Manual do usuário do Calibre

Release 8.7.0

Kovid Goyal

julho 18, 2025

Sumário

1 A Interface Gráfica do Usuário 3

2 Adicionar seus sites de notícia favoritos ao calibre 33

3 Visualizador de livros digitais 53

4 Conversão de E-book 61

5 Editando e-books 81

6 O servidor de Conteúdo do calibre 113

7 Comparando e-books 121

8 Editando metadados de e-books 125

9 Perguntas e Respostas Frequentes 129

10 Tutoriais 153

11 Esquema de URL calibre:// 251

12 Personalizar o calibre 255

13 Interface da Linha de Comando 303

14 Configurando um ambiente de desenvolvimento do calibre 343

15 Gerenciamento de Direitos Digitais (DRM) 373

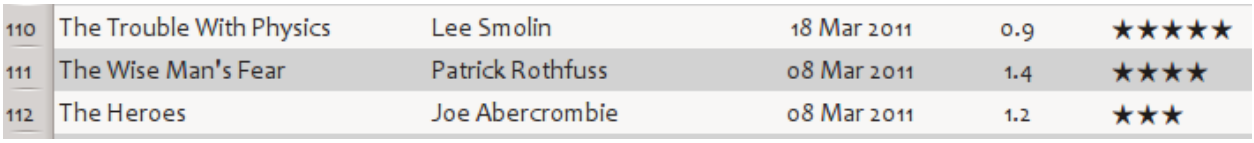
16 Glossário 375

Índice de Módulos Python 377

Índice 379

O calibre é um gerenciador de biblioteca de e-books. Ele pode visualizar, converter e catalogar e-books na maioria dos principais formatos. Ele pode também se comunicar com muitos leitores de e-books. Ele pode se conectar à internet e buscar metadados para seus livros. Ele pode baixar jornais e convertê-los para e-books para uma leitura mais conveniente. Ele é multi plataforma, rodando em Linux, Windows e macOS.

Você acabou de iniciar o calibre. O que fazer agora? Antes que o calibre possa fazer qualquer coisa com seus e-books, ele primeiro precisa saber sobre eles. Arraste e solte alguns arquivos de e-book dentro do calibre, ou clique no botão "Adicionar livros" e procure os e-books com os quais deseje trabalhar. Após tê-los adicionados, eles serão mostrados na janela principal mais ou menos assim:



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 você já admirou a lista de livros que acabou de adicionar aos seus favoritos, você provavelmente gostaria de ler um deles. Para fazer isso, você terá que converter o livro para um formato que seu leitor compreenda. Ao iniciar o calibre pela primeira vez, o *Assistente de Boas-Vindas* é iniciado e irá configurar o calibre para o seu dispositivo de leitura. A conversão é muito simples. Apenas selecione o livro que você deseja converter e então clique no botão "Converter livros". Ignore todas as opções por enquanto e clique em "OK". O pequeno ícone no canto inferior direito irá começar a girar. Quando ele terminar de girar, o seu livro convertidos estará pronto. Clique no botão "Visualizar" para ler o livro.

Se você quiser ler o livro no seu leitor, conecte-o ao computador, espere o calibre detectá-lo (10 a 20 segundos) e então clique no botão "Enviar para o dispositivo". Assim que o ícone parar novamente de girar, desconecte o seu leitor e comece a ler! Se você não converteu o livro no passo anterior, o calibre vai automaticamente convertê-lo para o formato que o seu dispositivo entende.

Para começar o uso mais avançado, você deve ler sobre *A Interface Gráfica do Usuário* (página 3). Para ainda mais potência e versatilidade, aprenda a generated/pt-BR/cli-index. Você também vai achar útil a lista de *Perguntas e Respostas Frequentes* (página 129).

Se você tiver mais perguntas, ou quiser discutir sobre o calibre com outros usuários ou pedir ajuda com coisas específicas, existem fóruns e outros recursos de ajuda disponíveis[1].

