ler os metadados do ficheiro OPF especificado e usa-los para definir os metadados no livro. Os metadados especificados na linha de comandos serão sobrepostos aos lidos do ficheiro OPF.

--get-cover

Obter capa do livro e guardá-la como ficheiro especificado.

--help, -h

mostrar esta mensagem de ajuda e sair

--identifier

Definir identificadores para este livro, o que pode ser especificado com múltiplos formatos. Por exemplo: *--identifier* (página 329) uri:https://acme.com *--identifier* (página 329) isbn:12345 Para remover um identificador basta não especificar o valor de um identificador *--identifier* (página 329) isbn: . Note que para ficheiros EPUB, um identificador marcado como o identificador de pacote não pode ser removido.

--index, -i

Definir o índice do livro nesta série.

--isbn

Definir ISBN do livro.

--language, -l

Definir idioma.

--lrf-bookid

Definir o ID do livro nos ficheiros LRF

--publisher, -p

Definir a editora do livro.

--rating, -r

Definir a avaliação. Deve ser um algarismo entre 1 e 5.

--series, -s

Definir a série a que este livro pertence.

--tags

Definir as etiquetas do livro. Deve ser uma lista separada por vírgulas.

--title, -t

Definir o título.

--title-sort

A versão do título a ser usada para a alfabetação. Se não especificada, e o título for especificado, então será gerada automaticamente a partir do título.

--to-opf

Especificar o nome de um ficheiro OPF. Os metadados serão escritos no ficheiro OPF.

--version

mostrar o número da versão do programa e sair

13.1.10 ebook-polish

```
ebook-polish [options] input_file [output_file]
```

Refinar livros serve para colocar um pouco mais de perfeição nos seus preciosos livros digitais. O refinamento tenta minimizar as alterações no código interno do seu livro digital. Ao contrario da conversão, não faz grandes alterações tais como achatar CSS, renomear ficheiros, alterar o tamanho da letra, ajustar as margens, etc. Cada ação realizada faz apenas o conjunto mínimo das alterações necessárias para o efeito desejado. Deve usar esta ferramenta como o passo final no processo de criação de livros digitais.

Note que o refinamento só funciona em livros nos formatos AZW3 ou EPUB.

Sempre que passar argumentos para o **ebook-polish** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--add-soft-hyphens, -H

Adiciona hífenes discricionários a todas as palavras no livro. Tal permite que o livro seja melhor renderizado quando o texto for justificado em leitores que não suportam hifenização.

--compress-images, -i

Comprimir sem perdas as imagens no livro para reduzir o tamanho do ficheiro e sem afetar a qualidade percetível das imagens.

--cover, -c

Caminho para a imagem de capa. Altera a capa especificada no livro digital. Se a capa não estiver presente ou não estiver identificada corretamente, será inserida uma nova capa.

--embed-fonts, -e

Incorporar todas os tipos de letra que sejam referenciados no documento e ainda não estejam incorporados. Isto irá procurar no seu computador pelos tipos de letra e, desde que sejam encontrados, vão ser incorporados no documento. Por favor, certifique-se que possui uma licença adequada para incorporar os tipos de letra utilizados neste documento.

--help, -h

mostrar esta mensagem de ajuda e sair

--jacket, -j

Insere uma página de "sobrecapa" no inicio do livro, mas após a capa, para apresentar todos os metadados padrão tal como o título, etiquetas, autor, séries, comentários, etc. Qualquer sobrecapa anterior será substituída.

--opf, -o

Localização do ficheiro OPF. Os metadados do livro são atualizados a partir do ficheiro OPF.

--remove-jacket

Remover a sobrecapa inserida anteriormente.

--remove-soft-hyphens

Remover hífenes discricionários de todo o texto no livro.

--remove-unused-css, -u

Remover todas as regras CSS de folhas de estilo e etiquetas de <estilo> não utilizadas. Alguns livros, criados a partir de modelos de produção, podem ter um grande número de regras CSS extra que não correspondem a qualquer conteúdo atual. Estas regras extra podem tornar mais lentos os dispositivos leitores porque terão de as analisar todas.

--smarten-punctuation, -p

Converter traços de texto, reticências, aspas, hífenes múltiplos, etc, nos seus equivalentes tipograficamente corretos. Note que o algoritmo poderá, por vezes, produzir resultados incorretos, especialmente quando estejam envolvidas aspas simples no começo das contrações.

--subset-fonts, -f

Segmentar tipos de letra significa reduzir um tipo de letra incorporado para conter apenas os caracteres utilizados no livro do ficheiro desse tipo de letra. Isto reduz consideravelmente o tamanho dos ficheiros de tipos de letra (em relação ao ficheiro do tipo de letra a redução pode ser de metade). Por exemplo, se o livro usa um tipo de letra específico para os cabeçalhos, então ao segmentar tipos de letra irá reduzir esse tipo de letra para conter apenas os caracteres presentes nos cabeçalhos atuais no livro. Ou, se o livro incorporar as versões em negrito e itálico de um tipo de letra, mas o uso de texto em negrito e itálico é relativamente raro, ou mesmo ausente, então os tipos de letra em negrito e itálico podem ser reduzidos para apenas alguns caracteres ou completamente removidas. A única desvantagem da segmentação de tipos de letra é que se, mais tarde, decidir adicionar mais texto aos seus livros, o texto recém-adicionado pode não ser totalmente abrangido pelo subconjunto de tipo de letra.

--upgrade-book, -U

Atualizar as estruturas internas do livro, se possível. Por exemplo, atualizar livros EPUB 2 para livros EPUB 3

--verbose

Produz saída de dados mais detalhada, útil para a depuração.

--version

mostrar o número da versão do programa e sair

13.1.11 ebook-viewer

```
ebook-viewer [options] ficheiro
```

Ver um livro.

Sempre que passar argumentos para o **ebook-viewer** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--continue

Continuar a ler o último livro aberto

--detach

Se existir, desligue do terminal de controlo (apenas para Linux)

--force-reload

Forçar reabertura de todos os livros abertos

--full-screen, --fullscreen, -f

Se especificado a janela do visualizador de ebooks irá tentar abrir em modo de ecrã inteiro quando iniciada.

--help, -h

mostrar esta mensagem de ajuda e sair

--open-at

A posição em que será aberto o livro especificado. A posição é um local ou posição que pode obter usando a ação Ir para->Local nos controlos do visualizador. Em alternativa, pode usar a expressão toc:algumacoisa e o livro será aberto no local da primeira entrada do índice que contenha o texto "algumacoisa". A expressão toc--href:algumacoisa irá corresponder ao href (hiperligação de destino interno) das entradas do índice. A correspondência é exata. Caso queira fazer a correspondência apenas a uma parte do texto, use a expressão de pesquisa toc-href-contains:algumacoisa. Já a expressão ref:algumacoisa usará posições do Modo de referência. A expressão search:algumacoisa irá procurar por "algumacoisa" após abrir o livro. A expressão regex:algumacoisa irá procurar a expressão regular "algumacoisa" após abrir o livro.

--raise-window

Se especificado a janela do visualizador de ebooks irá tentar vir para a frente quando iniciada.

--version

mostrar o número da versão do programa e sair

13.1.12 fetch-ebook-metadata

```
fetch-ebook-metadata [options]
```

Obter metadados para livros a partir de fontes em linha. Deve especificar pelo menos um título, autor ou ISBN.

Sempre que passar argumentos para o **fetch-ebook-metadata** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--allowed-plugin, -p

Especificar o nome de um plugin de obtenção de metadados a ser usado. Por predefinição, serão usados todos os plugins de metadados. Pode ser especificado várias vezes para vários plugins. Lista de nomes de plugins: Google, Google Images, Amazon.com, Edelweiss, Open Library, Big Book Search

--authors, -a

Autor do livro

--cover, -c

Especifique o nome do ficheiro. A capa, se disponível, irá ser guardada nele. Sem esta opção, nenhuma capa será descarregada.

--help, -h

mostrar esta mensagem de ajuda e sair

--identifier, -I

Identificadores como ASIN/ID Goodreads, etc. Pode ser especificado várias vezes para identificadores múltiplos. Exemplo:*--identifier* (página 333) asin:B0082BAJA0

--isbn, -i

ISBN do livro

--opf, -o

Criar metadados em formato OPF em vez de texto legível para humanos.

--timeout, -d

Tempo de espera em segundos. A predefinição é 30

--title, -t

Título do livro

--verbose, -v

Escrever o registo para a consola (sdterr)

--version

mostrar o número da versão do programa e sair

13.1.13 lrf2lrs

```
lrf2lrs book.lrf
```

Converte um ficheiro LRF num ficheiro LRS (XML codificado em UTF-8)

Sempre que passar argumentos para o **lrf2lrs** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--dont-output-resources

Não exportar as imagens e os ficheiros de tipo de letra incorporados

--help, -h

mostrar esta mensagem de ajuda e sair

--output, -o

Saída LRS

--verbose

Seja mais detalhado

--version

mostrar o número da versão do programa e sair

13.1.14 lrfviewer

```
lrfviewer [opções] book.lrf
```

Mostrar/editar ebook em LRF book.lrf.

Sempre que passar argumentos para o **lrfviewer** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--disable-hyphenation
Desativar translineação. Deve melhorar consideravelmente a velocidade da sintetização.

--help, -h
mostrar esta mensagem de ajuda e sair

--profile
Criar perfil do sintetizador de ficheiros LRF

--verbose
Mostrar mais informações sobre o processo de sintetização.

--version
mostrar o número da versão do programa e sair

--visual-debug
Ativar ajudas visuais para depurar erros do motor de sintetização.

--white-background
Por predefinição o fundo é acinzentado uma vez que se acha que tal é mais relaxante para os olhos. Use esta opção para tornar o fundo branco.

13.1.15 lrs2lrf

```
lrs2lrf [options] file.lrs
```

Compilar um ficheiro LRS num ficheiro LRF.

Sempre que passar argumentos para o **lrs2lrf** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--help, -h
mostrar esta mensagem de ajuda e sair

--lrs
Converter ficheiro LRS para LRS, útil para depurar.

--output, -o
Caminho para o ficheiro de destino

--verbose

Processamento com mais indicações

--version

mostrar o número da versão do programa e sair

13.1.16 web2disk

```
web2disk URL
```

Quando o URL é, por exemplo, https://google.com

Sempre que passar argumentos para o **web2disk** que contenham espaços ponha-os entre aspas. Por exemplo: «/some path/with spaces»

[opções]

--base-dir, -d

Pasta base onde o URL é guardado. A predefinição é .

--delay

Intervalo mínimo em segundos entre recolhas consecutivas. A predefinição é 0 s

--dont-download-stylesheets

Não descarregar folhas de estilos CSS.

--encoding

A codificação de caracteres para os sítios que está a tentar descarregar. A predefinição é tentar descobrir a codificação.

--filter-regexp

Qualquer ligação que corresponda a esta expressão regular será ignorada. Esta opção pode ser especificada várias vezes, neste caso, qualquer expressão regular que corresponda a uma ligação, será ignorada. Por predefinição, nenhuma ligação será ignorada. Se definir ao mesmo tempo filtrar expressão regular e combinar expressão regular, então filtrar expressão regular será aplicado primeiro.

--help, -h

mostrar esta mensagem de ajuda e sair

--match-regexp

Só os atalhos que correspondam a esta expressão regular é que serão seguidos. Esta opção pode ser especificada várias vezes, nesse caso enquanto um atalho corresponder a alguma expressão regular, ele será seguido. A predefinição é seguir todos os atalhos.

--max-files, -n

O número máximo de ficheiros para descarregar. Isto só se aplica a ficheiros com etiquetas <a href>. A predefinição é 9223372036854775807

--max-recursions, -r

O número máximo de níveis a seguir recursivamente, i.e. a profundidade de atalhos a seguir. A predefinição é 1

--timeout, -t

O tempo de espera em segundos para aguardar por uma resposta do servidor. A predefinição é: 10.0 sec.

--verbose

Mostrar informação de saída detalhada. Útil para depurar.

--version

mostrar o número da versão do programa e sair

13.2 Comandos não documentados

- ebook-device
- markdown-calibre

Pode saber como utilizar os comandos não documentados executando-os sem argumentos num terminal.

CAPÍTULO 14

Criar um ambiente de desenvolvimento para o Calibre

calibre is completely open source, licensed under the GNU GPL v3[109]. This means that you are free to download and modify the program to your heart's content. In this section, you will learn how to get a calibre development environment set up on the operating system of your choice. calibre is written primarily in Python[110] with some C/C++ code for speed and system interfacing. Note that calibre requires at least Python 3.8.

Conteúdos

- *Filosofia de design* (página 338)
  - *Visual de código* (página 338)
- *A obter o código* (página 339)
  - *A submeter as suas alterações a serem incluídas* (página 339)
- *Ambiente de desenvolvimento Windows* (página 340)
- *Ambiente de desenvolvimento macOS* (página 341)
- *Ambiente de desenvolvimento Linux* (página 342)
- *Having separate «normal» and «development» calibre installs on the same computer* (página 342)
- *Dicas de depuração* (página 343)
  - *Usar declarações de impressão* (página 343)
  - *Usando um interpretador* Python *interativo* (página 343)
  - *Usando o depurador* Python *como depurador remoto* (página 343)
  - *Usar o depurador no seu IDE Python favorito* (página 344)
  - *Executing arbitrary scripts in the calibre Python environment* (página 344)
- *Usar Calibre nos seus projetos* (página 345)

[109] https://www.gnu.org/licenses/gpl.html
[110] https://www.python.org

  - *Instalação por binários do Calibre* (página 345)
  - *Instalação de origem em Linux* (página 345)
- *API documentation for various parts of calibre* (página 345)

14.1 Filosofia de design

O Calibre tem as suas raízes no mundo Unix, o que se traduz por um design altamente modular. Os módulos interagem uns com os outros através de interfaces bem definidas. Tal torna o adicionar de novas funcionalidades e a correção de erros de programação muito fáceis no Calibre, resultando num ritmo frenético de desenvolvimento. Por causa das suas raízes, o Calibre tem uma interface de linha de comando muito completa para todas as suas funções, documentadas em *Interface de linha de comandos* (página 295).

The modular design of calibre is expressed via `Plugins`. There is a *tutorial* (página 249) on writing calibre plugins. For example, adding support for a new device to calibre typically involves writing less than a 100 lines of code in the form of a device driver plugin. You can browse the built-in drivers[111]. Similarly, adding support for new conversion formats involves writing input/output format plugins. Another example of the modular design is the *recipe system* (página 29) for fetching news. For more examples of plugins designed to add features to calibre, see the Index of plugins[112].

14.1.1 Visual de código

All the calibre Python code is in the `calibre` package. This package contains the following main sub-packages

- dispositivos - Todos os drivers de dispositivo. Olhe alguns dos drivers embutidos para ter uma ideia de como eles funcionam.
  - For details, see: `devices.interface` which defines the interface supported by device drivers and `devices.usbms` which defines a generic driver that connects to a USBMS device. All USBMS based drivers in calibre inherit from it.
- e-books - All the e-book conversion/metadata code. A good starting point is `calibre.ebooks.conversion.cli` which is the module powering the **`ebook-convert`** command. The conversion process is controlled via `conversion.plumber`. The format independent code is all in `ebooks.oeb` and the format dependent code is in `ebooks.format_name`.
  - A leitura, escrita e download de metadados estão em `ebooks.metadata`
  - Conversion happens in a pipeline, for the structure of the pipeline, see *Introdução* (página 57). The pipeline consists of an input plugin, various transforms and an output plugin. The code that constructs and drives the pipeline is in `plumber.py`. The pipeline works on a representation of an e-book that is like an unzipped epub, with manifest, spine, toc, guide, html content, etc. The class that manages this representation is OEBBook in `ebooks.oeb.base`. The various transformations that are applied to the book during conversions live in `oeb/transforms/*.py`. And the input and output plugins live in `conversion/plugins/*.py`.
  - E-book editing happens using a different container object. It is documented in *Documentação da API para as ferramentas de edição de ebooks* (página 352).
- db - The database back-end. See *Documentação da API para a interface do banco de dados* (página 345) for the interface to the calibre library.
- Content server: `srv` is the calibre Content server.

[111] https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices
[112] https://www.mobileread.com/forums/showthread.php?p=1362767#post1362767

- gui2 - The Graphical User Interface. GUI initialization happens in `gui2.main` and `gui2.ui`. The e-book-viewer is in `gui2.viewer`. The e-book editor is in `gui2.tweak_book`.

If you want to locate the entry points for all the various calibre executables, look at the `entry_points` structure in linux.py[113].

If you need help understanding the code, post in the development forum[114] and you will most likely get help from one of calibre's many developers.

14.2 A obter o código

You can get the calibre source code in two ways, using a version control system or directly downloading a tarball[115].

calibre uses Git[116], a distributed version control system. Git is available on all the platforms calibre supports. After installing Git, you can get the calibre source code with the command:

```
git clone https://github.com/kovidgoyal/calibre.git
```

On Windows you will need the complete path name, that will be something like `C:\Program Files\Git\git.exe`.

calibre is a very large project with a very long source control history, so the above can take a while (10 mins to an hour depending on your internet speed).

If you want to get the code faster, the source code for the latest release is always available as an archive[117].

To update a branch to the latest code, use the command:

```
git pull --no-edit
```

Também pode navegar no código em GitHub[118].

14.2.1 A submeter as suas alterações a serem incluídas

If you only plan to make a few small changes, you can make your changes and create a «merge directive» which you can then attach to a ticket in the calibre bug tracker[119]. To do this, make your changes, then run:

```
git commit -am "Comment describing your changes"
git format-patch origin/master --stdout > my-changes
```

This will create a `my-changes` file in the current folder, simply attach that to a ticket on the calibre bug tracker[120]. Note that this will include *all* the commits you have made. If you only want to send some commits, you have to change `origin/master` above. To send only the last commit, use:

```
git format-patch HEAD~1 --stdout > my-changes
```

To send the last *n* commits, replace *1* with *n*, for example, for the last 3 commits:

[113] https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.py
[114] https://www.mobileread.com/forums/forumdisplay.php?f=240
[115] https://calibre-ebook.com/dist/src
[116] https://www.git-scm.com/
[117] https://calibre-ebook.com/dist/src
[118] https://github.com/kovidgoyal/calibre
[119] https://bugs.launchpad.net/calibre
[120] https://bugs.launchpad.net/calibre

```
git format-patch HEAD~3 --stdout > my-changes
```

Be careful to not include merges when using `HEAD~n`.

If you plan to do a lot of development on calibre, then the best method is to create a GitHub[121] account. Below is a basic guide to setting up your own fork of calibre in a way that will allow you to submit pull requests for inclusion into the main calibre repository:

- Setup git on your machine as described in this article: Setup Git[122]
- Setup ssh keys for authentication to GitHub, as described here: Generating SSH keys[123]
- Go to https://github.com/kovidgoyal/calibre and click the *Fork* button.
- Num terminal execute:

  ```
  git clone git@github.com:<username>/calibre.git
  git remote add upstream https://github.com/kovidgoyal/calibre.git
  ```

  Replace <username> above with your GitHub username. That will get your fork checked out locally.

- You can make changes and commit them whenever you like. When you are ready to have your work merged, do a:

  ```
  git push
  ```

  and go to `https://github.com/<username>/calibre` and click the *Pull Request* button to generate a pull request that can be merged.

- You can update your local copy with code from the main repo at any time by doing:

  ```
  git pull upstream
  ```

You should also keep an eye on the calibre development forum[124]. Before making major changes, you should discuss them in the forum or contact Kovid directly (his email address is all over the source code).

14.3 Ambiente de desenvolvimento Windows

Nota: You must also get the calibre source code separately as described above.

Install calibre normally, using the Windows installer[125]. Then open a Command Prompt and change to the previously checked out calibre code folder. For example:

```
cd C:\Users\kovid\work\calibre
```

calibre is the folder that contains the src and resources sub-folders.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src folder. So, following the example above, it would be `C:\Users\kovid\work\calibre\src`. Here is a short guide[126] to setting environment variables on Windows.

[121] https://github.com
[122] https://help.github.com/articles/set-up-git
[123] https://help.github.com/articles/generating-ssh-keys
[124] https://www.mobileread.com/forums/forumdisplay.php?f=240
[125] https://calibre-ebook.com/download_windows
[126] https://docs.python.org/using/windows.html#excursus-setting-environment-variables

Once you have set the environment variable, open a new command prompt and check that it was correctly set by using the command:

```
echo %CALIBRE_DEVELOP_FROM%
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file `src\calibre\__init__.py` in your favorite editor and add the line:

```
print ("Hello, world!")
```

near the top of the file. Now run the command **`calibredb`**. The very first line of output should be `Hello, world!`.

You can also setup a calibre development environment inside the free Microsoft Visual Studio, if you like, following the instructions here[127].

14.4 Ambiente de desenvolvimento macOS

Nota: You must also get the calibre source code separately as described above.

Install calibre normally using the provided .dmg[128]. Then open a Terminal and change to the previously checked out calibre code folder, for example:

```
cd /Users/kovid/work/calibre
```

calibre is the folder that contains the src and resources sub-folders. The calibre command line tools are found inside the calibre app bundle, in `/Applications/calibre.app/Contents/MacOS` you should add this folder to your PATH environment variable, if you want to run the command line tools easily.

The next step is to create a bash script that will set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src folder when running calibre in debug mode.

Criar um ficheiro de texto simples:

```
#!/bin/sh
export CALIBRE_DEVELOP_FROM="/Users/kovid/work/calibre/src"
calibre-debug -g
```

Guardar este ficheiro como `/usr/local/bin/calibre-develop`, e depois defina as suas permissões para que possa ser executado:

```
chmod +x /usr/local/bin/calibre-develop
```

Quando tiver feito isto, execute:

```
calibre-develop
```

You should see some diagnostic information in the Terminal window as calibre starts up, and you should see an asterisk after the version number in the GUI window, indicating that you are running from source.

[127] https://www.mobileread.com/forums/showthread.php?t=251201
[128] https://calibre-ebook.com/download_osx

14.5 Ambiente de desenvolvimento Linux

Nota: You must also get the calibre source code separately as described above.

calibre is primarily developed on Linux. You have two choices in setting up the development environment. You can install the calibre binary as normal and use that as a runtime environment to do your development. This approach is similar to that used in Windows and macOS. Alternatively, you can install calibre from source. Instructions for setting up a development environment from source are in the INSTALL file in the source tree. Here we will address using the binary as a runtime, which is the recommended method.

Install calibre using the binary installer[129]. Then open a terminal and change to the previously checked out calibre code folder, for example:

```
cd /home/kovid/work/calibre
```

calibre is the folder that contains the src and resources sub-folders.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src folder. So, following the example above, it would be `/home/kovid/work/calibre/src`. How to set environment variables depends on your Linux distribution and what shell you are using.

Nota: It is recommended to use the binary installer provided from upstream. Should you insist on using a package provided by your distribution, use the `CALIBRE_PYTHON_PATH` and `CALIBRE_RESOURCES_PATH` variables instead.

Depois de definir a variável de ambiente, abra uma nova janela de terminal e verifique se foi configurado corretamente usando o comando:

```
echo $CALIBRE_DEVELOP_FROM
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file `src/calibre/__init__.py` in your favorite editor and add the line:

```
print ("Hello, world!")
```

near the top of the file. Now run the command **`calibredb`**. The very first line of output should be `Hello, world!`.

14.6 Having separate «normal» and «development» calibre installs on the same computer

The calibre source tree is very stable and rarely breaks, but if you feel the need to run from source on a separate test library and run the released calibre version with your everyday library, you can achieve this easily using .bat files or shell scripts to launch calibre. The example below shows how to do this on Windows using .bat files (the instructions for other platforms are the same, just use a shell script instead of a .bat file)

To launch the release version of calibre with your everyday library:

calibre-normal.bat:

[129] https://calibre-ebook.com/download_linux

```
calibre.exe "--with-library=C:\path\to\everyday\library folder"
```

calibre-dev.bat:

```
set CALIBRE_DEVELOP_FROM=C:\path\to\calibre\checkout\src
calibre.exe "--with-library=C:\path\to\test\library folder"
```

14.7 Dicas de depuração

Python is a dynamically typed language with excellent facilities for introspection. Kovid wrote the core calibre code without once using a debugger. There are many strategies to debug calibre code:

14.7.1 Usar declarações de impressão

This is Kovid's favorite way to debug. Simply insert print statements at points of interest and run your program in the terminal. For example, you can start the GUI from the terminal as:

```
calibre-debug -g
```

Similarly, you can start the E-book viewer as:

```
calibre-debug -w /path/to/file/to/be/viewed
```

O editor de ebook pode ser iniciado como:

```
calibre-debug -t /path/to/be/edited
```

14.7.2 Usando um interpretador *Python* interativo

É possível inserir as duas linhas de código a seguir para iniciar uma sessão *Python* interativa nesse ponto:

```
from calibre import ipython
ipython(locals())
```

When running from the command line, this will start an interactive Python interpreter with access to all locally defined variables (variables in the local scope). The interactive prompt even has `Tab` completion for object properties and you can use the various Python facilities for introspection, such as `dir()`, `type()`, `repr()`, etc.

14.7.3 Usando o depurador *Python* como depurador remoto

You can use the builtin Python debugger (pdb) as a remote debugger from the command line. First, start the remote debugger at the point in the calibre code you are interested in, like this:

```
from calibre.rpdb import set_trace
set_trace()
```

Then run calibre, either as normal, or using one of the calibre-debug commands described in the previous section. Once the above point in the code is reached, calibre will freeze, waiting for the debugger to connect.

Agora abra uma janela de terminal ou Prompt de Comando e use o seguinte comando para iniciar a sessão de depuração:

```
calibre-debug -c "from calibre.rpdb import cli; cli()"
```

You can read about how to use the Python debugger in the Python stdlib docs for the pdb module[130].

Nota: By default, the remote debugger will try to connect on port 4444. You can change it, by passing the port parameter to both the set_trace() and the cli() functions above, like this: `set_trace(port=1234)` and `cli(port=1234)`.

Nota: The Python debugger cannot handle multiple threads, so you have to call set_trace once per thread, each time with a different port number.

14.7.4 Usar o depurador no seu IDE Python favorito

It is possible to use the builtin debugger in your favorite Python IDE, if it supports remote debugging. The first step is to add the calibre src checkout to the `PYTHONPATH` in your IDE. In other words, the folder you set as `CALIBRE_DEVELOP_FROM` above, must also be in the `PYTHONPATH` of your IDE.

Then place the IDE's remote debugger module into the `src` sub-folder of the calibre source code checkout. Add whatever code is needed to launch the remote debugger to calibre at the point of interest, for example in the main function. Then run calibre as normal. Your IDE should now be able to connect to the remote debugger running inside calibre.

14.7.5 Executing arbitrary scripts in the calibre Python environment

The **`calibre-debug`** command provides a couple of handy switches to execute your own code, with access to the calibre modules:

```
calibre-debug -c "some Python code"
```

is great for testing a little snippet of code on the command line. It works in the same way as the -c switch to the Python interpreter:

```
calibre-debug myscript.py
```

can be used to execute your own Python script. It works in the same way as passing the script to the Python interpreter, except that the calibre environment is fully initialized, so you can use all the calibre code in your script. To use command line arguments with your script, use the form:

```
calibre-debug myscript.py -- --option1 arg1
```

The `--` causes all subsequent arguments to be passed to your script.

[130] https://docs.python.org/library/pdb.html#debugger-commands

14.8 Usar Calibre nos seus projetos

É possível usar funções/código do *Calibre* diretamente em seu projeto *Python*. Existem duas maneiras de fazer isso:

14.8.1 Instalação por binários do Calibre

If you have a binary install of calibre, you can use the Python interpreter bundled with calibre, like this:

```
calibre-debug /path/to/your/python/script.py -- arguments to your script
```

14.8.2 Instalação de origem em Linux

In addition to using the above technique, if you do a source install on Linux, you can also directly import calibre, as follows:

```
import init_calibre
import calibre

print calibre.__version__
```

It is essential that you import the init_calibre module before any other calibre modules/packages as it sets up the interpreter to run calibre code.

14.9 API documentation for various parts of calibre

14.9.1 Documentação da API para a interface do banco de dados

Esta API é thread-safe (ela usa um esquema de bloqueio de múltiplos leitores e gravador único). É possível acessar esta API assim:

```
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

Se você estiver em um plug-in do *Calibre* que faz parte da GUI do *Calibre* principal, você terá acesso a ele assim:

```
db = self.gui.current_db.new_api
```

class calibre.db.cache.**Cache**(*backend*, *library_database_instance=None*)

Um in-memory cache do arquivo metadata.db da biblioteca calibre. Essa classe também serve como um threadsafe API para acessar o banco de dados. O in-memory cache é mantido na forma normal para o máximo desempenho

SQLITE is simply used as a way to read and write from metadata.db robustly. All table reading/sorting/searching/caching logic is re-implemented. This was necessary for maximum performance and flexibility.

class EventType(*value*)

An enumeration.

book_created = 4

When a new book record is created in the database, with the book id as the only argument

book_edited = 8

When a book format is edited, with arguments: (book_id, fmt)

books_removed = 5

When books are removed from the database with the list of book ids as the only argument

format_added = 2

When a format is added to a book, with arguments: (book_id, format)

formats_removed = 3

When formats are removed from a book, with arguments: (mapping of book id to set of formats removed from the book)

indexing_progress_changed = 9

When the indexing progress changes

items_removed = 7

When items such as tags or authors are removed from some books. Arguments: (field_name, affected book ids, ids of removed items)

items_renamed = 6

When items such as tags or authors are renamed in some or all books. Arguments: (field_name, affected book ids, map of old item id to new item id)

metadata_changed = 1

When some metadata is changed for some books, with arguments: (name of changed field, set of affected book ids)

add_books (*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)

Add the specified books to the library. Books should be an iterable of 2-tuples, each 2-tuple of the form `(mi, format_map)` where mi is a Metadata object and format_map is a dictionary of the form `{fmt: path_or_stream}`, for example: `{'EPUB': '/path/to/file.epub'}`.

Returns a pair of lists: `ids, duplicates`. `ids` contains the book ids for all newly created books in the database. `duplicates` contains the `(mi, format_map)` for all books that already exist in the database as per the simple duplicate detection heuristic used by *`has_book()`* (página 350).

add_custom_book_data (*name*, *val_map*, *delete_first=False*)

Adicionar dado para nome onde val_map é um mapa de valores de book_ids. Se delete_first é verdadeiro, todos o dados de anteriores para o nome serão removidos

add_format (*book_id*, *fmt*, *stream_or_path*, *replace=True*, *run_hooks=True*, *dbapi=None*)

Add a format to the specified book. Return True if the format was added successfully.

Parâmetros

- **replace** – Se Verdadeiro substitui o formato existente; se não, se o formato já existe, devolve False.
- **run_hooks** – Se verdadeiro, os plugins de tipo de ficheiro são executados no formato antes e depois de serem adicionados.
- **dbapi** – Apenas para utilização interna.

add_listener (*event_callback_function*, *check_already_added=False*)

Register a callback function that will be called after certain actions are taken on this database. The function must take three arguments: (*`EventType`* (página 345), library_id, event_type_specific_data)

all_book_ids(*type=<class 'frozenset'>*)

Frozen set of all known book ids.

all_field_for(*field*, *book_ids*, *default_value=None*)

O mesmo que field_for, exceto que opera em múltiplos livros de uma só vez

all_field_ids(*name*)

Frozen set of ids for all values in the field `name`.

all_field_names(*field*)

Frozen set of all fields names (should only be used for many-one and many-many fields)

author_data(*author_ids=None*)

Retorna os dados do autor como um dicionário com chaves: nome, tipo, ligação

If no authors with the specified ids are found an empty dictionary is returned. If author_ids is None, data for all authors is returned.

author_sort_from_authors(*authors*, *key_func=<function make_change_case_func.<locals>.change_case>*)

Given a list of authors, return the author_sort string for the authors, preferring the author sort associated with the author over the computed string.

books_for_field(*name*, *item_id*)

Return all the books associated with the item identified by `item_id`, where the item belongs to the field `name`.

Returned value is a set of book ids, or the empty set if the item or the field does not exist.

books_in_virtual_library(*vl*, *search_restriction=None*, *virtual_fields=None*)

Devolver o conjunto de livros na biblioteca virtual especificada

compress_covers(*book_ids*, *jpeg_quality=100*, *progress_callback=None*)

Compress the cover images for the specified books. A compression quality of 100 will perform lossless compression, otherwise lossy compression.

The progress callback will be called with the book_id and the old and new sizes for each book that has been processed. If an error occurs, the new size will be a string with the error details.

copy_cover_to(*book_id*, *dest*, *use_hardlink=False*, *report_file_size=None*)

Copy the cover to the file like object `dest`. Returns False if no cover exists or dest is the same file as the current cover. dest can also be a path in which case the cover is copied to it if and only if the path is different from the current path (taking case sensitivity into account).

copy_format_to(*book_id*, *fmt*, *dest*, *use_hardlink=False*, *report_file_size=None*)

Copy the format `fmt` to the file like object `dest`. If the specified format does not exist, raises `NoSuchFormat` error. dest can also be a path (to a file), in which case the format is copied to it, iff the path is different from the current path (taking case sensitivity into account).

cover(*book_id*, *as_file=False*, *as_image=False*, *as_path=False*, *as_pixmap=False*)

Retorna a imagem de capa ou nada. Por padrão, retorna a capa como um bytestring.

WARNING: Using as_path will copy the cover to a temp file and return the path to the temp file. You should delete the temp file when you are done with it.

Parâmetros

- **as_file** – If True return the image as an open file object (a SpooledTemporaryFile)
- **as_image** – Se verdadeiro retorna a imagem como um objeto QImage.

- **as_pixmap** – If True return the image as a QPixmap object
- **as_path** – Se verdadeiro retorna a imagem como um caminho para um arquivo temporário.

data_for_find_identical_books()

Dados de retorno que podem ser usados para implementar *find_identical_books()* (página 348) num processo sem acesso à BD. Veja db.utils para uma implementação.

data_for_has_book()

Return data suitable for use in *has_book()* (página 350). This can be used for an implementation of *has_book()* (página 350) in a worker process without access to the db.

delete_custom_book_data(*name*, *book_ids=()*)

Delete data for name. By default deletes all data, if you only want to delete data for some book ids, pass in a list of book ids.

embed_metadata(*book_ids*, *only_fmts=None*, *report_error=None*, *report_progress=None*)

Update metadata in all formats of the specified book_ids to current metadata in the database.

fast_field_for(*field_obj*, *book_id*, *default_value=None*)

Same as field_for, except that it avoids the extra lookup to get the field object

field_for(*name*, *book_id*, *default_value=None*)

Return the value of the field `name` for the book identified by `book_id`. If no such book exists or it has no defined value for the field `name` or no such field exists, then `default_value` is returned.

`default_value` is not used for title, title_sort, authors, author_sort and series_index. This is because these always have values in the db. `default_value` is used for all custom columns.

The returned value for is_multiple fields are always tuples, even when no values are found (in other words, default_value is ignored). The exception is identifiers for which the returned value is always a dictionary. The returned tuples are always in link order, that is, the order in which they were created.

field_ids_for(*name*, *book_id*)

Return the ids (as a tuple) for the values that the field `name` has on the book identified by `book_id`. If there are no values, or no such book, or no such field, an empty tuple is returned.

find_identical_books(*mi*, *search_restriction=''*, *book_ids=None*)

Finds books that have a superset of the authors in mi and the same title (title is fuzzy matched). See also *data_for_find_identical_books()* (página 348).

format(*book_id*, *fmt*, *as_file=False*, *as_path=False*, *preserve_filename=False*)

Return the e-book format as a bytestring or *None* if the format doesn't exist, or we don't have permission to write to the e-book file.

Parâmetros

- **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()* (página 347) instead for performance.
- **as_path** – Copies the format file to a temp file and returns the path to the temp file
- **preserve_filename** – If True and returning a path the filename is the same as that used in the library. Note that using this means that repeated calls yield the same temp file (which is re-created each time)

format_abspath(*book_id*, *fmt*)

Return absolute path to the e-book file of format *format*. You should almost never use this, as it breaks the threadsafe promise of this API. Instead use, *copy_format_to()* (página 347).

Currently used only in calibredb list, the viewer, edit book, compare_format to original format, open with, bulk metadata edit and the catalogs (via get_data_as_dict()).

Apart from the viewer, open with and edit book, I don't believe any of the others do any file write I/O with the results of this call.

format_hash(*book_id*, *fmt*)

Return the hash of the specified format for the specified book. The kind of hash is backend dependent, but is usually SHA-256.

format_metadata(*book_id*, *fmt*, *allow_cache=True*, *update_db=False*)

Return the path, size and mtime for the specified format for the specified book. You should not use path unless you absolutely have to, since accessing it directly breaks the threadsafe guarantees of this API. Instead use the *copy_format_to()* (página 347) method.

Parâmetros

- **allow_cache** – If `True` cached values are used, otherwise a slow filesystem access is done. The cache values could be out of date if access was performed to the filesystem outside of this API.
- **update_db** – If `True` The max_size field of the database is updated for this book.

formats(*book_id*, *verify_formats=True*)

Return tuple of all formats for the specified book. If verify_formats is True, verifies that the files exist on disk.

get_categories(*sort='name'*, *book_ids=None*, *already_fixed=None*, *first_letter_sort=False*)

Used internally to implement the Tag Browser

get_custom_book_data(*name*, *book_ids=()*, *default=None*)

Get data for name. By default returns data for all book_ids, pass in a list of book ids if you only want some data. Returns a map of book_id to values. If a particular value could not be decoded, uses default for it.

get_id_map(*field*)

Return a mapping of id numbers to values for the specified field. The field must be a many-one or many-many field, otherwise a ValueError is raised.

get_ids_for_custom_book_data(*name*)

Return the set of book ids for which name has data.

get_item_id(*field*, *item_name*)

Return the item id for item_name (case-insensitive) or None if not found

get_item_ids(*field*, *item_names*)

Return the item id for item_name (case-insensitive)

get_item_name(*field*, *item_id*)

Return the item name for the item specified by item_id in the specified field. See also *get_id_map()* (página 349).

get_metadata(*book_id*, *get_cover=False*, *get_user_categories=True*, *cover_as_data=False*)

Return metadata for the book identified by book_id as a *calibre.ebooks.metadata.book.base.Metadata* (página 204) object. Note that the list of formats is not verified. If get_cover is True, the cover is returned, either a path to temp file as mi.cover or if cover_as_data is True then as mi.cover_data.

get_next_series_num_for(*series*, *field='series'*, *current_indices=False*)

Return the next series index for the specified series, taking into account the various preferences that control next series number generation.

Parâmetros

- **field** – The series-like field (defaults to the builtin series column)
- **current_indices** – If True, returns a mapping of book_id to current series_index value instead.

get_proxy_metadata(*book_id*)

Like *get_metadata()* (página 349) except that it returns a ProxyMetadata object that only reads values from the database on demand. This is much faster than get_metadata when only a small number of fields need to be accessed from the returned metadata object.

get_usage_count_by_id(*field*)

Return a mapping of id to usage count for all values of the specified field, which must be a many-one or many-many field.

has_book(*mi*)

Return True iff the database contains an entry with the same title as the passed in Metadata object. The comparison is case-insensitive. See also *data_for_has_book()* (página 348).

has_format(*book_id*, *fmt*)

Retorna Verdadeiro se o formato existir no disco.

has_id(*book_id*)

Retorna Verdadeiro se o id do livro especificado existir na base de dados.

init()

Initialize this cache with data from the backend.

multisort(*fields*, *ids_to_sort=None*, *virtual_fields=None*)

Return a list of sorted book ids. If ids_to_sort is None, all book ids are returned.

fields must be a list of 2-tuples of the form (field_name, ascending=True or False). The most significant field is the first 2-tuple.

pref(*name*, *default=None*, *namespace=None*)

Return the value for the specified preference or the value specified as `default` if the preference is not set.

read_backup(*book_id*)

Return the OPF metadata backup for the book as a bytestring or None if no such backup exists.

remove_books(*book_ids*, *permanent=False*)

Remove os livros, especificados pelos book_ids, da base de dados e apaga os respetivos ficheiros de formatos. Se `permanent` for False, então os ficheiros de formatos serão colocados na Reciclagem.

remove_formats(*formats_map*, *db_only=False*)

Remover os formatos especificados dos livros indicados.

Parâmetros

- **formats_map** – A mapping of book_id to a list of formats to be removed from the book.
- **db_only** – If True, only remove the record for the format from the db, do not delete the actual format file from the filesystem.

remove_items (*field*, *item_ids*, *restrict_to_book_ids=None*)

Delete all items in the specified field with the specified ids. Returns the set of affected book ids. `restrict_to_book_ids` is an optional set of books ids. If specified the items will only be removed from those books.

rename_items (*field*, *item_id_to_new_name_map*, *change_index=True*, *restrict_to_book_ids=None*)

Rename items from a many-one or many-many field such as tags or series.

Parâmetros

- **change_index** – When renaming in a series-like field also change the series_index values.
- **restrict_to_book_ids** – An optional set of book ids for which the rename is to be performed, defaults to all books.

restore_book (*book_id*, *mi*, *last_modified*, *path*, *formats*, *annotations=()*)

Restore the book entry in the database for a book that already exists on the filesystem

restore_original_format (*book_id*, *original_fmt*)

Restore the specified format from the previously saved ORIGINAL_FORMAT, if any. Return True on success. The ORIGINAL_FORMAT is deleted after a successful restore.

property safe_read_lock

A safe read lock is a lock that does nothing if the thread already has a write lock, otherwise it acquires a read lock. This is necessary to prevent DowngradeLockErrors, which can happen when updating the search cache in the presence of composite columns. Updating the search cache holds an exclusive lock, but searching a composite column involves reading field values via ProxyMetadata which tries to get a shared lock. There may be other scenarios that trigger this as well.

This property returns a new lock object on every access. This lock object is not recursive (for performance) and must only be used in a with statement as `with cache.safe_read_lock:` otherwise bad things will happen.

save_original_format (*book_id*, *fmt*)

Save a copy of the specified format as ORIGINAL_FORMAT, overwriting any existing ORIGINAL_FORMAT.

search (*query*, *restriction=''*, *virtual_fields=None*, *book_ids=None*)

Search the database for the specified query, returning a set of matched book ids.

Parâmetros

- **restriction** – A restriction that is ANDed to the specified query. Note that restrictions are cached, therefore the search for a AND b will be slower than a with restriction b.
- **virtual_fields** – Used internally (virtual fields such as on_device to search over).
- **book_ids** – If not None, a set of book ids for which books will be searched instead of searching all books.

set_conversion_options (*options*, *fmt='PIPE'*)

options must be a map of the form {book_id:conversion_options}

set_cover (*book_id_data_map*)

Set the cover for this book. The data can be either a QImage, QPixmap, file object or bytestring. It can also be None, in which case any existing cover is removed.

set_field (*name*, *book_id_to_val_map*, *allow_case_change=True*, *do_path_update=True*)

Set the values of the field specified by `name`. Returns the set of all book ids that were affected by the change.

Parâmetros

- **book_id_to_val_map** – Mapping of book_ids to values that should be applied.
- **allow_case_change** – If True, the case of many-one or many-many fields will be changed. For example, if a book has the tag `tag1` and you set the tag for another book to `Tag1` then the both books will have the tag `Tag1` if allow_case_change is True, otherwise they will both have the tag `tag1`.
- **do_path_update** – Usado internamente. Nunca deve ser alterado.

set_metadata (*book_id*, *mi*, *ignore_errors=False*, *force_changes=False*, *set_title=True*, *set_authors=True*, *allow_case_change=False*)

Set metadata for the book *id* from the *Metadata* object *mi*

Setting force_changes=True will force set_metadata to update fields even if mi contains empty values. In this case, "None" is distinguished from "empty". If mi.XXX is None, the XXX is not replaced, otherwise it is. The tags, identifiers, and cover attributes are special cases. Tags and identifiers cannot be set to None so they will always be replaced if force_changes is true. You must ensure that mi contains the values you want the book to have. Covers are always changed if a new cover is provided, but are never deleted. Also note that force_changes has no effect on setting title or authors.

set_pref (*name*, *val*, *namespace=None*)

Set the specified preference to the specified value. See also *pref()* (página 350).

split_if_is_multiple_composite (*f*, *val*)

If f is a composite column lookup key and the column is is_multiple then split v into unique non-empty values. The comparison is case sensitive. Order is not preserved. Return a list() for compatibility with proxy metadata field getters, for example tags.

tags_older_than (*tag*, *delta=None*, *must_have_tag=None*, *must_have_authors=None*)

Return the ids of all books having the tag `tag` that are older than the specified time. tag comparison is case insensitive.

Parâmetros

- **delta** – A timedelta object or None. If None, then all ids with the tag are returned.
- **must_have_tag** – If not None the list of matches will be restricted to books that have this tag
- **must_have_authors** – A list of authors. If not None the list of matches will be restricted to books that have these authors (case insensitive).

user_categories_for_books (*book_ids*, *proxy_metadata_map=None*)

Return the user categories for the specified books. proxy_metadata_map is optional and is useful for a performance boost, in contexts where a ProxyMetadata object for the books already exists. It should be a mapping of book_ids to their corresponding ProxyMetadata objects.

14.9.2 Documentação da API para as ferramentas de edição de ebooks

As ferramentas de edição de ebooks consistem num objeto contentor *calibre.ebooks.oeb.polish.container.Container* (página 353) que representa um livro como uma coleção de ficheiros HTML e de recursos, e diversas ferramentas que podem ser usadas para levar a cabo operações sobre o contentor. Todas as ferramentas estão na forma de funções ao nível de módulo nos vários `calibre.ebooks.oeb.polish.*` módulos.

Obtém um objeto contentor para um livro num caminho como este:

```
from calibre.ebooks.oeb.polish.container import get_container
container = get_container('Path to book file', tweak_mode=True)
```

Se estiver a escrever um plugin para o editor de ebooks, pode obter o contentor atual para o livro em edição, do seguinte modo:

```
from calibre.gui2.tweak_book import current_container
container = current_container()
if container is None:
    report_error # No book has been opened yet
```

O objeto contentor

class calibre.ebooks.oeb.polish.container.**Container**(*rootpath*, *opfpath*, *log*, *clone_data=None*)

A container represents an open e-book as a folder full of files and an OPF file. There are two important concepts:

- The root folder. This is the base of the e-book. All the e-books files are inside this folder or in its sub-folders.
- Names: These are paths to the books" files relative to the root folder. They always contain POSIX separators and are unquoted. They can be thought of as canonical identifiers for files in the book. Most methods on the container object work with names. Names are always in the NFC Unicode normal form.
- Clones: o objeto contentor suporta eficientemente a clonagem para disco, o que é usado para implementar pontos de controlo no editor de ebooks. Para que isto funcione, nunca deve aceder diretamente aos ficheiros no sistema de ficheiros. Em vez disso, use *raw_data()* (página 356) ou *open()* (página 355) para leitura/escrita nos ficheiros componentes do livro.

Aquando da conversão entre hrefs e nomes use os métodos fornecidos por esta classe, que assumem que todos os hrefs estão entre aspas.

abspath_to_name(*fullpath*, *root=None*)

Converte um caminho absoluto para um nome canónico relativo ao `root`

Parâmetros

root – A pasta base. Por predefinição, é usada a raiz deste objeto contentor.

add_file(*name*, *data*, *media_type=None*, *spine_index=None*, *modify_name_if_needed=False*, *process_manifest_item=None*)

Adiciona um ficheiro a este contentor. Entradas para o ficheiro são criadas automaticamente no manifesto OPF e na lombada (se o ficheiro for um documento de texto)

add_name_to_manifest(*name*, *process_manifest_item=None*)

Adiciona uma entrada ao manifesto para um ficheiro com o nome especificado. Devolve o ID do manifesto.

add_properties(*name*, **properties*)

Adiciona as propriedades especificadas ao item do manifesto identificado por nome.

apply_unique_properties(*name*, **properties*)

Assegura que as propriedades especificadas são definidas apenas ao item de manifesto identificado por nome. Pode passar None como valor do parâmetro Nome para remover a propriedade de todos os itens.

book_type = 'oeb'

O tipo de livro (epub para ficheiros EPUB e azw3 para ficheiros AZW3)

commit(*outpath=None*, *keep_parsed=False*)

Submeter ao sistema de ficheiros todos os objetos analisados que tenham sido identificados com problemas e escrever o ficheiro de ebook no caminho de saída.

Parâmetros

- **output** – O caminho onde será escrito o ficheiro do ebook guardado. Se for None, é usado o caminho do ficheiro do livro original
- **keep_parsed** – Se Verdadeiro, as representações analisadas dos itens confirmados são mantidas em cache.

commit_item (*name*, *keep_parsed=False*)

Confirma a escrita de um objeto analisado para o disco (será serializado e escrito no ficheiro subjacente). Se `keep_parsed for Verdadeiro a representação analisada é mantida na Cache. Ver também: *parsed()* (página 356)

dirty (*name*)

Marcar o objeto analisado correspondente ao nome como impuro. Ver também: *parsed()* (página 356).

exists (*name*)

Verdadeiro se e só se existir um ficheiro/pasta correspondente ao nome canónico. Note que esta função sofre as limitações subjacentes ao sistema de ficheiros do sistema operativo, em particular no que respeita à (in)sensibilidade com que são tratadas as maiúsculas e minúsculas. Assim, num sistema de ficheiros que não diferencie entre maiúsculas e minúsculas, será devolvido Verdadeiro mesmo se a maiusculização do nome for diferente da que se encontra no ficheiro do sistema de ficheiros subjacente. Ver também: *has_name()* (página 354)

filesize (*name*)

Devolve o tamanho em bytes do ficheiro representado pelo nome canónico especificado. Manipula automaticamente os objetos analisados como impuros. Ver também: *parsed()* (página 356)

generate_item (*name*, *id_prefix=None*, *media_type=None*, *unique_href=True*)

Adiciona um item ao manifesto com um href derivado do nome fornecido. Assegura automaticamente a unicidade do href e do id. Devolve o item gerado.

get_file_path_for_processing (*name*, *allow_modification=True*)

Similar a open() exceto por devolver o caminho de um ficheiro em vez de um objeto de ficheiro aberto.

property guide_type_map

Mapeamento de tipo de guia para o nome canónico

has_name (*name*)

Devolve Verdadeiro se e só se existe um ficheiro com o mesmo nome canónico que o especificado. Ao contrário de *exists()* (página 354) este método é sensível a maiúsculas e minúsculas.

href_to_name (*href*, *base=None*)

Converte um href (relativo à base) para um nome. base tem de ser um nome ou None, caso em que self.root será utilizado.

insert_into_xml (*parent*, *item*, *index=None*)

Insere item no contentor (ou acrescenta se Index for None), ajustando a indentação. Apenas funciona com itens auto-fecháveis.

is_dir = False

Se este contentor representa um livro não comprimido (uma pasta)

iterlinks (*name*, *get_line_numbers=True*)

Itera sobre todas as ligações presentes no nome. Se get_line_numbers for Verdadeiro, os resultados devolvidos estarão na forma (ligação, número_de_linha, offset), onde número_de_linha é o número_de_linha na qual a ligação ocorre e offset é o número de caracteres contado desde o início da linha. Note que offset pode englobar diversas linhas se não for zero.

make_name_unique(*name*)

Assegura que *nome* não existe previamente neste livro. Se existe, devolve uma versão modificada que não existe.

manifest_has_name(*name*)

Devolve Verdadeiro se o manifesto tiver uma entrada correspondente ao nome

property manifest_id_map

Mapeamento de ID do manifesto para nomes canónicos

manifest_items_of_type(*predicate*)

Os nomes de todos os itens de manifesto cujos tipos de media coincidem com o predicado. *predicate* pode ser um conjunto, uma lista, uma string ou uma função de um único argumento, que será chamada pelo tipo de media.

manifest_items_with_property(*property_name*)

Todos os itens de manifesto que contêm a propriedade especificada

property manifest_type_map

Mapeamento do tipo de media do manifesto para lista de nomes canónicos desse tipo de media

property mi

Os metadados deste livro sob a forma de um objeto Metadados. Note que este objeto é construído em tempo real de cada vez que esta propriedade é solicitada, pelo que use com moderação.

name_to_abspath(*name*)

Converte um nome canónico para um caminho absoluto dependente do sistema operativo

name_to_href(*name*, *base=None*)

Converte um nome para um href relativo à base, que tem de ser um nome ou None, caso em que self.root é usado como base

property names_that_must_not_be_changed

Conjunto de nomes que nunca podem ser alterados. Depende do formato do ficheiro do ebook.

property names_that_must_not_be_removed

Conjunto de nomes que nunca podem ser apagados do contentor. Depende do formato de ficheiro do ebook.

property names_that_need_not_be_manifested

Conjunto de nomes que são permitidos estarem em falta no manifesto. Depende do formato do ficheiro do ebook.

open(*name*, *mode='rb'*)

Abre o ficheiro apontado por nome para leitura/escrita direta. Note que isto efetuará as alterações ao ficheiro se o mesmo se encontrar deteriorado e irá removê-lo da cache do interpretador. É necessário terminar o trabalho neste ficheiro antes de aceder novamente à sua versão interpretada, ou coisas estranhas certamente acontecerão.

property opf

O ficheiro OPF interpretado

opf_get_or_create(*name*)

Método de conveniência que devolve o primeiro element XML com o nome especificado ou então cria-o debaixo do elemento opf:package e devolve-o de seguida no caso de ainda não existir.

property opf_version

A versão definida no elemento <package>OPF

property opf_version_parsed

A versão definida no elemento do <package> OPF sob a forma de um énuplo de inteiros

opf_xpath(*expr*)

Método de conveniência para avaliar uma expressão XPath no ficheiro OPF, que tem os prefixos opf; e dc:namespace predefinidos.

parsed(*name*)

Devolve uma representação analisada lexicograficamente do ficheiro especificado pelo nome. Para ficheiros HTML e XML é devolvida uma árvore lxml. Para ficheiros CSS é devolvida uma folha de estilo do tipo css_parser. Note que os objetos analisados ficam em cache para melhorar a performance. Se fizer quaisquer alterações ao objeto analisado, tem de invocar *dirty()* (página 354) de modo a que o contentor saiba que tem de atualizar a cache. Veja também *replace()* (página 356).

raw_data(*name*, *decode=True*, *normalize_to_nfc=True*)

Devolve os dados em bruto correspondentes ao ficheiro especificado por nome

Parâmetros

- **decode** – Se Verdadeiro, e se o ficheiro tiver um tipo de MIME baseado em texto, descodificar e devolver um objeto Unicode em vez de bytes em bruto.
- **normalize_to_nfc** – Se Verdadeiro o objeto Unicode devolvido é normalizado sob a forma normal NFC tal como requerido pelos formatos de ficheiro EPUB e AZW3.

relpath(*path*, *base=None*)

Converte um caminho absoluto (com separadores do SO) para um caminho relativo à base (por omissão, self.root). O caminho relativo *não* é um nome. Use *abspath_to_name()* (página 353) para esse fim.

remove_from_spine(*spine_items*, *remove_if_no_longer_in_spine=True*)

Remove os itens especificados (por nome canónico) da lombada. Se `remove_if_no_longer_in_spine` for Verdadeiro, os itens também são apagados do livro, e não apenas da lombada.

remove_from_xml(*item*)

Remove o item do contentor, ajustando a indentação (funciona apenas com itens que se auto-fecham).

remove_item(*name*, *remove_from_guide=True*)

Remove o item identificado por nome deste contentor. Isto remove todas as referências ao item do manifesto OPF, guia e lombada bem como de todas as caches internas.

rename(*current_name*, *new_name*)

Renames a file from current_name to new_name. It automatically rebases all links inside the file if the folder the file is in changes. Note however, that links are not updated in the other files that could reference this file. This is for performance, such updates should be done once, in bulk.

replace(*name*, *obj*)

Substitui os objetos analisados correspondentes ao nome por obj, que tem de ser um objeto similar, isto é, uma árvore lxml no caso de HTML/XML ou uma folha de estilo css_parser no caso de ficheiros CSS.

replace_links(*name*, *replace_func*)

Replace all links in name using replace_func, which must be a callable that accepts a URL and returns the replaced URL. It must also have a "replaced" attribute that is set to True if any actual replacement is done. Convenient ways of creating such callables are using the `LinkReplacer` and `LinkRebaser` classes.

serialize_item(*name*)

Convert a parsed object (identified by canonical name) into a bytestring. See *parsed()* (página 356).

set_spine(*spine_items*)

Set the spine to be spine_items where spine_items is an iterable of the form (name, linear). Will raise an error if one of the names is not present in the manifest.

property spine_items

An iterator yielding the path for every item in the books" spine. See also: *spine_iter* (página 357) and *spine_items* (página 357).

property spine_iter

An iterator that yields item, name is_linear for every item in the books" spine. item is the lxml element, name is the canonical file name and is_linear is True if the item is linear. See also: *spine_names* (página 357) and *spine_items* (página 357).

property spine_names

An iterator yielding name and is_linear for every item in the books" spine. See also: *spine_iter* (página 357) and *spine_items* (página 357).

Gerir ficheiros componentes num contentor

calibre.ebooks.oeb.polish.replace.**replace_links**(*container*, *link_map*, *frag_map=<function <lambda>>*, *replace_in_opf=False*)

Replace links to files in the container. Will iterate over all files in the container and change the specified links in them.

Parâmetros

- **link_map** – A mapping of old canonical name to new canonical name. For example: `{'images/old.png': 'images/new.png'}`
- **frag_map** – A callable that takes two arguments `(name, anchor)` and returns a new anchor. This is useful if you need to change the anchors in HTML files. By default, it does nothing.
- **replace_in_opf** – If False, links are not replaced in the OPF file.

calibre.ebooks.oeb.polish.replace.**rename_files**(*container*, *file_map*)

Rename files in the container, automatically updating all links to them.

Parâmetros

file_map – A mapping of old canonical name to new canonical name, for example: `{'text/chapter1.html': 'chapter1.html'}`.

calibre.ebooks.oeb.polish.replace.**get_recommended_folders**(*container*, *names*)

Return the folders that are recommended for the given filenames. The recommendation is based on where the majority of files of the same type are located in the container. If no files of a particular type are present, the recommended folder is assumed to be the folder containing the OPF file.

Impressão bonita e correção automática de erros de análise

calibre.ebooks.oeb.polish.pretty.**fix_html**(*container*, *raw*)

Corrija os erros de análise do HTML representados por uma cadeia de caracteres em bruto. A correção é efetuada usando o algoritmo de análise do HTML5.

calibre.ebooks.oeb.polish.pretty.**fix_all_html**(*container*)

Corrija os erros de análise em todos os ficheiros HTML do contentor. A correção é efetuada usando o algoritmo de análise do HTML5.

calibre.ebooks.oeb.polish.pretty.**pretty_html**(*container*, *name*, *raw*)

Pretty print the HTML represented as a string in raw

calibre.ebooks.oeb.polish.pretty.**pretty_css**(*container*, *name*, *raw*)

Pretty print the CSS represented as a string in raw

calibre.ebooks.oeb.polish.pretty.**pretty_xml**(*container*, *name*, *raw*)

Pretty print the XML represented as a string in raw. If `name` is the name of the OPF, extra OPF-specific prettying is performed.

calibre.ebooks.oeb.polish.pretty.**pretty_all**(*container*)

Imprime de forma visualmente apelativa todos os ficheiros HTML/CSS/XML existentes no contentor

Gerir sobrecapas de livros

calibre.ebooks.oeb.polish.jacket.**remove_jacket**(*container*)

Remove uma sobrecapa existente, se existir. Devolve False se não for encontrada uma sobrecapa.

calibre.ebooks.oeb.polish.jacket.**add_or_replace_jacket**(*container*)

Efetua uma de duas operações: cria uma nova sobrecapa a partir dos metadados do livro ou substitui a sobrecapa existente. Devolve Verdadeiro se a sobrecapa existente for substituída.

Separar e fundir ficheiros

calibre.ebooks.oeb.polish.split.**split**(*container*, *name*, *loc_or_xpath*, *before=True*, *totals=None*)

Split the file specified by name at the position specified by loc_or_xpath. Splitting automatically migrates all links and references to the affected files.

Parâmetros

- **loc_or_xpath** – Should be an XPath expression such as //h:div[@id=»split_here»]. Can also be a *loc* which is used internally to implement splitting in the preview panel.
- **before** – Se Verdadeiro a separação ocorre antes do elemento identificado; se não, ocorre depois dele.
- **totals** – Usado internamente

calibre.ebooks.oeb.polish.split.**multisplit**(*container*, *name*, *xpath*, *before=True*)

Split the specified file at multiple locations (all tags that match the specified XPath expression). See also: *split()* (página 358). Splitting automatically migrates all links and references to the affected files.

Parâmetros

before – Se Verdadeiro as separações ocorrem antes do elemento identificado; se não, ocorrem depois dele.

calibre.ebooks.oeb.polish.split.**merge**(*container*, *category*, *names*, *master*)

Funde os ficheiros especificados num único ficheiro, com a migração automática de todas as ligações e referências dos ficheiros afetados. Os ficheiros têm todos de ser HTML ou CSS.

Parâmetros

- **category** – Tem de ser `'text'` para ficheiros HTML ou `'styles'` para CSS
- **names** – A lista de ficheiros a serem fundidos
- **master** – Qual dos ficheiros a fundir será o ficheiro *mestre*, ou seja, qual é o ficheiro que será mantido depois da fusão.

Gerir capas

calibre.ebooks.oeb.polish.cover.**set_cover**(*container*, *cover_path*, *report=None*, *options=None*)

Torna a capa do livro na imagem apontada por cover_path.

Parâmetros

- **cover_path** – O caminho absoluto para um ficheiro de imagem ou o nome canónico de uma imagem no livro. Quando usa uma imagem no livro tem também de definir opções, ver abaixo.
- **report** – Um função opcional que pode ser chamada e que tem um único argumento. Será chamada com informação acerca das tarefas em processamento.
- **options** – Vazio, ou um dicionário que controla como a capa é criada. O dicionário pode conter as entradas: **keep_aspect**: Verdadeiro ou Falso (Preservar a proporção das imagens de capa no EPUB) **no_svg**: Verdadeiro ou Falso (Usar um embrulho SVG para a capa na página de título do EPUB) **existing**: Verdadeiro ou Falso (`cover_path` refere-se a uma imagem existente no livro)

calibre.ebooks.oeb.polish.cover.**mark_as_cover**(*container*, *name*)

Marca a imagem especificada como a imagem da capa.

calibre.ebooks.oeb.polish.cover.**mark_as_titlepage**(*container*, *name*, *move_to_start=True*)

Marca o ficheiro HTML especificado como a página de capa do EPUB.

Parâmetros

move_to_start – Se verdadeiro, o ficheiro HTML será movido para o início da lombada

Trabalhar com CSS

calibre.ebooks.oeb.polish.fonts.**change_font**(*container*, *old_name*, *new_name=None*)

Change a font family from old_name to new_name. Changes all occurrences of the font family in stylesheets, style tags and style attributes. If the old_name refers to an embedded font, it is removed. You can set new_name to None to remove the font family instead of changing it.

calibre.ebooks.oeb.polish.css.**remove_unused_css**(*container*, *report=None*, *remove_unused_classes=False*, *merge_rules=False*, *merge_rules_with_identical_properties=False*, *remove_unreferenced_sheets=False*)

Remove all unused CSS rules from the book. An unused CSS rule is one that does not match any actual content.

Parâmetros

- **report** – An optional callable that takes a single argument. It is called with information about the operations being performed.
- **remove_unused_classes** – If True, class attributes in the HTML that do not match any CSS rules are also removed.
- **merge_rules** – Se Verdadeiro, regras com selectores idênticos são fundidas.
- **merge_rules_with_identical_properties** – If True, rules with identical properties are merged.
- **remove_unreferenced_sheets** – If True, stylesheets that are not referenced by any content are removed

calibre.ebooks.oeb.polish.css.**filter_css**(*container*, *properties*, *names=()*)

Remove as propriedades CSS especificadas de todas as regras CSS no livro.

Parâmetros

- **properties** – Conjunto de propriedades a remover. Por exemplo: `{'font-family', 'color'}`.
- **names** – The files from which to remove the properties. Defaults to all HTML and CSS files in the book.

Trabalhar com o Índice

calibre.ebooks.oeb.polish.toc.**from_xpaths**(*container*, *xpaths*, *prefer_title=False*)

Cria um índice a partir de uma lista de expressões XPath. Cada expressão na lista corresponde a um nível do índice gerado. Por exemplo: `['//h:h1', '//h:h2', '//h:h3']` criará um índice de três níveis partindo das etiquetas `<h1>`, `<h2>` e `<h3>`.

calibre.ebooks.oeb.polish.toc.**from_links**(*container*)

Gerar um índice a partir de hiperligações no livro.

calibre.ebooks.oeb.polish.toc.**from_files**(*container*)

Gerar um índice a partir de ficheiros no livro.

calibre.ebooks.oeb.polish.toc.**create_inline_toc**(*container*, *title=None*)

Create an inline (HTML) Table of Contents from an existing NCX Table of Contents.

Parâmetros

title – O título para este índice.

Editar ferramenta do livro

class calibre.gui2.tweak_book.plugin.**Tool**

Bases: `object`

The base class for individual tools in an Edit Book plugin. Useful members include:

- `self.plugin`: A reference to the *calibre.customize.Plugin* (página 250) object to which this tool belongs.
- self. *boss* (página 361)
- self. *gui* (página 361)

Methods that must be overridden in sub classes:

- *create_action()* (página 361)
- *register_shortcut()* (página 361)

name = None

Set this to a unique name it will be used as a key

allowed_in_toolbar = True

If True the user can choose to place this tool in the plugins toolbar

allowed_in_menu = True

If True the user can choose to place this tool in the plugins menu

toolbar_button_popup_mode = 'delayed'

The popup mode for the menu (if any) of the toolbar button. Possible values are "delayed", "instant", "button"

property boss

The *calibre.gui2.tweak_book.boss.Boss* (página 362) object. Used to control the user interface.

property gui

The main window of the user interface

property current_container

Return the current *calibre.ebooks.oeb.polish.container.Container* (página 353) object that represents the book being edited.

register_shortcut(*qaction*, *unique_name*, *default_keys=()*, *short_text=None*, *description=None*, ***extra_data*)

Register a keyboard shortcut that will trigger the specified `qaction`. This keyboard shortcut will become automatically customizable by the user in the Keyboard shortcuts section of the editor preferences.

Parâmetros

- **qaction** – A QAction object, it will be triggered when the configured key combination is pressed by the user.
- **unique_name** – A unique name for this shortcut/action. It will be used internally, it must not be shared by any other actions in this plugin.
- **default_keys** – A list of the default keyboard shortcuts. If not specified no default shortcuts will be set. If the shortcuts specified here conflict with either builtin shortcuts or shortcuts from user configuration/other plugins, they will be ignored. In that case, users will have to configure the shortcuts manually via Preferences. For example: `default_keys=('Ctrl+J', 'F9')`.
- **short_text** – Uma descrição breve opcional desta ação. Se não especificada será usado o texto de QAction.
- **description** – Uma descrição mais detalhada desta ação, que será usada na entrada das preferências deste atalho.

create_action(*for_toolbar=True*)

Cria uma QAction que será adicionada à barra de ferramentas dos plugins ou ao menu de plugins, dependendo valor de `for_toolbar`. Por exemplo:

```
def create_action(self, for_toolbar=True):
    ac = QAction(get_icons('myicon.png'), 'Do something')
    if for_toolbar:
```

(continues on next page)

(continuação da página anterior)

```
        # 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
```

Veja também:

Método *register_shortcut()* (página 361).

Controlar a interface de utilizador do editor

A interface do usuário do editor de e-book é controlada por um único objeto *Boss* global. Isso tem muitos métodos úteis que podem ser usados no código do plugin para realizar tarefas comuns.

class calibre.gui2.tweak_book.boss.**Boss**(*parent*, *notify=None*)

add_savepoint(*msg*)

Create a restore checkpoint with the name specified as `msg`

apply_container_update_to_gui(*mark_as_modified=True*)

Update all the components of the user interface to reflect the latest data in the current book container.

Parâmetros

mark_as_modified – If True, the book will be marked as modified, so the user will be prompted to save it when quitting.

close_editor(*name*)

Close the editor that is editing the file specified by `name`

commit_all_editors_to_container()

Commit any changes that the user has made to files open in editors to the container. You should call this method before performing any actions on the current container

property currently_editing

Return the name of the file being edited currently or None if no file is being edited

edit_file(*name*, *syntax=None*, *use_template=None*)

Open the file specified by name in an editor

Parâmetros

- **syntax** – The media type of the file, for example, `'text/html'`. If not specified it is guessed from the file extension.
- **use_template** – A template to initialize the opened editor with

open_book(*path=None*, *edit_file=None*, *clear_notify_data=True*, *open_folder=False*, *search_text=None*)

Open the e-book at `path` for editing. Will show an error if the e-book is not in a supported format or the current book has unsaved changes.

Parâmetros

edit_file – The name of a file inside the newly opened book to start editing. Can also be a list of names.

rewind_savepoint()

Undo the previous creation of a restore checkpoint, useful if you create a checkpoint, then abort the operation with no changes

save_book()

Save the book. Saving is performed in the background

set_modified()

Mark the book as having been modified

show_current_diff(*allow_revert=True*, *to_container=None*)

Show the changes to the book from its last checkpointed state

Parâmetros

- **allow_revert** – If True the diff dialog will have a button to allow the user to revert all changes
- **to_container** – A container object to compare the current container to. If None, the previously checkpointed container is used

show_editor(*name*)

Show the editor that is editing the file specified by `name`

sync_preview_to_editor()

Sincronizar posição do painel de pré-visualização com a posição atual do cursor no editor atual

CAPÍTULO 15

Gestão de direitos digitais (DRM em inglês)

Gestão de Direitos Digitais (Digital Rights Management - DRM) é um termo genérico para tecnologias de controlo de acesso que podem ser usadas por fabricantes de hardware, editoras, detentores de copyright e indivíduos para tentar impor limitações ao uso de conteúdos e dispositivos digitais. É também, em alguns casos, descrito depreciativamente como Gestão de Restrições Digitais (Digital Restrictions Management). O termo é usado para descrever qualquer tecnologia que iniba usos (legítimos ou não) de conteúdos digitais que não são desejados ou previstos pelo fornecedor dos conteúdos. O termo geralmente não se refere a outras formas de proteção contra cópia que possam ser contornadas sem modificar o ficheiro ou o dispositivo, tais como números de série ou ficheiros-chave. Pode também referir-se a restrições associadas a instâncias específicas de produtos ou dispositivos digitais. As tecnologias DRM tentam controlar o uso de media digital ao evitar o acesso, cópia ou conversão para outros formatos pelos utilizadores finais. Veja Wikipedia[131].

15.1 Quais as consequências do DRM para mim?

Quando adquire um ebook protegido com DRM você não é o seu dono, apenas adquiriu a licença para o usar de um modo que lhe é ditado pelo vendedor. A DRM limita o que pode fazer com os ebooks que «comprou». Muitas vezes as pessoas que adquirem livros com DRM não estão alertadas para a extensão destas restrições. Estas restrições impedem--no de reformatar o ebook ao seu gosto, incluindo o fazer mudanças de estilo tais como ajustar o tamanho dos tipos de letra, embora haja software que lhe dê a possibilidade de efetuar estas mudanças em livros não protegidos por DRM. As pessoas são muitas vezes surpreendidas com o facto de que um ebook que compraram num formato particular não possa ser convertido para outro formato se o ebook estiver protegido por DRM. Portanto, se tiver um Kindle da Amazon e comprar um livro vendido pela Barnes and Noble, é conveniente saber que, se esse ebook está protegido por DRM, você não conseguirá lê-lo no seu Kindle. Tenha em atenção de que estou a referir-me a um livro que comprou, não que roubou ou pirateou, mas sim COMPROU.

[131] https://en.wikipedia.org/wiki/Digital_rights_management

15.2 O que faz o DRM pelos autores?

A editoras dos ebooks protegidos por DRM argumentam que o DRM se destina a salvaguardar os autores e para proteger a sua integridade artística e prevenir a pirataria. Mas o DRM NÃO previne a pirataria. Aqueles que querem piratear conteúdos, ou usam conteúdos pirateados, continuam a fazê-lo com sucesso. Os três maiores esquemas de DRM para ebooks da atualidade são os implementados pela Amazon, Adobe e Barnes and Noble, e todos esses três esquemas de DRM foram já crackados. Tudo o que o DRM faz é trazer inconveniências aos utilizadores legítimos. Pode argumentar-se que o DRM na verdade prejudica os autores, porque as pessoas que iriam comprar um livro escolhem encontrar uma versão pirateada pois não estão para aturar as limitações impostas pelo DRM. Aqueles que pirateiam continuam a fazê-lo, haja ou não DRM. Reitera-se que o ponto chave é que o DRM *não previne a pirataria*. Assim, o DRM não só não tem qualquer sentido e é prejudicial para os compradores de ebooks, como é também um desperdício de dinheiro.

15.3 DRM e liberdade

Embora o conteúdo digital possa ser usado para produzir informação ou trabalhos criativos facilmente disponíveis a todos e para dar poder à humanidade, isto não se coaduna com os interesses de alguns editores que querem direcionar as pessoas para longe desta possibilidade de liberdade apenas para manter a sua relevância num mundo onde não se conseguem manter a par com a rapidez com que o desenvolvimento se desenrola.

15.4 Porque é que o Calibre não suporta DRM?

O Calibre é um software de código aberto enquanto que o DRM, pela sua própria natureza, é fechado. Se o Calibre suportasse a abertura ou visualização de ficheiros protegidos por DRM poderia ser trivialmente modificado para ser usado como uma ferramenta para remover o DRM, o que é ilegal pelas leis atuais. O software de código aberto e o DRM representam um choque de princípios. Enquanto o DRM se foca em controlar o utilizador, o software de código aberto centra-se em dar o poder ao utilizador. Ambos não podem, simplesmente, coexistir.

15.5 Qual é a posição do Calibre sobre fornecedores de conteúdos?

Acreditamos firmemente que os autores e outros produtores de conteúdos devem ter a devida compensação pelos seus esforços, mas a DRM não é o caminho para o conseguir. Estamos a desenvolver esta base de dados de ebooks livres de DRM a partir de várias fontes para o ajudar a encontrar alternativas sem DRM e para ajudar autores independentes e editores de ebooks sem DRM a publicitar os seus conteúdos. Esperamos que ache isto útil e pedimos-lhe que não pirateie o conteúdo que aqui lhe disponibilizamos.

15.6 Como posso ajudar a luta contra o DRM?

Na qualidade de quem lê e compra ebooks você pode ajudar a lutar contra a DRM. Não compre ebooks com DRM. Existem alguns editores que publicam ebooks sem DRM. Faça um esforço para verificar eles lhe disponibilizam o ebook que está a procurar. Se gosta de livros de certos autores independentes que vendem ebooks sem DRM, e se tem posses para isso, envie-lhes donativos. Esse é dinheiro bem gasto, dado que os respetivos ebooks tendem a ser mais baratos (pode haver exceções) do que aqueles que compraria a editores de livros com DRM, e provavelmente são utilizáveis em todos os dispositivos que possuir futuramente poupando-lhe o custo de recomprar o ebook. Não desencoraje os editores e autores de ebooks sem DRM ao piratear o respetivo conteúdo. Os produtores de conteúdos merecem compensação pelo seu esforço. Não os puna por tentar tornar melhor a sua experiência de leitura ao disponibilizarem ebooks sem DRM: a longo prazo tal resultará em seu detrimento. Se comprou livros a vendedores que comercializam livros com e sem DRM,

sem saber se eles têm ou não DRM, faça ponto de honra em deixar um comentário no respetivo sítio Web informando compradores futuros do respetivo modo de DRM. Muitos vendedores não acham ser importante a indicação clara aos seus compradores sobre se um ebook tem ou não DRM. *Aqui <https://www.defectivebydesign.org/guide/ebooks>* encontrará um guia sobre como viver sem DRM.

CAPÍTULO 16

Glossário

RSS

RSS *(Really Simple Syndication) é um formato de sinais Web que pode ser usado para publicar conteúdos atualizados frequentemente, como artigos de notícias, blogues, etc. É um formato particularmente adequado para ser lido por computadores, e é por isso a forma favorita para a transformação de conteúdos web para livros digitais. Existem muitos outros formatos em atualização na Internet e o Calibre é capaz de trabalhar com a maior parte deles, em especial o formato *ATOM* que é normalmente usado em blogues.

receita

Uma receita é um conjunto de instruções que guiam o Calibre na conversão de uma fonte de notícias Web, tal como uma revista ou um blogue, num livro digital. A receita é essencialmente código *Python <https://www.python.org>*. Como tal, é capaz de converter fontes de notícias especialmente complexas em livros digitais. Num nível mais simplificado, é apenas um conjunto de variáveis, como o URL, que dão ao Calibre informação suficiente para aceder a Internet e transferir notícias.

HTML

HTML *(Hyper Text Mark-Up Language)*, é um subconjunto da Standard Generalized Mark-Up Language (SGML) para publicação eletrónica. É o padrão especificamente usado para a World Wide Web.

CSS

CSS *(Cascading Style Sheets)* é uma linguagem usada para descrever como um documento *HTML* deve ser apresentado (estilos visuais).

API

API *(Application Programming Interface)* é um interface de código-fonte fornecido por uma biblioteca para suportar pedidos de serviços a serem efetuados por programas de computador.

LRF

LRF O formato de livro que é lido pelos leitores de livros digitais da SONY

URL

URL *(Uniform Resource Locator)* por exemplo: «http://example.com»

regexp

Expressões regulares fornece uma forma concisa e flexível de identificar cadeias de caracteres de texto com interesse, tais como caracteres especiais, palavras ou padrões de caracteres. Ver *tutorial* (página 208) para introdução a expressões regulares.

Índice de Módulos do Python

C

calibre.customize, 250
calibre.customize.conversion, 259
calibre.db.cache, 345
calibre.devices.interface, 262
calibre.ebooks.metadata.book.base, 204
calibre.ebooks.metadata.sources.base, 256
calibre.ebooks.oeb.polish.container, 352
calibre.ebooks.oeb.polish.cover, 359
calibre.ebooks.oeb.polish.css, 359
calibre.ebooks.oeb.polish.jacket, 358
calibre.ebooks.oeb.polish.pretty, 358
calibre.ebooks.oeb.polish.replace, 357
calibre.ebooks.oeb.polish.split, 358
calibre.ebooks.oeb.polish.toc, 360
calibre.gui2.tweak_book.boss, 362
calibre.utils.formatter_functions, 185
calibre.web.feeds.news, 38

Índice

Símbolos

-1
    calibredb-add command line option, 308
-H
    ebook-polish command line option, 330
-I
    calibredb-add command line option, 308
    fetch-ebook-metadata command line option, 333
-S
    calibredb-add command line option, 308
-T
    calibredb-add command line option, 308
-U
    ebook-polish command line option, 331
--access-log
    calibre-server command line option, 299
--add
    calibredb-add command line option, 308
--add-plugin
    calibre-customize command line option, 296
--add-simple-plugin
    calibre-debug command line option, 297
--add-soft-hyphens
    ebook-polish command line option, 330
--ajax-timeout
    calibre-server command line option, 299
--all
    calibredb-backup_metadata command line option, 316
    calibredb-export command line option, 310
--allowed-plugin
    fetch-ebook-metadata command line option, 332
--append
    calibredb-set_custom command line option, 315
--as-opf
    calibredb-show_metadata command line option, 309
--ascending
    calibredb-list command line option, 306
--asciiize
    ebook-convert command line option, 321
--attachment
    calibre-smtp command line option, 303
--auth-mode
    calibre-server command line option, 299
--author-sort
    ebook-convert command line option, 326
    ebook-meta command line option, 328
--authors
    calibredb-add command line option, 307
    ebook-convert command line option, 326
    ebook-meta command line option, 328
    fetch-ebook-metadata command line option, 332
--auto-reload
    calibre-server command line option, 299
--automerge
    calibredb-add command line option, 307

--ban-after
calibre-server command line option, 299
--ban-for
calibre-server command line option, 299
--base-dir
web2disk command line option, 335
--base-font-size
ebook-convert command line option, 321
--book-list-mode
calibre-server command line option, 300
--book-producer
ebook-convert command line option, 326
ebook-meta command line option, 328
--build-plugin
calibre-customize command line option, 296
--cafile
calibre-smtp command line option, 304
--catalog-title
calibredb-catalog command line option, 312
--categories
calibredb-list_categories command line option, 316
--category
ebook-meta command line option, 329
--change-justification
ebook-convert command line option, 321
--chapter
ebook-convert command line option, 325
--chapter-mark
ebook-convert command line option, 325
--command
calibre-debug command line option, 297
--comments
ebook-convert command line option, 326
ebook-meta command line option, 329
--compress-images
ebook-polish command line option, 330
--compress-min-size
calibre-server command line option, 300
--continue
ebook-viewer command line option, 332
--cover
calibredb-add command line option, 307
ebook-convert command line option, 327
ebook-meta command line option, 329
ebook-polish command line option, 330
fetch-ebook-metadata command line option, 332
--cross-reference-authors
calibredb-catalog command line option, 312
--csv
calibredb-check_library command line option, 315
calibredb-list_categories command line option, 316
--custom-list-template
calibre-server command line option, 300
--customize-plugin
calibre-customize command line option, 296
--daemonize
calibre-server command line option, 300
--date
ebook-meta command line option, 329
--debug-device-driver
calibre-debug command line option, 297
--debug-pipeline
calibredb-catalog command line option, 312
ebook-convert command line option, 327
--default-programs
calibre-debug command line option, 297
--delay
web2disk command line option, 335
--detach
calibre command line option, 296
ebook-edit command line option, 328
ebook-viewer command line option, 332
--details
calibredb-custom_columns command line option, 314
--dialect
calibredb-list_categories command line option, 316
--diff
calibre-debug command line option, 297

--disable-allow-socket-preallocation
calibre-server command line option, 300
--disable-auth
calibre-server command line option, 300
--disable-dehyphenate
ebook-convert command line option, 324
--disable-delete-blank-paragraphs
ebook-convert command line option, 324
--disable-fallback-to-detected--interface
calibre-server command line option, 300
--disable-fix-indents
ebook-convert command line option, 324
--disable-font-rescaling
ebook-convert command line option, 321
--disable-format-scene-breaks
ebook-convert command line option, 324
--disable-hyphenation
lrfviewer command line option, 334
--disable-italicize-common-cases
ebook-convert command line option, 324
--disable-local-write
calibre-server command line option, 300
--disable-log-not-found
calibre-server command line option, 300
--disable-markup-chapter-headings
ebook-convert command line option, 324
--disable-plugin
calibre-customize command line option, 296
--disable-remove-fake-margins
ebook-convert command line option, 325
--disable-renumber-headings
ebook-convert command line option, 324
--disable-unwrap-lines
ebook-convert command line option, 324
--disable-use-bonjour
calibre-server command line option, 300
--disable-use-sendfile
calibre-server command line option, 301
--display
calibredb-add_custom_column command line option, 314
--displayed-fields
calibre-server command line option, 300
--do-not-match-on-related-words
calibredb-fts_search command line option, 318
--dont-asciiize
calibredb-export command line option, 310
--dont-download-stylesheets
web2disk command line option, 335
--dont-output-resources
lrf2lrs command line option, 333
--dont-replace
calibredb-add_format command line option, 309
--dont-save-cover
calibredb-export command line option, 310
--dont-split-on-page-breaks
ebook-convert command line option, 320
--dont-update-metadata
calibredb-export command line option, 310
--dont-verify-server-certificate
calibre-smtp command line option, 304
--dont-write-opf
calibredb-export command line option, 310
--duplicate-links-in-toc
ebook-convert command line option, 326
--duplicates
calibredb-add command line option, 307
--edit-book
calibre-debug command line option, 297
--embed-all-fonts
ebook-convert command line option, 321
--embed-font-family
ebook-convert command line option, 321
--embed-fonts
ebook-polish command line option, 330
--empty

calibredb-add command line option, 307
--enable-allow-socket-preallocation
calibre-server command line option, 300
--enable-auth
calibre-server command line option, 300
--enable-fallback-to-detected-interface
calibre-server command line option, 300
--enable-heuristics
ebook-convert command line option, 324
--enable-local-write
calibre-server command line option, 300
--enable-log-not-found
calibre-server command line option, 300
--enable-plugin
calibre-customize command line option, 296
--enable-use-bonjour
calibre-server command line option, 300
--enable-use-sendfile
calibre-server command line option, 301
--encoding
web2disk command line option, 335
--encryption-method
calibre-smtp command line option, 304
--epub-flatten
ebook-convert command line option, 320
--epub-inline-toc
ebook-convert command line option, 320
--epub-toc-at-end
ebook-convert command line option, 320
--epub-version
ebook-convert command line option, 320
--exclude-genre
calibredb-catalog command line option, 312
--exclusion-rules
calibredb-catalog command line option, 312
--exec-file
calibre-debug command line option, 298
--expand-css
ebook-convert command line option, 321
--explode-book
calibre-debug command line option, 298
--export-all-calibre-data
calibre-debug command line option, 298
--extra-css
ebook-convert command line option, 321
--extract-to