Seções

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

CAPÍTULO 1

A Interface Gráfica do Usuário

A Interface Gráfica do Usuário *(GUI)* oferece acesso a todos os recursos de gerenciamento de biblioteca e conversão de formato de e-books. O fluxo de trabalho básico para usar o 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 que estejam no banco de dados, você pode executar várias ações neles que incluem conversão de um formato para outro, transferência para o dispositivo de leitura, visualização no computador e edição de metadados. Esta última inclui a modificação da capa, descrição e tags, entre outros detalhes. Observe que o calibre cria cópias dos arquivos que você adiciona a ele. Seus arquivos originais permanecem intocados.

A interface é dividida em várias seções:

- *Ações* (página 4)
- *Preferências* (página 11)
- *Catálogos* (página 11)
- *Pesquisar e classificar* (página 12)
- *A interface de pesquisa* (página 12)
- *Salvando pesquisas* (página 18)
- *Pesquisando o texto completo de todos os livros* (página 19)
- *Bibliotecas virtuais* (página 19)
- *Marcando livros temporariamente* (página 19)
- *Adivinhando metadados a partir de nomes de arquivo* (página 19)
- *Detalhes do livro* (página 21)
- *Navegador de tags* (página 23)
- *Grade de capas* (página 26)
- *Navegador de capas* (página 27)

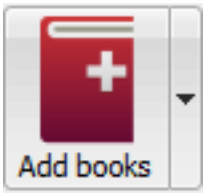
- *Adicionando notas para autores, séries, etc.* (página 28)
- *Visualização rápida* (página 29)
- *Tarefas* (página 30)
- *Atalhos de teclado* (página 30)

1.1 Ações

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

- *Adicionar livros* (página 4)
- *Editar metadados* (página 5)
- *Converter livros* (página 6)
- *Visualizar* (página 6)
- *Enviar para o dispositivo* (página 7)
- *Obter notícias* (página 7)
- *Biblioteca* (página 8)
- *Dispositivo* (página 9)
- *Salvar no disco* (página 9)
- *Conectar/compartilhar* (página 10)
- *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. **Adicionar livros de uma única pasta**: Abre uma caixa de diálogo de seleção de arquivos e permite que você especifique quais livros em uma pasta devem ser adicionados. Esta ação é *sensível ao contexto*, ou seja, depende de qual *catálogo* (página 11) você selecionou. Se você selecionou a *Biblioteca*, os livros serão adicionados à biblioteca. Se você selecionou o dispositivo leitor de e-books, os livros serão carregados no dispositivo e assim por diante.
2. **Adicionar livros de pastas e subpastas**: Permite que você escolha uma pasta. A pasta e todas as suas subpastas são verificadas recursivamente e quaisquer e-books encontrados são adicionados à biblioteca. Você pode escolher

se deseja que o calibre adicione todos os arquivos presentes em uma única pasta a um único registro de livro ou a vários registros de livro. O calibre assume que cada pasta contém um único livro. Assume-se que todos os arquivos de e-book em uma pasta sejam o mesmo livro em formatos diferentes. Esta ação é o inverso da ação *Salvar no disco* (página 9), ou seja, você pode *Salvar no disco*, excluir os livros e adicioná-los novamente no modo de um livro por pasta, sem perda de informações, exceto pela data (isso pressupõe que você não tenha alterado nenhuma das configurações da ação Salvar no disco).