ebook-convert command line option, 320
--field
calibredb-set_metadata command line option, 310
--fields
calibredb-list command line option, 306
--filter-css
ebook-convert command line option, 322
--filter-regexp
web2disk command line option, 335
--fix-multiprocessing
calibre-debug command line option, 298
--flow-size
ebook-convert command line option, 320
--font-size-mapping
ebook-convert command line option, 322
--for-machine
calibredb-list command line option, 306
--force
calibredb-remove_custom_column command line option, 314
--force-reload
ebook-viewer command line option, 332
--fork
calibre-smtp command line option, 303
--formats
calibredb-export command line option, 310
--from-opf
ebook-convert command line option, 327
ebook-meta command line option, 329
--full-screen

ebook-viewer command line option, 332
--fullscreen
ebook-viewer command line option, 332
--generate-authors
calibredb-catalog command line option, 312
--generate-descriptions
calibredb-catalog command line option, 312
--generate-genres
calibredb-catalog command line option, 312
--generate-recently-added
calibredb-catalog command line option, 312
--generate-series
calibredb-catalog command line option, 312
--generate-titles
calibredb-catalog command line option, 312
--genre-source-field
calibredb-catalog command line option, 312
--get-cover
ebook-meta command line option, 329
--gui
calibre-debug command line option, 298
--gui-debug
calibre-debug command line option, 298
--header-note-source-field
calibredb-catalog command line option, 312
--help
calibre command line option, 296
calibre-customize command line option, 297
calibre-debug command line option, 298
calibre-server command line option, 301
calibre-smtp command line option, 303
command line option, 306
ebook-convert command line option, 319
ebook-edit command line option, 328
ebook-meta command line option, 329
ebook-polish command line option, 330
ebook-viewer command line option, 332
fetch-ebook-metadata command line option, 333
lrf2lrs command line option, 333
lrfviewer command line option, 334
lrs2lrf command line option, 334
web2disk command line option, 335
--html-unwrap-factor
ebook-convert command line option, 324
--identifier
calibredb-add command line option, 308
ebook-meta command line option, 329
fetch-ebook-metadata command line option, 333
--ids
calibredb-catalog command line option, 311
--ignore
calibredb-add command line option, 308
--ignore_extensions
calibredb-check_library command line option, 315
--ignore_names
calibredb-check_library command line option, 315
--ignore-plugins
calibre command line option, 296
--ignored-fields
calibre-server command line option, 301
--implode-book
calibre-debug command line option, 298
--import-calibre-data
calibre-debug command line option, 298
--include-snippets
calibredb-fts_search command line option, 318
--index
ebook-meta command line option, 329
--indexing-speed
calibredb-fts_index command line option, 318
--indexing-threshold
calibredb-fts_search command line option, 318
--input-encoding
ebook-convert command line option, 320
--input-profile
ebook-convert command line option, 319
--insert-blank-line
ebook-convert command line option,

322
--insert-blank-line-size
ebook-convert command line option, 322
--insert-metadata
ebook-convert command line option, 325
--inspect-mobi
calibre-debug command line option, 298
--is-multiple
calibredb-add_custom_column command line option, 314
--isbn
calibredb-add command line option, 308
ebook-convert command line option, 327
ebook-meta command line option, 329
fetch-ebook-metadata command line option, 333
--item_count
calibredb-list_categories command line option, 316
--jacket
ebook-polish command line option, 330
--keep-ligatures
ebook-convert command line option, 322
--language
ebook-convert command line option, 327
ebook-meta command line option, 329
--languages
calibredb-add command line option, 308
--level1-toc
ebook-convert command line option, 326
--level2-toc
ebook-convert command line option, 326
--level3-toc
ebook-convert command line option, 326
--library-path
command line option, 306
--limit
calibredb-list command line option, 306
calibredb-search command line option, 317
--line-height
ebook-convert command line option, 322
--line-width
calibredb-list command line option, 306
--linearize-tables
ebook-convert command line option, 322
--list-fields
calibredb-set_metadata command line option, 310
--list-plugins
calibre-customize command line option, 297
--list-recipes
ebook-convert command line option, 319
--listen-on
calibre-server command line option, 301
--localhost
calibre-smtp command line option, 303
--log
calibre-server command line option, 301
--lrf-bookid
ebook-meta command line option, 329
--lrs
lrs2lrf command line option, 334
--manage-users
calibre-server command line option, 301
--margin-bottom
ebook-convert command line option, 322
--margin-left
ebook-convert command line option, 322
--margin-right
ebook-convert command line option, 322
--margin-top
ebook-convert command line option, 322
--match-end-marker
calibredb-fts_search command line option, 318
--match-regexp
web2disk command line option, 335
--match-start-marker
calibredb-fts_search command line option, 318
--max-files
web2disk command line option, 335
--max-header-line-size

calibre-server command line option, 301
--max-job-time
calibre-server command line option, 301
--max-jobs
calibre-server command line option, 301
--max-log-size
calibre-server command line option, 301
--max-opds-items
calibre-server command line option, 301
--max-opds-ungrouped-items
calibre-server command line option, 301
--max-recursions
web2disk command line option, 335
--max-request-body-size
calibre-server command line option, 301
--max-toc-links
ebook-convert command line option, 326
--merge-comments-rule
calibredb-catalog command line option, 312
--minimum-line-height
ebook-convert command line option, 323
--no-chapters-in-toc
ebook-convert command line option, 326
--no-default-epub-cover
ebook-convert command line option, 320
--no-svg-cover
ebook-convert command line option, 320
--no-update-check
calibre command line option, 296
--num-per-page
calibre-server command line option, 302
--one-book-per-directory
calibredb-add command line option, 308
--only-formats
calibredb-embed_metadata command line option, 317
--open-at
ebook-viewer command line option, 332
--opf
ebook-polish command line option, 330
fetch-ebook-metadata command line option, 333
--outbox
calibre-smtp command line option, 303
--output
lrf2lrs command line option, 333
lrs2lrf command line option, 334
--output-format
calibredb-fts_search command line option, 318
--output-profile
calibredb-catalog command line option, 313
ebook-convert command line option, 319
--page-breaks-before
ebook-convert command line option, 325
--password
calibre-smtp command line option, 304
command line option, 306
--paths
calibre-debug command line option, 298
--permanent
calibredb-remove command line option, 309
--pidfile
calibre-server command line option, 302
--port
calibre-server command line option, 302
calibre-smtp command line option, 304
--prefer-metadata-cover
ebook-convert command line option, 325
--prefix
calibredb-list command line option, 307
--prefix-rules
calibredb-catalog command line option, 313
--preserve-cover-aspect-ratio
ebook-convert command line option, 320
--preset
calibredb-catalog command line option, 313
--pretty-print
ebook-convert command line option, 321
--profile

lrfviewer command line option, 334
--progress
calibredb-export command line option, 311
--pubdate
ebook-convert command line option, 327
--publisher
ebook-convert command line option, 327
ebook-meta command line option, 329
--raise-window
ebook-viewer command line option, 332
--rating
ebook-convert command line option, 327
ebook-meta command line option, 329
--read-metadata-from-opf
ebook-convert command line option, 327
--really-do-it
calibredb-restore_database command line option, 315
--recurse
calibredb-add command line option, 308
--relay
calibre-smtp command line option, 304
--remove-first-image
ebook-convert command line option, 325
--remove-jacket
ebook-polish command line option, 330
--remove-paragraph-spacing
ebook-convert command line option, 323
--remove-paragraph-spacing-indent-size
ebook-convert command line option, 323
--remove-plugin
calibre-customize command line option, 297
--remove-soft-hyphens
ebook-polish command line option, 331
--remove-unused-css
ebook-polish command line option, 331
--replace-scene-breaks
ebook-convert command line option, 324
--replace-whitespace
calibredb-export command line option, 311
--report
calibredb-check_library command line option, 315
--restrict-to
calibredb-fts_search command line option, 318
--run-plugin
calibre-debug command line option, 298
--run-test
calibre-debug command line option, 298
--run-without-debug
calibre-debug command line option, 298
--search
calibredb-catalog command line option, 311
calibredb-list command line option, 307
--search-replace
ebook-convert command line option, 324
--search-the-net-urls
calibre-server command line option, 302
--select-text
ebook-edit command line option, 328
--separator
calibredb-list command line option, 307
--series
calibredb-add command line option, 308
ebook-convert command line option, 327
ebook-meta command line option, 329
--series-index
calibredb-add command line option, 308
ebook-convert command line option, 327
--shutdown-running-calibre
calibre command line option, 296
calibre-debug command line option, 299
--shutdown-timeout
calibre-server command line option, 302
--single-dir
calibredb-export command line option, 311
--smarten-punctuation
ebook-convert command line option, 323

ebook-polish command line option, 331
--sort-by
calibredb-list command line option, 307
--sr1-replace
ebook-convert command line option, 324
--sr1-search
ebook-convert command line option, 324
--sr2-replace
ebook-convert command line option, 325
--sr2-search
ebook-convert command line option, 325
--sr3-replace
ebook-convert command line option, 325
--sr3-search
ebook-convert command line option, 325
--ssl-certfile
calibre-server command line option, 302
--ssl-keyfile
calibre-server command line option, 302
--start-in-tray
calibre command line option, 296
--start-reading-at
ebook-convert command line option, 325
--subject
calibre-smtp command line option, 303
--subset-embedded-fonts
ebook-convert command line option, 323
--subset-font
calibre-debug command line option, 299
--subset-fonts
ebook-polish command line option, 331
--tags
calibredb-add command line option, 308
ebook-convert command line option, 327
ebook-meta command line option, 329
--template
calibredb-export command line option, 311
calibredb-list command line option, 307
--template_file
calibredb-list command line option, 307
--template_heading
calibredb-list command line option, 307
--test-build
calibre-debug command line option, 299
--thumb-width
calibredb-catalog command line option, 313
--timefmt
calibredb-export command line option, 311
--timeout
calibre-server command line option, 302
calibre-smtp command line option, 303
command line option, 306
fetch-ebook-metadata command line option, 333
web2disk command line option, 335
--timestamp
ebook-convert command line option, 327
--title
calibredb-add command line option, 308
ebook-convert command line option, 327
ebook-meta command line option, 329
fetch-ebook-metadata command line option, 333
--title-sort
ebook-convert command line option, 327
ebook-meta command line option, 329
--to-dir
calibredb-export command line option, 311
--to-lowercase
calibredb-export command line option, 311
--to-opf
ebook-meta command line option, 330
--toc-filter
ebook-convert command line option, 326
--toc-threshold
ebook-convert command line option, 326
--toc-title
ebook-convert command line option,

321
--transform-css-rules
ebook-convert command line option, 323
--transform-html-rules
ebook-convert command line option, 323
--trusted-ips
calibre-server command line option, 302
--unsmarten-punctuation
ebook-convert command line option, 323
--upgrade-book
ebook-polish command line option, 331
--url-prefix
calibre-server command line option, 302
--use-auto-toc
ebook-convert command line option, 326
--use-existing-cover
calibredb-catalog command line option, 313
--userdb
calibre-server command line option, 302
--username
calibre-smtp command line option, 304
command line option, 306
--vacuum-fts-db
calibredb-check_library command line option, 315
--verbose
calibre command line option, 296
calibre-smtp command line option, 303
calibredb-catalog command line option, 311
ebook-convert command line option, 327
ebook-polish command line option, 331
fetch-ebook-metadata command line option, 333
lrf2lrs command line option, 333
lrfviewer command line option, 334
lrs2lrf command line option, 334
web2disk command line option, 335
--version
calibre command line option, 296
calibre-customize command line option, 297
calibre-debug command line option, 299
calibre-server command line option, 302
calibre-smtp command line option, 303
command line option, 306
ebook-convert command line option, 320
ebook-edit command line option, 328
ebook-meta command line option, 330
ebook-polish command line option, 331
ebook-viewer command line option, 332
fetch-ebook-metadata command line option, 333
lrf2lrs command line option, 333
lrfviewer command line option, 334
lrs2lrf command line option, 335
web2disk command line option, 336
--viewer
calibre-debug command line option, 299
--visual-debug
lrfviewer command line option, 334
--wait-for-completion
calibredb-fts_index command line option, 318
--white-background
lrfviewer command line option, 334
--width
calibredb-list_categories command line option, 316
--with-library
calibre command line option, 296
command line option, 306
--worker-count
calibre-server command line option, 302
-a
calibre-customize command line option, 296
calibre-smtp command line option, 303
calibredb-add command line option, 307
calibredb-set_custom command line option, 315
ebook-meta command line option, 328
fetch-ebook-metadata command line option, 332
-b
calibre-customize command line option, 296
-c
calibre-debug command line option, 297
calibredb-add command line option, 307

calibredb-check_library command line option, 315
calibredb-list_categories command line option, 316
ebook-meta command line option, 329
ebook-polish command line option, 330
fetch-ebook-metadata command line option, 332
-d
calibre-debug command line option, 297
calibredb-add command line option, 307
calibredb-custom_columns command line option, 314
ebook-convert command line option, 327
ebook-meta command line option, 329
fetch-ebook-metadata command line option, 333
web2disk command line option, 335
-e
calibre-debug command line option, 298
calibre-smtp command line option, 304
calibredb-add command line option, 307
calibredb-check_library command line option, 315
ebook-polish command line option, 330
-f
calibre-debug command line option, 299
calibre-smtp command line option, 303
calibredb-embed_metadata command line option, 317
calibredb-list command line option, 306
calibredb-remove_custom_column command line option, 314
calibredb-set_metadata command line option, 310
ebook-polish command line option, 331
ebook-viewer command line option, 332
-g
calibre-debug command line option, 298
-h
calibre command line option, 296
calibre-customize command line option, 297
calibre-debug command line option, 298
calibre-server command line option, 301
calibre-smtp command line option, 303
command line option, 306
ebook-convert command line option, 319
ebook-edit command line option, 328
ebook-meta command line option, 329
ebook-polish command line option, 330
ebook-viewer command line option, 332
fetch-ebook-metadata command line option, 333
lrf2lrs command line option, 333
lrfviewer command line option, 334
lrs2lrf command line option, 334
web2disk command line option, 335
-i
calibre-debug command line option, 298
calibredb-add command line option, 308
calibredb-catalog command line option, 311
calibredb-list_categories command line option, 316
ebook-meta command line option, 329
ebook-polish command line option, 330
fetch-ebook-metadata command line option, 333
-j
ebook-polish command line option, 330
-k
ebook-meta command line option, 328
-l
calibre-customize command line option, 297
calibre-smtp command line option, 303
calibredb-add command line option, 308
calibredb-search command line option, 317
calibredb-set_metadata command line option, 310
ebook-meta command line option, 329
-m
calibre-debug command line option, 298
calibredb-add command line option, 307
ebook-convert command line option, 327
-n
calibredb-check_library command line option, 315
web2disk command line option, 335

-o
  calibre-smtp command line option, 303
  ebook-polish command line option, 330
  fetch-ebook-metadata command line option, 333
  lrf2lrs command line option, 333
  lrs2lrf command line option, 334
-p
  calibre-smtp command line option, 304
  ebook-meta command line option, 329
  ebook-polish command line option, 331
  fetch-ebook-metadata command line option, 332
-r
  calibre-customize command line option, 297
  calibre-debug command line option, 298
  calibre-smtp command line option, 304
  calibredb-add command line option, 308
  calibredb-check_library command line option, 315
  calibredb-list_categories command line option, 316
  calibredb-restore_database command line option, 315
  ebook-meta command line option, 329
  web2disk command line option, 335
-s
  calibre command line option, 296
  calibre-debug command line option, 299
  calibre-smtp command line option, 303
  calibredb-add command line option, 308
  calibredb-catalog command line option, 311
  calibredb-list command line option, 307
  ebook-meta command line option, 329
-t
  calibre-debug command line option, 298
  calibre-smtp command line option, 303
  calibredb-add command line option, 308
  calibredb-list command line option, 307
  ebook-meta command line option, 329
  fetch-ebook-metadata command line option, 333
  web2disk command line option, 335
-u
  calibre-smtp command line option, 304
  ebook-polish command line option, 331
-v
  calibre command line option, 296
  calibre-smtp command line option, 303
  calibredb-catalog command line option, 311
  ebook-convert command line option, 327
  fetch-ebook-metadata command line option, 333
-w
  calibre-debug command line option, 299
  calibredb-list command line option, 306
  calibredb-list_categories command line option, 316
-x
  calibre-debug command line option, 298

A

abort_article() (*método calibre.web.feeds.news.BasicNewsRecipe*), 38
abort_recipe_processing() (*método calibre.web.feeds.news.BasicNewsRecipe*), 38
abspath_to_name() (*método calibre.ebooks.oeb.polish.container.Container*), 353
accept_drag_move_event() (*método calibre.gui2.actions.InterfaceAction*), 277
accept_enter_event() (*método calibre.gui2.actions.InterfaceAction*), 277
accepts_drops (*atributo calibre.gui2.actions.InterfaceAction*), 277
action_add_menu (*atributo calibre.gui2.actions.InterfaceAction*), 276
action_menu_clone_qaction (*atributo calibre.gui2.actions.InterfaceAction*), 276
action_spec (*atributo calibre.gui2.actions.InterfaceAction*), 276
action_type (*atributo calibre.gui2.actions.InterfaceAction*), 277
add_annotation_to_library() (*método calibre.devices.usbms.device.Device*), 273
add_book() (*método calibre.devices.interface.BookList*), 270
add_books() (*método calibre.db.cache.Cache*), 346
add_books_to_metadata() (*método calibre.devices.usbms.driver.USBMS*), 275
add_books_to_metadata() (*método de classe calibre.devices.interface.DevicePlugin*), 267

add_custom_book_data() (*método calibre.db.cache.Cache*), 346
add_file() (*método calibre.ebooks.oeb.polish.container.Container*), 353
add_format() (*método calibre.db.cache.Cache*), 346
add_listener() (*método calibre.db.cache.Cache*), 346
add_name_to_manifest() (*método calibre.ebooks.oeb.polish.container.Container*), 353
add_or_replace_jacket() (*no módulo calibre.ebooks.oeb.polish.jacket*), 358
add_properties() (*método calibre.ebooks.oeb.polish.container.Container*), 353
add_savepoint() (*método calibre.gui2.tweak_book.boss.Boss*), 362
add_toc_thumbnail() (*método calibre.web.feeds.news.BasicNewsRecipe*), 38
adeify_images() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 38
all_book_ids() (*método calibre.db.cache.Cache*), 346
all_field_for() (*método calibre.db.cache.Cache*), 347
all_field_ids() (*método calibre.db.cache.Cache*), 347
all_field_keys() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
all_field_names() (*método calibre.db.cache.Cache*), 347
all_non_none_fields() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
allowed_in_menu (*atributo calibre.gui2.tweak_book.plugin.Tool*), 361
allowed_in_toolbar (*atributo calibre.gui2.tweak_book.plugin.Tool*), 361
API, **369**
apply_container_update_to_gui() (*método calibre.gui2.tweak_book.boss.Boss*), 362
apply_unique_properties() (*método calibre.ebooks.oeb.polish.container.Container*), 353
articles_are_obfuscated (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 42
ASK_TO_ALLOW_CONNECT (*atributo calibre.devices.interface.DevicePlugin*), 263
author (*atributo calibre.customize.InterfaceActionBase*), 279
author (*atributo calibre.customize.MetadataReaderPlugin*), 254
author (*atributo calibre.customize.MetadataWriterPlugin*), 255
author (*atributo calibre.customize.Plugin*), 250
author (*atributo calibre.customize.PreferencesPlugin*), 279
author (*atributo calibre.devices.usbms.driver.USBMS*), 274
author (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
author_data() (*método calibre.db.cache.Cache*), 347
author_sort_from_authors() (*método calibre.db.cache.Cache*), 347
auto_cleanup (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 42
auto_cleanup_keep (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 42
auto_repeat (*atributo calibre.gui2.actions.InterfaceAction*), 276
auto_trim_covers (*atributo calibre.ebooks.metadata.sources.base.Source*), 256

B

BACKLOADING_ERROR_MESSAGE (*atributo calibre.devices.usbms.device.Device*), 271
BasicNewsRecipe (*classe em calibre.web.feeds.news*), 38
BCD (*atributo calibre.devices.interface.DevicePlugin*), 262
BCD (*atributo calibre.devices.usbms.device.Device*), 271
book_class (*atributo calibre.devices.usbms.driver.USBMS*), 274
book_created (*atributo calibre.db.cache.Cache.EventType*), 345
book_edited (*atributo calibre.db.cache.Cache.EventType*), 345
book_type (*atributo calibre.ebooks.oeb.polish.container.Container*), 353
BookList (*classe em calibre.devices.interface*), 270
booklist_class (*atributo calibre.devices.usbms.driver.USBMS*), 274
books() (*método calibre.devices.interface.DevicePlugin*), 266
books() (*método calibre.devices.usbms.driver.USBMS*), 274
books_for_field() (*método calibre.db.cache.Cache*), 347
books_in_virtual_library() (*método calibre.db.cache.Cache*), 347
books_removed (*atributo calibre.db.cache.Cache.EventType*), 346
boss (*calibre.gui2.tweak_book.plugin.Tool property*), 361
Boss (*classe em calibre.gui2.tweak_book.boss*), 362

BuiltinAdd (*classe em calibre.utils.formatter_functions*), 185
BuiltinAnd (*classe em calibre.utils.formatter_functions*), 186
BuiltinAnnotationCount (*classe em calibre.utils.formatter_functions*), 189
BuiltinApproximateFormats (*classe em calibre.utils.formatter_functions*), 189
BuiltinArguments (*classe em calibre.utils.formatter_functions*), 199
BuiltinAssign (*classe em calibre.utils.formatter_functions*), 199
BuiltinAuthorLinks (*classe em calibre.utils.formatter_functions*), 189
BuiltinAuthorSorts (*classe em calibre.utils.formatter_functions*), 189
BuiltinBookCount (*classe em calibre.utils.formatter_functions*), 203
BuiltinBooksize (*classe em calibre.utils.formatter_functions*), 189
BuiltinBookValues (*classe em calibre.utils.formatter_functions*), 204
BuiltinCapitalize (*classe em calibre.utils.formatter_functions*), 201
BuiltinCeiling (*classe em calibre.utils.formatter_functions*), 185
BuiltinCharacter (*classe em calibre.utils.formatter_functions*), 201
BuiltinCheckYesNo (*classe em calibre.utils.formatter_functions*), 193
BuiltinCmp (*classe em calibre.utils.formatter_functions*), 200
BuiltinConnectedDeviceName (*classe em calibre.utils.formatter_functions*), 190
BuiltinConnectedDeviceUUID (*classe em calibre.utils.formatter_functions*), 190
BuiltinContains (*classe em calibre.utils.formatter_functions*), 193
BuiltinCount (*classe em calibre.utils.formatter_functions*), 195
BuiltinCurrentLibraryName (*classe em calibre.utils.formatter_functions*), 190
BuiltinCurrentLibraryPath (*classe em calibre.utils.formatter_functions*), 190
BuiltinCurrentVirtualLibraryName (*classe em calibre.utils.formatter_functions*), 190
BuiltinDateArithmetic (*classe em calibre.utils.formatter_functions*), 187
BuiltinDaysBetween (*classe em calibre.utils.formatter_functions*), 187
BuiltinDivide (*classe em calibre.utils.formatter_functions*), 185
BuiltinEval (*classe em calibre.utils.formatter_functions*), 199
BuiltinField (*classe em calibre.utils.formatter_functions*), 190
BuiltinFieldExists (*classe em calibre.utils.formatter_functions*), 193
BuiltinFinishFormatting (*classe em calibre.utils.formatter_functions*), 187
BuiltinFirstMatchingCmp (*classe em calibre.utils.formatter_functions*), 200
BuiltinFirstNonEmpty (*classe em calibre.utils.formatter_functions*), 194
BuiltinFloor (*classe em calibre.utils.formatter_functions*), 185
BuiltinFormatDate (*classe em calibre.utils.formatter_functions*), 188
BuiltinFormatNumber (*classe em calibre.utils.formatter_functions*), 188
BuiltinFormatsModtimes (*classe em calibre.utils.formatter_functions*), 191
BuiltinFormatsPaths (*classe em calibre.utils.formatter_functions*), 191
BuiltinFormatsSizes (*classe em calibre.utils.formatter_functions*), 191
BuiltinFractionalPart (*classe em calibre.utils.formatter_functions*), 185
BuiltinGlobals (*classe em calibre.utils.formatter_functions*), 199
BuiltinHasCover (*classe em calibre.utils.formatter_functions*), 191
BuiltinHumanReadable (*classe em calibre.utils.formatter_functions*), 188
BuiltinIdentifierInList (*classe em calibre.utils.formatter_functions*), 194
BuiltinIfempty (*classe em calibre.utils.formatter_functions*), 193
BuiltinInList (*classe em calibre.utils.formatter_functions*), 195
BuiltinIsMarked (*classe em calibre.utils.formatter_functions*), 191
BuiltinLanguageCodes (*classe em calibre.utils.formatter_functions*), 191
BuiltinLanguageStrings (*classe em calibre.utils.formatter_functions*), 192
BuiltinListCountMatching (*classe em calibre.utils.formatter_functions*), 196
BuiltinListDifference (*classe em calibre.utils.formatter_functions*), 196
BuiltinListEquals (*classe em calibre.utils.formatter_functions*), 196
BuiltinListIntersection (*classe em calibre.utils.formatter_functions*), 196
BuiltinListitem (*classe em calibre.utils.formatter_functions*), 195
BuiltinListJoin (*classe em calibre.utils.formatter_functions*), 196

BuiltinListRe (*classe em calibre.utils.formatter_functions*), 197
BuiltinListReGroup (*classe em calibre.utils.formatter_functions*), 197
BuiltinListRemoveDuplicates (*classe em calibre.utils.formatter_functions*), 197
BuiltinListSort (*classe em calibre.utils.formatter_functions*), 197
BuiltinListSplit (*classe em calibre.utils.formatter_functions*), 197
BuiltinListUnion (*classe em calibre.utils.formatter_functions*), 198
BuiltinLookup (*classe em calibre.utils.formatter_functions*), 194
BuiltinLowercase (*classe em calibre.utils.formatter_functions*), 201
BuiltinMod (*classe em calibre.utils.formatter_functions*), 186
BuiltinMultiply (*classe em calibre.utils.formatter_functions*), 186
BuiltinNot (*classe em calibre.utils.formatter_functions*), 186
BuiltinOndevice (*classe em calibre.utils.formatter_functions*), 192
BuiltinOr (*classe em calibre.utils.formatter_functions*), 187
BuiltinPrint (*classe em calibre.utils.formatter_functions*), 199
BuiltinRange (*classe em calibre.utils.formatter_functions*), 198
BuiltinRatingToStars (*classe em calibre.utils.formatter_functions*), 188
BuiltinRawField (*classe em calibre.utils.formatter_functions*), 192
BuiltinRawList (*classe em calibre.utils.formatter_functions*), 192
BuiltinRe (*classe em calibre.utils.formatter_functions*), 201
BuiltinReGroup (*classe em calibre.utils.formatter_functions*), 202
BuiltinRound (*classe em calibre.utils.formatter_functions*), 186
BuiltinSelect (*classe em calibre.utils.formatter_functions*), 195
BuiltinSeriesSort (*classe em calibre.utils.formatter_functions*), 192
BuiltinSetGlobals (*classe em calibre.utils.formatter_functions*), 204
BuiltinShorten (*classe em calibre.utils.formatter_functions*), 202
BuiltinStrcat (*classe em calibre.utils.formatter_functions*), 202
BuiltinStrcatMax (*classe em calibre.utils.formatter_functions*), 202
BuiltinStrcmp (*classe em calibre.utils.formatter_functions*), 200
BuiltinStrcmpcase (*classe em calibre.utils.formatter_functions*), 200
BuiltinStrInList (*classe em calibre.utils.formatter_functions*), 195
BuiltinStrlen (*classe em calibre.utils.formatter_functions*), 202
BuiltinSubitems (*classe em calibre.utils.formatter_functions*), 198
BuiltinSublist (*classe em calibre.utils.formatter_functions*), 198
BuiltinSubstr (*classe em calibre.utils.formatter_functions*), 203
BuiltinSubtract (*classe em calibre.utils.formatter_functions*), 186
BuiltinSwapAroundArticles (*classe em calibre.utils.formatter_functions*), 203
BuiltinSwapAroundComma (*classe em calibre.utils.formatter_functions*), 203
BuiltinSwitch (*classe em calibre.utils.formatter_functions*), 194
BuiltinSwitchIf (*classe em calibre.utils.formatter_functions*), 194
BuiltinTemplate (*classe em calibre.utils.formatter_functions*), 200
BuiltinTest (*classe em calibre.utils.formatter_functions*), 194
BuiltinTitlecase (*classe em calibre.utils.formatter_functions*), 201
BuiltinToday (*classe em calibre.utils.formatter_functions*), 187
BuiltinToHex (*classe em calibre.utils.formatter_functions*), 203
BuiltinTransliterate (*classe em calibre.utils.formatter_functions*), 203
BuiltinUppercase (*classe em calibre.utils.formatter_functions*), 201
BuiltinUrlsFromIdentifiers (*classe em calibre.utils.formatter_functions*), 188
BuiltinUserCategories (*classe em calibre.utils.formatter_functions*), 192
BuiltinVirtualLibraries (*classe em calibre.utils.formatter_functions*), 193

C

Cache (*classe em calibre.db.cache*), 345
Cache.EventType (*classe em calibre.db.cache*), 345
cached_cover_url_is_reliable (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
calibre command line option
    --detach, 296
    --help, 296
    --ignore-plugins, 296

--no-update-check, 296
--shutdown-running-calibre, 296
--start-in-tray, 296
--verbose, 296
--version, 296
--with-library, 296
-h, 296
-s, 296
-v, 296
calibre.customize
módulo, 250
calibre.customize.conversion
módulo, 259
calibre.db.cache
módulo, 345
calibre.devices.interface
módulo, 262
calibre.ebooks.metadata.book.base
módulo, 204
calibre.ebooks.metadata.sources.base
módulo, 256
calibre.ebooks.oeb.polish.container
módulo, 352
calibre.ebooks.oeb.polish.cover
módulo, 359
calibre.ebooks.oeb.polish.css
módulo, 359
calibre.ebooks.oeb.polish.jacket
módulo, 358
calibre.ebooks.oeb.polish.pretty
módulo, 358
calibre.ebooks.oeb.polish.replace
módulo, 357
calibre.ebooks.oeb.polish.split
módulo, 358
calibre.ebooks.oeb.polish.toc
módulo, 360
calibre.gui2.tweak_book.boss
módulo, 362
calibre.utils.formatter_functions
módulo, 185
calibre.web.feeds.news
módulo, 38
calibre-customize command line option
--add-plugin, 296
--build-plugin, 296
--customize-plugin, 296
--disable-plugin, 296
--enable-plugin, 296
--help, 297
--list-plugins, 297
--remove-plugin, 297
--version, 297
-a, 296
-b, 296
-h, 297
-l, 297
-r, 297
calibre-debug command line option
--add-simple-plugin, 297
--command, 297
--debug-device-driver, 297
--default-programs, 297
--diff, 297
--edit-book, 297
--exec-file, 298
--explode-book, 298
--export-all-calibre-data, 298
--fix-multiprocessing, 298
--gui, 298
--gui-debug, 298
--help, 298
--implode-book, 298
--import-calibre-data, 298
--inspect-mobi, 298
--paths, 298
--run-plugin, 298
--run-test, 298
--run-without-debug, 298
--shutdown-running-calibre, 299
--subset-font, 299
--test-build, 299
--version, 299
--viewer, 299
-c, 297
-d, 297
-e, 298
-f, 299
-g, 298
-h, 298
-i, 298
-m, 298
-r, 298
-s, 299
-t, 298
-w, 299
-x, 298
calibre-server command line option
--access-log, 299
--ajax-timeout, 299
--auth-mode, 299
--auto-reload, 299
--ban-after, 299
--ban-for, 299
--book-list-mode, 300
--compress-min-size, 300
--custom-list-template, 300
--daemonize, 300

--disable-allow-socket--preallocation, 300
--disable-auth, 300
--disable-fallback-to-detected--interface, 300
--disable-local-write, 300
--disable-log-not-found, 300
--disable-use-bonjour, 300
--disable-use-sendfile, 301
--displayed-fields, 300
--enable-allow-socket--preallocation, 300
--enable-auth, 300
--enable-fallback-to-detected--interface, 300
--enable-local-write, 300
--enable-log-not-found, 300
--enable-use-bonjour, 300
--enable-use-sendfile, 301
--help, 301
--ignored-fields, 301
--listen-on, 301
--log, 301
--manage-users, 301
--max-header-line-size, 301
--max-job-time, 301
--max-jobs, 301
--max-log-size, 301
--max-opds-items, 301
--max-opds-ungrouped-items, 301
--max-request-body-size, 301
--num-per-page, 302
--pidfile, 302
--port, 302
--search-the-net-urls, 302
--shutdown-timeout, 302
--ssl-certfile, 302
--ssl-keyfile, 302
--timeout, 302
--trusted-ips, 302
--url-prefix, 302
--userdb, 302
--version, 302
--worker-count, 302
-h, 301
calibre-smtp command line option
--attachment, 303
--cafile, 304
--dont-verify-server-certificate, 304
--encryption-method, 304
--fork, 303
--help, 303
--localhost, 303
--outbox, 303
--password, 304
--port, 304
--relay, 304
--subject, 303
--timeout, 303
--username, 304
--verbose, 303
--version, 303
-a, 303
-e, 304
-f, 303
-h, 303
-l, 303
-o, 303
-p, 304
-r, 304
-s, 303
-t, 303
-u, 304
-v, 303
calibredb-add command line option
-1, 308
-I, 308
-S, 308
-T, 308
--add, 308
--authors, 307
--automerge, 307
--cover, 307
--duplicates, 307
--empty, 307
--identifier, 308
--ignore, 308
--isbn, 308
--languages, 308
--one-book-per-directory, 308
--recurse, 308
--series, 308
--series-index, 308
--tags, 308
--title, 308
-a, 307
-c, 307
-d, 307
-e, 307
-i, 308
-l, 308
-m, 307
-r, 308
-s, 308
-t, 308
calibredb-add_custom_column command line option

--display, 314
--is-multiple, 314
calibredb-add_format command line option
--dont-replace, 309
calibredb-backup_metadata command line option
--all, 316
calibredb-catalog command line option
--catalog-title, 312
--cross-reference-authors, 312
--debug-pipeline, 312
--exclude-genre, 312
--exclusion-rules, 312
--generate-authors, 312
--generate-descriptions, 312
--generate-genres, 312
--generate-recently-added, 312
--generate-series, 312
--generate-titles, 312
--genre-source-field, 312
--header-note-source-field, 312
--ids, 311
--merge-comments-rule, 312
--output-profile, 313
--prefix-rules, 313
--preset, 313
--search, 311
--thumb-width, 313
--use-existing-cover, 313
--verbose, 311
-i, 311
-s, 311
-v, 311
calibredb-check_library command line option
--csv, 315
--ignore_extensions, 315
--ignore_names, 315
--report, 315
--vacuum-fts-db, 315
-c, 315
-e, 315
-n, 315
-r, 315
calibredb-custom_columns command line option
--details, 314
-d, 314
calibredb-embed_metadata command line option
--only-formats, 317
-f, 317
calibredb-export command line option
--all, 310
--dont-asciiize, 310
--dont-save-cover, 310
--dont-update-metadata, 310
--dont-write-opf, 310
--formats, 310
--progress, 311
--replace-whitespace, 311
--single-dir, 311
--template, 311
--timefmt, 311
--to-dir, 311
--to-lowercase, 311
calibredb-fts_index command line option
--indexing-speed, 318
--wait-for-completion, 318
calibredb-fts_search command line option
--do-not-match-on-related-words, 318
--include-snippets, 318
--indexing-threshold, 318
--match-end-marker, 318
--match-start-marker, 318
--output-format, 318
--restrict-to, 318
calibredb-list command line option
--ascending, 306
--fields, 306
--for-machine, 306
--limit, 306
--line-width, 306
--prefix, 307
--search, 307
--separator, 307
--sort-by, 307
--template, 307
--template_file, 307
--template_heading, 307
-f, 306
-s, 307
-t, 307
-w, 306
calibredb-list_categories command line option
--categories, 316
--csv, 316
--dialect, 316
--item_count, 316
--width, 316
-c, 316
-i, 316
-r, 316
-w, 316
calibredb-remove command line option

--permanent, 309
calibredb-remove_custom_column command line option
--force, 314
-f, 314
calibredb-restore_database command line option
--really-do-it, 315
-r, 315
calibredb-search command line option
--limit, 317
-l, 317
calibredb-set_custom command line option
--append, 315
-a, 315
calibredb-set_metadata command line option
--field, 310
--list-fields, 310
-f, 310
-l, 310
calibredb-show_metadata command line option
--as-opf, 309
can_be_disabled (*atributo calibre.customize.conversion.InputFormatPlugin*), 259
can_be_disabled (*atributo calibre.customize.conversion.OutputFormatPlugin*), 261
can_be_disabled (*atributo calibre.customize.InterfaceActionBase*), 279
can_be_disabled (*atributo calibre.customize.Plugin*), 251
can_be_disabled (*atributo calibre.customize.PreferencesPlugin*), 279
CAN_DO_DEVICE_DB_PLUGBOARD (*atributo calibre.devices.interface.DevicePlugin*), 263
can_get_multiple_covers (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
can_handle() (*método calibre.devices.interface.DevicePlugin*), 265
can_handle_windows() (*método calibre.devices.interface.DevicePlugin*), 264
can_handle_windows() (*método calibre.devices.usbms.device.Device*), 272
CAN_SET_METADATA (*atributo calibre.devices.interface.DevicePlugin*), 263
CAN_SET_METADATA (*atributo calibre.devices.usbms.driver.USBMS*), 274
canonicalize_internal_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 38
capabilities (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
card_prefix() (*método calibre.devices.interface.DevicePlugin*), 266
card_prefix() (*método calibre.devices.usbms.device.Device*), 272
CatalogPlugin (*classe em calibre.customize*), 255
category (*atributo calibre.customize.PreferencesPlugin*), 280
category_order (*atributo calibre.customize.PreferencesPlugin*), 279
center_navbar (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
change_font() (*no módulo calibre.ebooks.oeb.polish.fonts*), 359
changed_signal (*atributo calibre.gui2.preferences.ConfigWidgetBase*), 281
changed_signal (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 280
clean_downloaded_metadata() (*método calibre.ebooks.metadata.sources.base.Source*), 257
cleanup() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
CLI (*classe em calibre.devices.usbms.cli*), 273
cli_main() (*método calibre.customize.Plugin*), 252
cli_options (*atributo calibre.customize.CatalogPlugin*), 255
clone_browser() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