3. **Adicionar vários livros do arquivo (ZIP/RAR)**: Permite que você adicione vários e-books armazenados dentro dos arquivos ZIP ou RAR selecionados. É um atalho conveniente que evita ter que primeiro extrair o arquivo e depois adicionar os livros por meio de uma das duas opções acima.
4. **Adicionar livro vazio (entrada de livro sem formatos)**: Permite que você crie um registro de livro em branco. Isso pode ser usado para então preencher manualmente as informações sobre um livro que você ainda não possui em sua coleção.
5. **Adicionar de ISBN**: Permite que você adicione um ou mais livros através de seus códigos ISBN.
6. **Adicionar arquivos aos registros de livros selecionados**: Permite que você adicione ou atualize os arquivos associados a um livro existente em sua biblioteca.
7. **Adicionar arquivos de dados aos registros de livros selecionados**: Permite adicionar qualquer número de arquivos extras que serão armazenados em um subdiretório `data` no diretório do livro. Consulte *Adding extra data files to a book* (página 127) para obter detalhes.
8. **Adicionar um arquivo vazio aos registros de livros selecionados**: Permite adicionar um arquivo vazio no formato especificado aos registros de livros selecionados.

A ação *Adicionar livros* pode ler metadados de uma ampla variedade de formatos de e-book. Além disso, ele tenta adivinhar os metadados a partir do nome do arquivo. Consulte a seção *Adivinhando metadados a partir de nomes de arquivo* (página 19) para saber como configurar isso.

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

1. Arraste e solte o arquivo no painel Detalhes do livro no lado direito da janela principal
2. Clique com o botão direito do mouse no botão *Adicionar livros* e escolha *Adicionar arquivos aos livros selecionados*.
3. Clique no botão *Adicionar livros* na área superior direita da caixa de diálogo *Editar metadados*, acessada pela ação *Editar metadados* (página 5).

1.1.2 Editar metadados

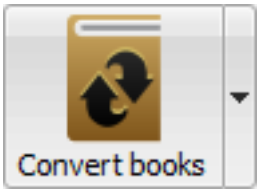
A ação *Editar metadados* tem quatro variações que podem ser acessadas clicando com o botão direito do mouse no botão.

1. **Editar metadados individualmente**: 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 específicos de e-book de um livro.
2. **Editar metadados em massa**: Permite editar campos de metadados comuns para um grande número de livros simultaneamente. Ele opera em todos os livros que você selecionou na *Visualização da biblioteca* (página 12).
3. **Baixar metadados e capas**: Baixa metadados e capas (se disponíveis) para os livros que estão selecionados na lista de livros.

4. **Mesclar registros de livros**: Dá a você a capacidade de mesclar os metadados e formatos de dois ou mais registros de livros. Você pode optar em excluir ou manter os registros que não foram clicados primeiro.
5. **Gerenciar arquivos de dados**: Gerencia os arquivos de dados extras associados aos livros selecionados.

Para mais detalhes, consulte metadados.

1.1.3 Converter livros



E-books podem ser convertidos de vários formatos para qualquer formato que seu leitor de e-books preferir. Muitos e-books disponíveis para compra estarão protegidos pela tecnologia *Gerenciamento de Direitos Digitais* (página 373) *(DRM)*. O calibre não converterá esses e-books. É fácil remover o DRM de muitos formatos, mas como isso pode ser ilegal, você terá que encontrar ferramentas para liberar seus livros por conta própria e entãousar o calibre para convertê-los.

Para a maioria das pessoas, a conversão deve ser um procedimento simples de um clique. Se você quiser saber mais sobre o processo de conversão, consulte conversão.

A ação *Converter livros* possui três variações, acessadas clicando com o botão direito do mouse no botão.

1. **Converter individualmente**: Permite que você especifique opções de conversão para personalizar a conversão de cada e-book selecionado.
2. **Converter em massa**: Permite que você especifique opções apenas uma vez para converter vários e-books em massa.
3. **Criar um catálogo dos livros em sua biblioteca do calibre**: Permite gerar uma lista completa dos livros em sua biblioteca, incluindo todos os metadados, em vários formatos, como XML, CSV, BiBTeX, EPUB e MOBI. O catálogo conterá todos os livros atualmente exibidos na visualização da biblioteca. Isso permite que você use os recursos de pesquisa para limitar os livros a serem catalogados. Além disso, se você selecionar vários livros usando o mouse, apenas esses livros serão adicionados ao catálogo. Se você gerar o catálogo em um formato de e-book como EPUB, MOBI ou AZW3, na próxima vez que você conectar seu leitor de e-books, o catálogo será enviado automaticamente para o dispositivo. Para obter mais informações sobre como os catálogos funcionam, leia *Criar catálogos AZW3 • EPUB • MOBI* (página 244).