close_editor() (*método calibre.gui2.tweak_book.boss.Boss*), 362
command line option
--help, 306
--library-path, 306
--password, 306
--timeout, 306
--username, 306
--version, 306
--with-library, 306
-h, 306
commit() (*método calibre.ebooks.oeb.polish.container.Container*), 353
commit() (*método calibre.gui2.preferences.ConfigWidgetBase*), 281
commit() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 280
commit_all_editors_to_container() (*método calibre.gui2.tweak_book.boss.Boss*), 362
commit_item() (*método cali-*

bre.ebooks.oeb.polish.container.Container), 354
common_options (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
common_options (*atributo calibre.customize.conversion.OutputFormatPlugin*), 261
compress_covers() (*método calibre.db.cache.Cache*), 347
compress_news_images (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
compress_news_images_auto_size (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
compress_news_images_max_size (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
config_help_message (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
config_widget (*atributo calibre.customize.PreferencesPlugin*), 279
config_widget() (*método calibre.customize.Plugin*), 251
config_widget() (*método calibre.ebooks.metadata.sources.base.Source*), 257
config_widget() (*método de classe calibre.devices.interface.DevicePlugin*), 267
ConfigWidgetBase (*classe em calibre.gui2.preferences*), 281
ConfigWidgetInterface (*classe em calibre.gui2.preferences*), 280
Container (*classe em calibre.ebooks.oeb.polish.container*), 353
contains(), **157**
conversion_options (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
convert() (*método calibre.customize.conversion.InputFormatPlugin*), 260
convert() (*método calibre.customize.conversion.OutputFormatPlugin*), 261
copy_cover_to() (*método calibre.db.cache.Cache*), 347
copy_format_to() (*método calibre.db.cache.Cache*), 347
core_usage (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
cover() (*método calibre.db.cache.Cache*), 347
cover_margins (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
create_action() (*método calibre.gui2.tweak_book.plugin.Tool*), 361
create_inline_toc() (*no módulo calibre.ebooks.oeb.polish.toc*), 360
create_menu_action() (*método calibre.gui2.actions.InterfaceAction*), 277
create_widget() (*método calibre.customize.PreferencesPlugin*), 280
CSS, **369**
current_container (*calibre.gui2.tweak_book.plugin.Tool property*), 361
currently_editing (*calibre.gui2.tweak_book.boss.Boss property*), 362
custom_field_keys() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
customization_help() (*método calibre.customize.Plugin*), 252
customization_help() (*método calibre.ebooks.metadata.sources.base.Source*), 256

D

data_for_find_identical_books() (*método calibre.db.cache.Cache*), 348
data_for_has_book() (*método calibre.db.cache.Cache*), 348
debug_managed_device_detection() (*método calibre.devices.interface.DevicePlugin*), 264
deepcopy() (*método calibre.ebooks.metadata.book.base.Metadata*), 204
default_cover() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
delay (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
delete_books() (*método calibre.devices.interface.DevicePlugin*), 267
delete_books() (*método calibre.devices.usbms.driver.USBMS*), 275
delete_custom_book_data() (*método calibre.db.cache.Cache*), 348
description (*atributo calibre.customize.Plugin*), 250
description (*atributo calibre.customize.PreferencesPlugin*), 280
description (*atributo calibre.devices.usbms.driver.USBMS*), 274
description (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
description (*calibre.customize.conversion.OutputFormatPlugin property*), 261
detect_managed_devices() (*método calibre.devices.interface.DevicePlugin*), 264

Device (*classe em calibre.devices.usbms.device*), 271
DevicePlugin (*classe em calibre.devices.interface*), 262
dirty() (*método calibre.ebooks.oeb.polish.container.Container*), 354
do_user_config() (*método calibre.customize.Plugin*), 251
dont_add_to (*atributo calibre.gui2.actions.InterfaceAction*), 277
dont_remove_from (*atributo calibre.gui2.actions.InterfaceAction*), 277
download() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
download_cover() (*método calibre.ebooks.metadata.sources.base.Source*), 259
drop_event() (*método calibre.gui2.actions.InterfaceAction*), 277

E

ebook-convert command line option
    --asciiize, 321
    --author-sort, 326
    --authors, 326
    --base-font-size, 321
    --book-producer, 326
    --change-justification, 321
    --chapter, 325
    --chapter-mark, 325
    --comments, 326
    --cover, 327
    --debug-pipeline, 327
    --disable-dehyphenate, 324
    --disable-delete-blank-paragraphs, 324
    --disable-fix-indents, 324
    --disable-font-rescaling, 321
    --disable-format-scene-breaks, 324
    --disable-italicize-common-cases, 324
    --disable-markup-chapter-headings, 324
    --disable-remove-fake-margins, 325
    --disable-renumber-headings, 324
    --disable-unwrap-lines, 324
    --dont-split-on-page-breaks, 320
    --duplicate-links-in-toc, 326
    --embed-all-fonts, 321
    --embed-font-family, 321
    --enable-heuristics, 324
    --epub-flatten, 320
    --epub-inline-toc, 320
    --epub-toc-at-end, 320
    --epub-version, 320
    --expand-css, 321
    --extra-css, 321
    --extract-to, 320
    --filter-css, 322
    --flow-size, 320
    --font-size-mapping, 322
    --from-opf, 327
    --help, 319
    --html-unwrap-factor, 324
    --input-encoding, 320
    --input-profile, 319
    --insert-blank-line, 322
    --insert-blank-line-size, 322
    --insert-metadata, 325
    --isbn, 327
    --keep-ligatures, 322
    --language, 327
    --level1-toc, 326
    --level2-toc, 326
    --level3-toc, 326
    --line-height, 322
    --linearize-tables, 322
    --list-recipes, 319
    --margin-bottom, 322
    --margin-left, 322
    --margin-right, 322
    --margin-top, 322
    --max-toc-links, 326
    --minimum-line-height, 323
    --no-chapters-in-toc, 326
    --no-default-epub-cover, 320
    --no-svg-cover, 320
    --output-profile, 319
    --page-breaks-before, 325
    --prefer-metadata-cover, 325
    --preserve-cover-aspect-ratio, 320
    --pretty-print, 321
    --pubdate, 327
    --publisher, 327
    --rating, 327
    --read-metadata-from-opf, 327
    --remove-first-image, 325
    --remove-paragraph-spacing, 323
    --remove-paragraph-spacing-indent-size, 323
    --replace-scene-breaks, 324
    --search-replace, 324
    --series, 327
    --series-index, 327
    --smarten-punctuation, 323
    --sr1-replace, 324
    --sr1-search, 324
    --sr2-replace, 325
    --sr2-search, 325

--sr3-replace, 325
--sr3-search, 325
--start-reading-at, 325
--subset-embedded-fonts, 323
--tags, 327
--timestamp, 327
--title, 327
--title-sort, 327
--toc-filter, 326
--toc-threshold, 326
--toc-title, 321
--transform-css-rules, 323
--transform-html-rules, 323
--unsmarten-punctuation, 323
--use-auto-toc, 326
--verbose, 327
--version, 320
-d, 327
-h, 319
-m, 327
-v, 327
ebook-edit command line option
--detach, 328
--help, 328
--select-text, 328
--version, 328
-h, 328
ebook-meta command line option
--author-sort, 328
--authors, 328
--book-producer, 328
--category, 329
--comments, 329
--cover, 329
--date, 329
--from-opf, 329
--get-cover, 329
--help, 329
--identifier, 329
--index, 329
--isbn, 329
--language, 329
--lrf-bookid, 329
--publisher, 329
--rating, 329
--series, 329
--tags, 329
--title, 329
--title-sort, 329
--to-opf, 330
--version, 330
-a, 328
-c, 329
-d, 329
-h, 329
-i, 329
-k, 328
-l, 329
-p, 329
-r, 329
-s, 329
-t, 329
ebook-polish command line option
-H, 330
-U, 331
--add-soft-hyphens, 330
--compress-images, 330
--cover, 330
--embed-fonts, 330
--help, 330
--jacket, 330
--opf, 330
--remove-jacket, 330
--remove-soft-hyphens, 331
--remove-unused-css, 331
--smarten-punctuation, 331
--subset-fonts, 331
--upgrade-book, 331
--verbose, 331
--version, 331
-c, 330
-e, 330
-f, 331
-h, 330
-i, 330
-j, 330
-o, 330
-p, 331
-u, 331
ebook-viewer command line option
--continue, 332
--detach, 332
--force-reload, 332
--full-screen, 332
--fullscreen, 332
--help, 332
--open-at, 332
--raise-window, 332
--version, 332
-f, 332
-h, 332
edit_file() (*método calibre.gui2.tweak_book.boss.Boss*), 362
eject() (*método calibre.devices.interface.DevicePlugin*), 265
eject() (*método calibre.devices.usbms.device.Device*), 273

embed_metadata() (*método calibre.db.cache.Cache*), 348
encoding (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 43
exists() (*método calibre.ebooks.oeb.polish.container.Container*), 354
extra_css (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 44
extract_readable_article() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39

F

fast_field_for() (*método calibre.db.cache.Cache*), 348
feeds (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 44
fetch-ebook-metadata command line option
-I, 333
--allowed-plugin, 332
--authors, 332
--cover, 332
--help, 333
--identifier, 333
--isbn, 333
--opf, 333
--timeout, 333
--title, 333
--verbose, 333
--version, 333
-a, 332
-c, 332
-d, 333
-h, 333
-i, 333
-o, 333
-p, 332
-t, 333
-v, 333
field_for() (*método calibre.db.cache.Cache*), 348
field_ids_for() (*método calibre.db.cache.Cache*), 348
file_type (*atributo calibre.customize.conversion.OutputFormatPlugin*), 261
file_types (*atributo calibre.customize.CatalogPlugin*), 255
file_types (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
file_types (*atributo calibre.customize.FileTypePlugin*), 252
file_types (*atributo calibre.customize.MetadataReaderPlugin*), 254
file_types (*atributo calibre.customize.MetadataWriterPlugin*), 254
filename_callback() (*método calibre.devices.usbms.device.Device*), 273
filesize() (*método calibre.ebooks.oeb.polish.container.Container*), 354
FileTypePlugin (*classe em calibre.customize*), 252
filter_css() (*no módulo calibre.ebooks.oeb.polish.css*), 360
filter_regexps (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 44
find_identical_books() (*método calibre.db.cache.Cache*), 348
fix_all_html() (*no módulo calibre.ebooks.oeb.polish.pretty*), 358
fix_html() (*no módulo calibre.ebooks.oeb.polish.pretty*), 358
for_viewer (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
format() (*método calibre.db.cache.Cache*), 348
format_abspath() (*método calibre.db.cache.Cache*), 348
format_added (*atributo calibre.db.cache.Cache.EventType*), 346
format_field() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
format_hash() (*método calibre.db.cache.Cache*), 349
format_metadata() (*método calibre.db.cache.Cache*), 349
FORMATS (*atributo calibre.devices.interface.DevicePlugin*), 262
FORMATS (*atributo calibre.devices.usbms.driver.USBMS*), 274
formats() (*método calibre.db.cache.Cache*), 349
formats_removed (*atributo calibre.db.cache.Cache.EventType*), 346
free_space() (*método calibre.devices.interface.DevicePlugin*), 266
free_space() (*método calibre.devices.usbms.device.Device*), 272
from_files() (*no módulo calibre.ebooks.oeb.polish.toc*), 360
from_links() (*no módulo calibre.ebooks.oeb.polish.toc*), 360
from_xpaths() (*no módulo calibre.ebooks.oeb.polish.toc*), 360

G

generate_item() (*método cali-*

bre.ebooks.oeb.polish.container.Container), 354
genesis() (*método calibre.gui2.actions.InterfaceAction*), 278
genesis() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 280
get_all_standard_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
get_all_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
get_annotations() (*método calibre.devices.usbms.device.Device*), 273
get_article_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
get_author_tokens() (*método calibre.ebooks.metadata.sources.base.Source*), 257
get_book_url() (*método calibre.ebooks.metadata.sources.base.Source*), 257
get_book_url_name() (*método calibre.ebooks.metadata.sources.base.Source*), 257
get_book_urls() (*método calibre.ebooks.metadata.sources.base.Source*), 257
get_browser() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
get_cached_cover_url() (*método calibre.ebooks.metadata.sources.base.Source*), 258
get_categories() (*método calibre.db.cache.Cache*), 349
get_collections() (*método calibre.devices.interface.BookList*), 270
get_cover_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
get_custom_book_data() (*método calibre.db.cache.Cache*), 349
get_device_information() (*método calibre.devices.interface.DevicePlugin*), 265
get_device_information() (*método calibre.devices.usbms.driver.USBMS*), 274
get_device_uid() (*método calibre.devices.interface.DevicePlugin*), 268
get_driveinfo() (*método calibre.devices.interface.DevicePlugin*), 266
get_extra_css() (*método calibre.web.feeds.news.BasicNewsRecipe*), 39
get_feeds() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
get_file() (*método calibre.devices.interface.DevicePlugin*), 267
get_file_path_for_processing() (*método calibre.ebooks.oeb.polish.container.Container*), 354
get_id_map() (*método calibre.db.cache.Cache*), 349
get_identifiers() (*método calibre.ebooks.metadata.book.base.Metadata*), 204
get_ids_for_custom_book_data() (*método calibre.db.cache.Cache*), 349
get_images() (*método calibre.customize.conversion.InputFormatPlugin*), 260
get_item_id() (*método calibre.db.cache.Cache*), 349
get_item_ids() (*método calibre.db.cache.Cache*), 349
get_item_name() (*método calibre.db.cache.Cache*), 349
get_masthead_title() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
get_masthead_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
get_metadata() (*método calibre.customize.MetadataReaderPlugin*), 254
get_metadata() (*método calibre.db.cache.Cache*), 349
get_next_series_num_for() (*método calibre.db.cache.Cache*), 349
get_obfuscated_article() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
get_open_popup_message() (*método de classe calibre.devices.interface.DevicePlugin*), 264
get_option() (*método calibre.devices.interface.DevicePlugin*), 269
get_proxy_metadata() (*método calibre.db.cache.Cache*), 350
get_recommended_folders() (*no módulo calibre.ebooks.oeb.polish.replace*), 357
get_standard_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
get_title_tokens() (*método calibre.ebooks.metadata.sources.base.Source*), 257
get_usage_count_by_id() (*método calibre.db.cache.Cache*), 350
get_user_blacklisted_devices() (*método calibre.devices.interface.DevicePlugin*), 268
get_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
gui (*calibre.gui2.tweak_book.plugin.Tool property*), 361
gui_category (*atributo calibre.customize.PreferencesPlugin*), 280

gui_configuration_widget() (*método calibre.customize.conversion.InputFormatPlugin*), 261
gui_configuration_widget() (*método calibre.customize.conversion.OutputFormatPlugin*), 262
gui_layout_complete() (*método calibre.gui2.actions.InterfaceAction*), 278
gui_name (*atributo calibre.customize.PreferencesPlugin*), 280
guide_type_map (*calibre.ebooks.oeb.polish.container.Container property*), 354

H

handle_gzip (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 44
has_book() (*método calibre.db.cache.Cache*), 350
has_format() (*método calibre.db.cache.Cache*), 350
has_html_comments (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
has_id() (*método calibre.db.cache.Cache*), 350
has_name() (*método calibre.ebooks.oeb.polish.container.Container*), 354
href_to_name() (*método calibre.ebooks.oeb.polish.container.Container*), 354
HTML, **369**

I

icon (*atributo calibre.customize.PreferencesPlugin*), 280
icon (*atributo calibre.devices.interface.DevicePlugin*), 263
id_from_url() (*método calibre.ebooks.metadata.sources.base.Source*), 258
identify() (*método calibre.ebooks.metadata.sources.base.Source*), 258
identify_results_keygen() (*método calibre.ebooks.metadata.sources.base.Source*), 258
ignore_connected_device() (*método calibre.devices.interface.DevicePlugin*), 268
ignore_duplicate_articles (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 44
ignore_ssl_errors (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
image_url_processor() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 40
index_to_soup() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
indexing_progress_changed (*atributo calibre.db.cache.Cache.EventType*), 346
init() (*método calibre.db.cache.Cache*), 350
initial_tab_changed() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 281
initialization_complete() (*método calibre.gui2.actions.InterfaceAction*), 278
initialize() (*método calibre.customize.CatalogPlugin*), 255
initialize() (*método calibre.customize.Plugin*), 251
initialize() (*método calibre.gui2.preferences.ConfigWidgetBase*), 281
initialize() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 280
InputFormatPlugin (*classe em calibre.customize.conversion*), 259
insert_into_xml() (*método calibre.ebooks.oeb.polish.container.Container*), 354
installation_type (*atributo calibre.customize.Plugin*), 251
InterfaceAction (*classe em calibre.gui2.actions*), 276
InterfaceActionBase (*classe em calibre.customize*), 279
InternalMetadataCompareKeyGen (*classe em calibre.ebooks.metadata.sources.base*), 259
is_configured() (*método calibre.ebooks.metadata.sources.base.Source*), 256
is_dir (*atributo calibre.ebooks.oeb.polish.container.Container*), 354
is_dynamically_controllable() (*método calibre.devices.interface.DevicePlugin*), 269
is_image_collection (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
is_link_wanted() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
is_null() (*método calibre.ebooks.metadata.book.base.Metadata*), 204
is_running() (*método calibre.devices.interface.DevicePlugin*), 269
is_usb_connected() (*método calibre.devices.interface.DevicePlugin*), 264
items_removed (*atributo calibre.db.cache.Cache.EventType*), 346
items_renamed (*atributo calibre.db.cache.Cache.EventType*), 346
iterlinks() (*método calibre.ebooks.oeb.polish.container.Container*),

354

K

keep_only_tags (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 44

L

language (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 44
library_about_to_change() (*método calibre.gui2.actions.InterfaceAction*), 278
library_changed() (*método calibre.gui2.actions.InterfaceAction*), 278
load_actual_plugin() (*método calibre.customize.InterfaceActionBase*), 279
load_resources() (*método calibre.customize.Plugin*), 251
load_resources() (*método calibre.gui2.actions.InterfaceAction*), 278
location_selected() (*método calibre.gui2.actions.InterfaceAction*), 278
LRF, **369**
lrf2lrs command line option
    --dont-output-resources, 333
    --help, 333
    --output, 333
    --verbose, 333
    --version, 333
    -h, 333
    -o, 333
lrfviewer command line option
    --disable-hyphenation, 334
    --help, 334
    --profile, 334
    --verbose, 334
    --version, 334
    --visual-debug, 334
    --white-background, 334
    -h, 334
lrs2lrf command line option
    --help, 334
    --lrs, 334
    --output, 334
    --verbose, 334
    --version, 335
    -h, 334
    -o, 334

M

make_name_unique() (*método calibre.ebooks.oeb.polish.container.Container*), 354
MANAGES_DEVICE_PRESENCE (*atributo calibre.devices.interface.DevicePlugin*), 263
manifest_has_name() (*método calibre.ebooks.oeb.polish.container.Container*), 355
manifest_id_map (*calibre.ebooks.oeb.polish.container.Container property*), 355
manifest_items_of_type() (*método calibre.ebooks.oeb.polish.container.Container*), 355
manifest_items_with_property() (*método calibre.ebooks.oeb.polish.container.Container*), 355