1.1.4 Visualizar

A ação *Visualizar* exibe o livro em um programa visualizador de e-books. O calibre possui um visualizador integrado para muitos formatos de e-book. Para outros formatos, ele usa o aplicativo padrão do sistema operacional. Você pode configurar quais formatos devem ser abertos com o visualizador interno por meio de *Preferências → Interface → Comportamento*. Se um livro tiver mais de um formato, você pode visualizar um formato específico clicando com o botão direito do mouse no botão.

1.1.5 Enviar para o dispositivo

A ação *Enviar para o dispositivo* tem oito variações, podendo serem acessadas clicando com o botão direito do mouse no botão.

1. **Enviar para a memória principal**: Os livros selecionados são transferidos para a memória principal do leitor de e-books.
2. **Enviar para o cartão (A)**: Os livros selecionados são transferidos para o cartão de armazenamento (A) no leitor de e-books.
3. **Enviar para o cartão (B)**: Os livros selecionados são transferidos para o cartão de armazenamento (B) no leitor de e-books.
4. **Enviar formato específico para**: Os livros selecionados são transferidos para o local de armazenamento selecionado no dispositivo, no formato especificado por você.
5. **Ejetar dispositivo**: Desconecta o dispositivo do calibre.
6. **Definir ação padrão de envio para o dispositivo**: Permite que você especifique qual das opções, de 1 a 5 acima ou 7 abaixo, será a ação padrão quando você clicar no botão principal.
7. **Enviar e excluir da biblioteca**: Os livros selecionados são transferidos para o local de armazenamento selecionado no dispositivo e então **excluídos** da biblioteca.
8. **Buscar Anotações (experimental)**: Transfere anotações que você possa ter feito em um e-book em seu dispositivo para os metadados de comentários do livro na biblioteca calibre.

Você pode controlar o nome do arquivo e a estrutura de pastas dos arquivos enviados para o dispositivo configurando um modelo em *Preferências → Importar/exportar → Enviar livros para dispositivos*. Consulte também *A linguagem padrão do calibre* (página 160).

1.1.6 Obter notícias

A ação *Baixar notícias* baixa notícias de vários sites da Web e as converte em um e-book que pode ser lido em seu leitor de e-books. Normalmente, o e-book recém-criado é adicionado à sua biblioteca de e-books, mas se um leitor de e-books estiver conectado no momento em que o download for concluído, as notícias também serão carregadas no leitor automaticamente.

A ação *Baixar notícias* usa receitas simples (10-15 linhas de código) para cada site de notícias. Para saber como criar receitas para suas próprias fontes de notícias, consulte notícias.

A ação *Baixar notícias* possui três variações, acessadas clicando com o botão direito do mouse no botão.

1. **Agendar download de notícias**: Permite agendar o download das fontes de notícias selecionadas de uma lista de centenas disponíveis. O agendamento pode ser definido individualmente para cada fonte de notícias que você selecionar e o agendamento é flexível, permitindo que você selecione dias específicos da semana ou uma frequência de dias entre os downloads.

2. **Adicionar uma fonte de notícias personalizada**: Permite que você crie uma receita simples para baixar notícias de um site de notícias personalizado que você deseja acessar. Criar a receita pode ser tão simples quanto especificar um URL de feed de notícias RSS, ou você pode ser mais prescritivo criando um código baseado em Python para a tarefa. Para obter mais informações, consulte notícias.
3. **Baixar todas as fontes de notícias agendadas**: Faz com que o calibre comece imediatamente a baixar todas as fontes de notícias que você agendou.