manifest_type_map (*calibre.ebooks.oeb.polish.container.Container property*), 355
mark_as_cover() (*no módulo calibre.ebooks.oeb.polish.cover*), 359
mark_as_titlepage() (*no módulo calibre.ebooks.oeb.polish.cover*), 359
masthead_url (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
match_regexps (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
max_articles_per_feed (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
MAX_PATH_LEN (*atributo calibre.devices.usbms.device.Device*), 271
merge() (*no módulo calibre.ebooks.oeb.polish.split*), 358
Metadata (*classe em calibre.ebooks.metadata.book.base*), 204
metadata_changed (*atributo calibre.db.cache.Cache.EventType*), 346
metadata_for_field() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
MetadataReaderPlugin (*classe em calibre.customize*), 254
MetadataWriterPlugin (*classe em calibre.customize*), 254
mi (*calibre.ebooks.oeb.polish.container.Container property*), 355
minimum_calibre_version (*atributo calibre.customize.Plugin*), 251
multisort() (*método calibre.db.cache.Cache*), 350
multisplit() (*no módulo calibre.ebooks.oeb.polish.split*), 358
módulo
    calibre.customize, 250
    calibre.customize.conversion, 259
    calibre.db.cache, 345
    calibre.devices.interface, 262
    calibre.ebooks.metadata.book.base, 204
    calibre.ebooks.metadata.sources.base, 256

calibre.ebooks.oeb.polish.container, 352
calibre.ebooks.oeb.polish.cover, 359
calibre.ebooks.oeb.polish.css, 359
calibre.ebooks.oeb.polish.jacket, 358
calibre.ebooks.oeb.polish.pretty, 358
calibre.ebooks.oeb.polish.replace, 357
calibre.ebooks.oeb.polish.split, 358
calibre.ebooks.oeb.polish.toc, 360
calibre.gui2.tweak_book.boss, 362
calibre.utils.formatter_functions, 185
calibre.web.feeds.news, 38

N

name (*atributo calibre.customize.Plugin*), 250
name (*atributo calibre.gui2.actions.InterfaceAction*), 276
name (*atributo calibre.gui2.tweak_book.plugin.Tool*), 361
name(), **157**
name_order (*atributo calibre.customize.PreferencesPlugin*), 279
name_to_abspath() (*método calibre.ebooks.oeb.polish.container.Container*), 355
name_to_href() (*método calibre.ebooks.oeb.polish.container.Container*), 355
names_that_must_not_be_changed (*calibre.ebooks.oeb.polish.container.Container property*), 355
names_that_must_not_be_removed (*calibre.ebooks.oeb.polish.container.Container property*), 355
names_that_need_not_be_manifested (*calibre.ebooks.oeb.polish.container.Container property*), 355
needs_subscription (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
NEWS_IN_FOLDER (*atributo calibre.devices.usbms.device.Device*), 271
no_stylesheets (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
normalize_path() (*método de classe calibre.devices.usbms.driver.USBMS*), 275
NUKE_COMMENTS (*atributo calibre.devices.interface.DevicePlugin*), 263

O

oldest_article (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
on_import (*atributo calibre.customize.FileTypePlugin*), 252
on_postconvert (*atributo calibre.customize.FileTypePlugin*), 252
on_postdelete (*atributo calibre.customize.FileTypePlugin*), 252
on_postimport (*atributo calibre.customize.FileTypePlugin*), 252
on_postprocess (*atributo calibre.customize.FileTypePlugin*), 253
on_preprocess (*atributo calibre.customize.FileTypePlugin*), 252
open() (*método calibre.devices.interface.DevicePlugin*), 265
open() (*método calibre.devices.usbms.device.Device*), 273
open() (*método calibre.ebooks.oeb.polish.container.Container*), 355
open_book() (*método calibre.gui2.tweak_book.boss.Boss*), 362
OPEN_FEEDBACK_MESSAGE (*atributo calibre.devices.interface.DevicePlugin*), 263
opf (*calibre.ebooks.oeb.polish.container.Container property*), 355
opf_get_or_create() (*método calibre.ebooks.oeb.polish.container.Container*), 355
opf_version (*calibre.ebooks.oeb.polish.container.Container property*), 355
opf_version_parsed (*calibre.ebooks.oeb.polish.container.Container property*), 355
opf_xpath() (*método calibre.ebooks.oeb.polish.container.Container*), 356
options (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
options (*atributo calibre.customize.conversion.OutputFormatPlugin*), 261
options (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
OSX_MAIN_MEM_VOL_PAT (*atributo calibre.devices.usbms.device.Device*), 271
output_encoding (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
OutputFormatPlugin (*classe em calibre.customize.conversion*), 261

P

parse_feeds() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
parse_index() (*método calibre.web.feeds.news.BasicNewsRecipe*), 40
parsed() (*método calibre.ebooks.oeb.polish.container.Container*), 356
path_sep (*atributo calibre.devices.interface.DevicePlugin*), 263
Plugin (*classe em calibre.customize*), 250
populate_article_metadata() (*método calibre.web.feeds.news.BasicNewsRecipe*), 41
popup_type (*atributo calibre.gui2.actions.InterfaceAction*), 276
post_yank_cleanup() (*método calibre.devices.interface.DevicePlugin*), 265
post_yank_cleanup() (*método calibre.devices.usbms.device.Device*), 273
postadd() (*método calibre.customize.FileTypePlugin*), 254
postconvert() (*método calibre.customize.FileTypePlugin*), 253
postdelete() (*método calibre.customize.FileTypePlugin*), 253
postimport() (*método calibre.customize.FileTypePlugin*), 253
postprocess_book() (*método calibre.customize.conversion.InputFormatPlugin*), 261
postprocess_book() (*método calibre.web.feeds.news.BasicNewsRecipe*), 41
postprocess_html() (*método calibre.web.feeds.news.BasicNewsRecipe*), 41
pref() (*método calibre.db.cache.Cache*), 350
prefer_results_with_isbn (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
PreferencesPlugin (*classe em calibre.customize*), 279
prepare_addable_books() (*método calibre.devices.interface.DevicePlugin*), 268
preprocess_html() (*método calibre.web.feeds.news.BasicNewsRecipe*), 41
preprocess_image() (*método calibre.web.feeds.news.BasicNewsRecipe*), 41
preprocess_raw_html() (*método calibre.web.feeds.news.BasicNewsRecipe*), 41
preprocess_regexps (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
pretty_all() (*no módulo calibre.ebooks.oeb.polish.pretty*), 358
pretty_css() (*no módulo calibre.ebooks.oeb.polish.pretty*), 358
pretty_html() (*no módulo calibre.ebooks.oeb.polish.pretty*), 358
pretty_xml() (*no módulo calibre.ebooks.oeb.polish.pretty*), 358
print_version() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 42
priority (*atributo calibre.customize.Plugin*), 251
priority (*atributo calibre.gui2.actions.InterfaceAction*), 276
PRODUCT_ID (*atributo calibre.devices.interface.DevicePlugin*), 262
PRODUCT_ID (*atributo calibre.devices.usbms.device.Device*), 271
publication_date() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
publication_type (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45

R

raw_data() (*método calibre.ebooks.oeb.polish.container.Container*), 356
re:test(), **157**
read_backup() (*método calibre.db.cache.Cache*), 350
receita, **369**
recipe_disabled (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
recommendations (*atributo calibre.customize.conversion.InputFormatPlugin*), 260
recommendations (*atributo calibre.customize.conversion.OutputFormatPlugin*), 261
recursions (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
refresh_gui() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 281
regexp, **369**
register() (*método calibre.gui2.preferences.ConfigWidgetBase*), 281
register_shortcut() (*método calibre.gui2.tweak_book.plugin.Tool*), 361
relpath() (*método calibre.ebooks.oeb.polish.container.Container*), 356
remove_attributes (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 45
remove_book() (*método calibre.devices.interface.BookList*), 270
remove_books() (*método calibre.db.cache.Cache*), 350
remove_books_from_metadata() (*método calibre.devices.usbms.driver.USBMS*), 275

remove_books_from_metadata() (*método de classe calibre.devices.interface.DevicePlugin*), 267
remove_empty_feeds (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
remove_formats() (*método calibre.db.cache.Cache*), 350
remove_from_spine() (*método calibre.ebooks.oeb.polish.container.Container*), 356
remove_from_xml() (*método calibre.ebooks.oeb.polish.container.Container*), 356
remove_item() (*método calibre.ebooks.oeb.polish.container.Container*), 356
remove_items() (*método calibre.db.cache.Cache*), 350
remove_jacket() (*no módulo calibre.ebooks.oeb.polish.jacket*), 358
remove_javascript (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
remove_stale_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
remove_tags (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
remove_tags_after (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
remove_tags_before (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
remove_unused_css() (*no módulo calibre.ebooks.oeb.polish.css*), 359
rename() (*método calibre.ebooks.oeb.polish.container.Container*), 356
rename_files() (*no módulo calibre.ebooks.oeb.polish.replace*), 357
rename_items() (*método calibre.db.cache.Cache*), 351
replace() (*método calibre.ebooks.oeb.polish.container.Container*), 356
replace_links() (*método calibre.ebooks.oeb.polish.container.Container*), 356
replace_links() (*no módulo calibre.ebooks.oeb.polish.replace*), 357
requires_version (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
reset() (*método calibre.devices.interface.DevicePlugin*), 264
reset() (*método calibre.devices.usbms.device.Device*), 272
resolve_internal_links (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
restart_critical (*atributo calibre.gui2.preferences.ConfigWidgetBase*), 281
restart_critical (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 280
restore_book() (*método calibre.db.cache.Cache*), 351
restore_defaults() (*método calibre.gui2.preferences.ConfigWidgetBase*), 281
restore_defaults() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 280
restore_defaults_desc (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 280
restore_original_format() (*método calibre.db.cache.Cache*), 351
reverse_article_order (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
rewind_savepoint() (*método calibre.gui2.tweak_book.boss.Boss*), 363
RSS, **369**
run() (*método calibre.customize.CatalogPlugin*), 255
run() (*método calibre.customize.FileTypePlugin*), 253

S

safe_read_lock (*calibre.db.cache.Cache property*), 351
sanitize_callback() (*método calibre.devices.usbms.device.Device*), 273
sanitize_path_components() (*método calibre.devices.usbms.device.Device*), 273
save_book() (*método calibre.gui2.tweak_book.boss.Boss*), 363
save_original_format() (*método calibre.db.cache.Cache*), 351
save_settings() (*método calibre.customize.Plugin*), 251
save_settings() (*método calibre.ebooks.metadata.sources.base.Source*), 257
save_settings() (*método de classe calibre.devices.interface.DevicePlugin*), 267
scale_news_images (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
scale_news_images_to_device (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
search() (*método calibre.db.cache.Cache*), 351
serialize_item() (*método calibre.ebooks.oeb.polish.container.Container*), 356
set_all_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*),

205
set_conversion_options() (*método calibre.db.cache.Cache*), 351
set_cover() (*método calibre.db.cache.Cache*), 351
set_cover() (*no módulo calibre.ebooks.oeb.polish.cover*), 359
set_driveinfo_name() (*método calibre.devices.interface.DevicePlugin*), 268
set_driveinfo_name() (*método calibre.devices.usbms.driver.USBMS*), 274
set_field() (*método calibre.db.cache.Cache*), 351
set_identifier() (*método calibre.ebooks.metadata.book.base.Metadata*), 204
set_identifiers() (*método calibre.ebooks.metadata.book.base.Metadata*), 204
set_library_info() (*método calibre.devices.interface.DevicePlugin*), 268
set_metadata() (*método calibre.customize.MetadataWriterPlugin*), 255
set_metadata() (*método calibre.db.cache.Cache*), 352
set_modified() (*método calibre.gui2.tweak_book.boss.Boss*), 363
set_option() (*método calibre.devices.interface.DevicePlugin*), 269
set_plugboards() (*método calibre.devices.interface.DevicePlugin*), 267
set_pref() (*método calibre.db.cache.Cache*), 352
set_progress_reporter() (*método calibre.devices.interface.DevicePlugin*), 265
set_progress_reporter() (*método calibre.devices.usbms.device.Device*), 272
set_spine() (*método calibre.ebooks.oeb.polish.container.Container*), 356
set_user_blacklisted_devices() (*método calibre.devices.interface.DevicePlugin*), 268
set_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
settings() (*método de classe calibre.devices.interface.DevicePlugin*), 267
show_current_diff() (*método calibre.gui2.tweak_book.boss.Boss*), 363
show_editor() (*método calibre.gui2.tweak_book.boss.Boss*), 363
shutdown() (*método calibre.devices.interface.DevicePlugin*), 268
shutting_down() (*método calibre.gui2.actions.InterfaceAction*), 279
simultaneous_downloads (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
skip_ad_pages() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
SLOW_DRIVEINFO (*atributo calibre.devices.interface.DevicePlugin*), 263
smart_update() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
sort_index_by() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
Source (*classe em calibre.ebooks.metadata.sources.base*), 256
specialize() (*método calibre.customize.conversion.InputFormatPlugin*), 261
specialize_css_for_output() (*método calibre.customize.conversion.OutputFormatPlugin*), 262
specialize_global_preferences() (*método calibre.devices.interface.DevicePlugin*), 268
specialize_options() (*método calibre.customize.conversion.OutputFormatPlugin*), 262
spine_items (*calibre.ebooks.oeb.polish.container.Container property*), 357
spine_iter (*calibre.ebooks.oeb.polish.container.Container property*), 357
spine_names (*calibre.ebooks.oeb.polish.container.Container property*), 357
split() (*no módulo calibre.ebooks.oeb.polish.split*), 358
split_if_is_multiple_composite() (*método calibre.db.cache.Cache*), 352
split_jobs() (*método calibre.ebooks.metadata.sources.base.Source*), 257
standard_field_keys() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
STANDARD_METADATA_FIELDS (*no módulo calibre.ebooks.metadata.book.base*), 205
start_plugin() (*método calibre.devices.interface.DevicePlugin*), 269
startup() (*método calibre.devices.interface.DevicePlugin*), 268
stop_plugin() (*método calibre.devices.interface.DevicePlugin*), 269
summary_length (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
supported_platforms (*atributo calibre.customize.conversion.InputFormatPlugin*), 259
supported_platforms (*atributo calibre.customize.conversion.OutputFormatPlugin*), 261
supported_platforms (*atributo cali-*

bre.customize.InterfaceActionBase), 279
supported_platforms (*atributo calibre.customize.MetadataReaderPlugin*), 254
supported_platforms (*atributo calibre.customize.MetadataWriterPlugin*), 254
supported_platforms (*atributo calibre.customize.Plugin*), 250
supported_platforms (*atributo calibre.customize.PreferencesPlugin*), 279
supported_platforms (*atributo calibre.devices.usbms.driver.USBMS*), 274
supported_platforms (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
supports_collections() (*método calibre.devices.interface.BookList*), 270
supports_gzip_transfer_encoding (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
supports_restoring_to_defaults (*atributo calibre.gui2.preferences.ConfigWidgetBase*), 281
supports_restoring_to_defaults (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 280
sync_booklists() (*método calibre.devices.interface.DevicePlugin*), 267
sync_booklists() (*método calibre.devices.usbms.driver.USBMS*), 275
sync_preview_to_editor() (*método calibre.gui2.tweak_book.boss.Boss*), 363
synchronize_with_db() (*método calibre.devices.interface.DevicePlugin*), 269

T

tag_browser_context_action() (*método calibre.gui2.actions.InterfaceAction*), 278
tag_to_string() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 42
tags_older_than() (*método calibre.db.cache.Cache*), 352
template_css (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
template_to_attribute() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
temporary_file() (*método calibre.customize.Plugin*), 252
test_fields() (*método calibre.ebooks.metadata.sources.base.Source*), 257
THUMBNAIL_COMPRESSION_QUALITY (*atributo calibre.devices.interface.DevicePlugin*), 263
THUMBNAIL_HEIGHT (*atributo calibre.devices.interface.DevicePlugin*), 263
timefmt (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
timeout (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
title (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
to_html() (*método calibre.ebooks.metadata.book.base.Metadata*), 205
Tool (*classe em calibre.gui2.tweak_book.plugin*), 360
toolbar_button_popup_mode (*atributo calibre.gui2.tweak_book.plugin.Tool*), 361
total_space() (*método calibre.devices.interface.DevicePlugin*), 266
total_space() (*método calibre.devices.usbms.device.Device*), 272
touched_fields (*atributo calibre.ebooks.metadata.sources.base.Source*), 256
type (*atributo calibre.customize.CatalogPlugin*), 255
type (*atributo calibre.customize.conversion.InputFormatPlugin*), 259
type (*atributo calibre.customize.conversion.OutputFormatPlugin*), 261
type (*atributo calibre.customize.FileTypePlugin*), 253
type (*atributo calibre.customize.InterfaceActionBase*), 279
type (*atributo calibre.customize.MetadataReaderPlugin*), 254
type (*atributo calibre.customize.MetadataWriterPlugin*), 255
type (*atributo calibre.customize.Plugin*), 251
type (*atributo calibre.customize.PreferencesPlugin*), 279
type (*atributo calibre.devices.interface.DevicePlugin*), 262
type (*atributo calibre.ebooks.metadata.sources.base.Source*), 256

U

upload_books() (*método calibre.devices.interface.DevicePlugin*), 266
upload_books() (*método calibre.devices.usbms.driver.USBMS*), 274
upload_cover() (*método calibre.devices.usbms.driver.USBMS*), 275
URL, **369**
USBMS (*classe em calibre.devices.usbms.driver*), 273
use_embedded_content (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
user_categories_for_books() (*método calibre.db.cache.Cache*), 352
user_feedback_after_callback (*atributo calibre.devices.interface.DevicePlugin*), 264
UserAnnotation (*atributo calibre.devices.interface.DevicePlugin*), 263

V

VENDOR_ID (*atributo calibre.devices.interface.DevicePlugin*), 262
VENDOR_ID (*atributo calibre.devices.usbms.device.Device*), 271
version (*atributo calibre.customize.MetadataReaderPlugin*), 254
version (*atributo calibre.customize.MetadataWriterPlugin*), 255
version (*atributo calibre.customize.Plugin*), 250
VIRTUAL_BOOK_EXTENSION_MESSAGE (*atributo calibre.devices.interface.DevicePlugin*), 263
VIRTUAL_BOOK_EXTENSIONS (*atributo calibre.devices.interface.DevicePlugin*), 263

W

WANTS_UPDATED_THUMBNAILS (*atributo calibre.devices.interface.DevicePlugin*), 263
web2disk command line option
    --base-dir, 335
    --delay, 335
    --dont-download-stylesheets, 335
    --encoding, 335
    --filter-regexp, 335
    --help, 335
    --match-regexp, 335
    --max-files, 335
    --max-recursions, 335
    --timeout, 335
    --verbose, 335
    --version, 336
    -d, 335
    -h, 335
    -n, 335
    -r, 335
    -t, 335
WINDOWS_CARD_A_MEM (*atributo calibre.devices.usbms.device.Device*), 271
WINDOWS_CARD_B_MEM (*atributo calibre.devices.usbms.device.Device*), 271
WINDOWS_MAIN_MEM (*atributo calibre.devices.usbms.device.Device*), 271
windows_sort_drives() (*método calibre.devices.usbms.device.Device*), 272