1.1.7 Biblioteca

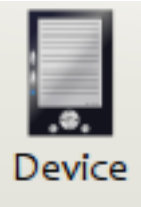
A ação *Biblioteca* permite criar, alternar entre, renomear ou remover uma biblioteca. O calibre permite que você crie quantas bibliotecas desejar. Você pode, por exemplo, criar uma biblioteca de ficção, uma biblioteca de não-ficção, uma biblioteca de língua estrangeira, uma biblioteca de projetos ou qualquer estrutura que atenda às suas necessidades. As bibliotecas são a mais alta estrutura organizacional dentro do calibre. Cada biblioteca possui seu próprio conjunto de livros, tags, categorias e local de armazenamento base.

1. **Alternar/criar biblioteca…**: Permite que você; a) conecte-se a uma biblioteca do calibre pré-existente em outro local, b) crie uma biblioteca vazia em um novo local ou, c) mova a biblioteca atual para um local recém-especificado.
2. **Troca rápida**: Permite alternar entre bibliotecas que foram registradas ou criadas no calibre.
3. **Renomear biblioteca**: Permite renomear uma biblioteca.
4. **Escolher um livro aleatório**: Escolhe um livro aleatório na biblioteca para você
5. **Remover biblioteca**: Permite que você cancele o registro de uma biblioteca do calibre.
6. **Exportar/importar todos os dados do calibre**: Permite exportar dados do calibre para migrar para um novo computador ou importar dados exportados anteriormente.
7. **<nome da biblioteca>**: As ações 7, 8 etc… fornecem acesso de troca imediata entre várias bibliotecas que você criou ou conectou. Esta lista contém apenas as 5 bibliotecas usadas com mais frequência. Para a lista completa, use o menu Troca rápida.
8. **Manutenção da biblioteca**: Permite verificar a biblioteca atual em busca de problemas de consistência de dados e restaurar o banco de dados da biblioteca atual a partir de backups.

> **Nota**
>
> Os metadados sobre seus e-books, por exemplo, título, autor e tags, são armazenados em um único arquivo na pasta da biblioteca calibre chamado metadata.db. Se este arquivo for corrompido (um evento muito raro), você pode perder os metadados. Felizmente, o calibre faz backup automaticamente dos metadados de cada livro individual na pasta do livro como um arquivo OPF. Usando a ação Restaurar banco de dados em Manutenção da biblioteca descrita acima, você pode fazer com que o calibre reconstrua o arquivo metadata.db a partir dos arquivos OPF individuais para você.

Você pode copiar ou mover livros entre bibliotecas diferentes (depois de configurar mais de uma biblioteca) clicando com o botão direito do mouse no livro e selecionando a ação *Copiar para a biblioteca*.

1.1.8 Dispositivo



A ação *Dispositivo* permite que você visualize os livros na memória principal ou nos cartões de armazenamento do seu dispositivo, ou para ejetar o dispositivo (desconectá-lo do calibre). Este ícone aparece automaticamente na barra de ferramentas principal do calibre quando você conecta um dispositivo compatível. Você pode clicar nele para ver os livros em seu dispositivo. Você também pode arrastar e soltar livros da sua biblioteca calibre no ícone para transferi-los para o seu dispositivo. Por outro lado, você pode arrastar e soltar livros do seu dispositivo no ícone da biblioteca na barra de ferramentas para transferir livros do seu dispositivo para a biblioteca calibre.

1.1.9 Salvar no disco

A ação *Salvar no disco* tem cinco variações, acessadas clicando com o botão direito do mouse no botão.

1. **Salvar no disco**: Salva os livros selecionados no disco organizados em pastas. A estrutura da pasta é semelhante a:

```
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 *Preferências → Importar/exportar → Salvar livros no disco*. Consulte também *A linguagem padrão do calibre* (página 160).

2. **Salvar no disco em uma única pasta**: Salva os livros selecionados no disco em uma única pasta.

Para 1. e 2., todos os formatos disponíveis, bem como os metadados, são armazenados no disco para cada livro selecionado. Os metadados são armazenados em um arquivo OPF. Os livros salvos podem ser reimportados para a biblioteca sem qualquer perda de informação usando a ação *Adicionar livros* (página 4).

3. **Salvar apenas o formato *<seu preferido>* no disco**: Salva os livros selecionados no disco na estrutura de pastas mostrada em (1.), mas apenas no formato de e-book de sua preferência. Você pode definir seu formato preferido em *Preferências → Interface → Comportamento → Formato de saída preferido*

4. **Salvar apenas o formato *<seu preferido>* no disco em uma única pasta**: Salva os livros selecionados no disco em uma única pasta, mas apenas no formato de e-book de sua preferência. Você pode definir seu formato preferido em *Preferências → Interface → Comportamento → Formato de saída preferido*

5. **Salvar formato único no disco…**: Salva os livros selecionados no disco na estrutura de pastas mostrada em (1.), mas apenas no formato que você selecionar na lista pop-up.

1.1.10 Conectar/compartilhar

A ação *Conectar/compartilhar* permite que você se conecte manualmente a um dispositivo ou pasta em seu computador. Também permite configurar sua biblioteca calibre para acesso por meio de um navegador da Web ou e-mail.

A ação *Conectar/compartilhar* tem quatro variações, acessadas clicando com o botão direito do mouse no botão.

1. **Conectar à pasta**: Permite que você se conecte a qualquer pasta em seu computador como se fosse um dispositivo e use todos os recursos que o calibre tem para dispositivos com essa pasta. Útil se o seu dispositivo não puder ser suportado pelo calibre, mas estiver disponível como um disco USB.
2. **Iniciar servidor de conteúdo**: Inicia o servidor da Web integrado do calibre. Quando iniciado, sua biblioteca calibre estará acessível por meio de um navegador da Web a partir da Internet (se você escolher). Você pode configurar como o servidor da Web é acessado definindo as preferências em *Preferências → Compartilhamento → Compartilhamento pela rede*
3. **Configurar compartilhamento de livros por e-mail**: Permite o compartilhamento de livros e feeds de notícias por e-mail. Depois de configurar os endereços de e-mail para esta opção, o calibre enviará atualizações de notícias e livros para os endereços de e-mail inseridos. Você pode configurar como o calibre envia e-mail definindo as preferências em *Preferências → Compartilhamento → Compartilhamento de livros por e-mail*. Depois de configurar um ou mais endereços de e-mail, esta entrada de menu será substituída por entradas de menu para enviar livros para os endereços de e-mail configurados.

1.1.11 Remover livros

A ação *Remover livros* **exclui livros permanentemente**, portanto, use-a com cuidado. É *sensível ao contexto*, ou seja, depende de qual *catálogo* (página 11) você selecionou. Se você selecionou a *Biblioteca*, os livros serão removidos da biblioteca. Se você selecionou o dispositivo leitor de e-books, os livros serão removidos do dispositivo. Para remover apenas um formato específico para um determinado livro, use a ação *Editar metadados* (página 5). Remover livros também tem cinco variações que podem ser acessadas clicando com o botão direito do mouse no botão.

1. **Remover livros selecionados**: Permite que você **exclua permanentemente** todos os livros que estão selecionados na lista de livros.
2. **Remover arquivos de um formato específico de livros selecionados…**: Permite que você **exclua permanentemente** arquivos de e-book de um formato especificado dos livros que estão selecionados na lista de livros.
3. **Remover todos os formatos de livros selecionados, exceto…**: Permite que você **exclua permanentemente** arquivos de e-book de qualquer formato, exceto um formato especificado, dos livros que estão selecionados na lista de livros.
4. **Remover todos os formatos de livros selecionados**: Permite que você **exclua permanentemente** todos os arquivos de e-book dos livros que estão selecionados na lista de livros. Apenas os metadados serão mantidos.

5. **Remover capas de livros selecionados**: Permite que você **exclua permanentemente** os arquivos de imagem de capa dos livros que estão selecionados na lista de livros.
6. **Remover livros correspondentes do dispositivo**: Permite que você remova arquivos de e-book de um dispositivo conectado que correspondam aos livros que estão selecionados na lista de livros.
7. **Restaurar excluídos recentemente**: Permite desfazer a remoção de livros ou formatos.

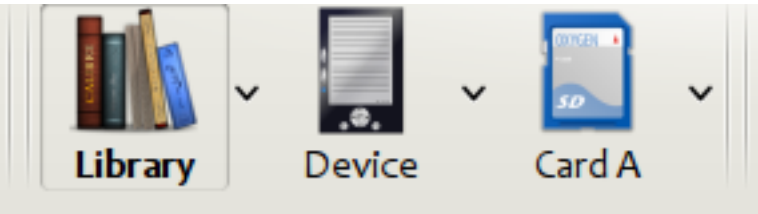
> **Nota**
>
> Observe que quando você usa *Remover livros* para excluir livros da sua biblioteca calibre, o registro do livro é excluído, mas os livros são armazenados temporariamente, por alguns dias, em uma pasta de lixo. Você pode desfazer a exclusão clicando com o botão direito do mouse no botão *Remover livros* e escolhendo *Restaurar excluídos recentemente*.

1.2 Preferências

A ação *Preferências* permite que você altere a maneira como vários aspectos do calibre funcionam. Possui quatro variações, acessadas clicando com o botão direito do mouse no botão.

1. **Preferências**: Permite que você altere a maneira como vários aspectos do calibre funcionam. Clicar no botão também executa esta ação.
2. **Executar assistente de boas-vindas**: Permite que você inicie o *Assistente de boas-vindas* que apareceu na primeira vez que você iniciou o calibre.
3. **Obter plugins para aprimorar o calibre**: Abre uma nova janela que mostra os plugins para o calibre. Esses plugins são desenvolvidos por terceiros para estender a funcionalidade do calibre.
4. **Reiniciar em modo de depuração**: Permite que você ative um modo de depuração que pode ajudar os desenvolvedores do calibre a resolver problemas encontrados no programa. Para a maioria dos usuários, isso deve permanecer desativado, a menos que seja instruído por um desenvolvedor a ativá-lo.

1.3 Catálogos



Um *catálogo* é uma coleção de livros. O calibre pode gerenciar dois tipos diferentes de catálogos:

1. **Biblioteca**: Esta é uma coleção de livros armazenados em sua biblioteca calibre em seu computador.
2. **Dispositivo**: Esta é uma coleção de livros armazenados em seu leitor de e-books. Ela estará disponível quando você conectar o leitor ao seu computador.

Muitas operações, como adicionar livros, excluir, visualizar, etc., são sensíveis ao contexto. Então, por exemplo, se você clicar no botão *Visualizar* quando tiver o catálogo **Dispositivo** selecionado, o calibre abrirá os arquivos no dispositivo para visualização. Se você tiver o catálogo **Biblioteca** selecionado, os arquivos em sua biblioteca calibre serão abertos.

1.4 Pesquisar e classificar

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

- Você pode classificá-los por título, autor, data, classificação, etc. clicando nos títulos das colunas. Você também pode classificar subitens, ou seja, classificar em várias colunas. Por exemplo, se você clicar na coluna de título e depois na coluna de autor, o livro será classificado por autor e, em seguida, todas as entradas para o mesmo autor serão classificadas por título.
- Você pode pesquisar um livro específico ou um conjunto de livros usando a barra de pesquisa. Mais sobre isso abaixo.
- Você pode editar metadados de forma rápida e conveniente selecionando a entrada que deseja alterar na lista e pressionando a tecla `E`.
- Você pode realizar ações em conjuntos de livros. Para selecionar vários livros, você pode:
  - Mantenha a tecla `Ctrl` pressionada e clique nos livros que deseja selecionar.
  - Mantenha a tecla `Shift` pressionada e clique no livro inicial e final de um intervalo de livros que deseja selecionar.
- Você pode configurar quais campos deseja exibir usando a caixa de diálogo configuração.
- Para executar a classificação secundária complexa baseada em várias colunas, adicione a ferramenta *Classificar por* a uma barra de ferramentas por meio de *Preferências → Barras de ferramentas e menus*.

1.5 A interface de pesquisa

Você pode pesquisar todos os metadados do livro inserindo termos de pesquisa na barra de pesquisa. Por exemplo:

```
Asimov Foundation format:lrf
```

Isso corresponderá a todos os livros em sua biblioteca que possuem `Asimov` e `Foundation` 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
```

Tipos de pesquisa

Existem quatro tipos de pesquisa: *contém*, *igualdade*, *expressão regular* (consulte expressões regulares[2]) e *variante de caractere*. Você escolhe o tipo de pesquisa com um caractere de prefixo.

Pesquisas 'Contém'

As pesquisas sem caractere de prefixo são *contém* e diferenciam maiúsculas de minúsculas por padrão. Um item corresponde se a string de pesquisa aparecer em qualquer lugar nos metadados indicados. Você pode tornar as pesquisas *contém* sensíveis a maiúsculas e minúsculas marcando a opção *Pesquisa sensível a maiúsculas e minúsculas* em *Preferências / Pesquisa*. Se a opção de pesquisa *Caracteres sem acento correspondem a caracteres acentuados e a pontuação é ignorada* estiver marcada, um caractere corresponderá a todas as suas variantes (por exemplo, *e* corresponde a *é*, *è*, *ê* e *ë*) e toda a pontuação e espaços em branco serão ignorados. Por exemplo, se a opção *Caracteres sem acento correspondem…* estiver marcada, então, dados os dois títulos de livros:

1. Grande, Incomodado e Mau
2. Grande Chatice

então essas pesquisas encontram:

- `title:"er"` corresponde a ambos ('e' corresponde a 'é' e 'e').
- `title:"g "` corresponde a ambos porque os espaços são ignorados.
- `title:"g,"` corresponde a ambos porque a vírgula é ignorada.
- `title:"gb"` corresponde a ambos porque ', ' é ignorado no livro 1 e os espaços são ignorados no livro 2.
- `title:"g b"` corresponde a ambos porque a vírgula e o espaço são ignorados.
- `title:"db"` corresponde a #1 porque o espaço em 'e Mau' é ignorado.
- `title:","` corresponde a ambos (na verdade, corresponde a todos os livros) porque as vírgulas são ignoradas.

Se a opção *Caracteres sem acento correspondem…* não estiver marcada, as variantes de caracteres, pontuação e espaçamento serão significativas.

Você pode definir apenas uma das opções de pesquisa *Pesquisa sensível a maiúsculas e minúsculas* e *Caracteres sem acento correspondem a caracteres acentuados e a pontuação é ignorada*.

Pesquisas de 'Igualdade'

As pesquisas de igualdade são indicadas prefixando a string de pesquisa com um sinal de igual (=). Por exemplo, a consulta `tag:"=science"` corresponderá a *science*, mas não a *science fiction* ou *hard science*. As variantes de caracteres são significativas: *é* não corresponde a *e*.

Duas variantes de pesquisas de igualdade são usadas para itens hierárquicos (por exemplo, A.B.C): pesquisas de prefixo hierárquico e pesquisas de componente hierárquico. A primeira, indicada por um único ponto após o sinal de igual (`=.`), corresponde às partes iniciais de um item hierárquico. A segunda, indicada por dois pontos após o sinal de igual (`=..`), corresponde a um nome interno no item hierárquico. Exemplos, usando a tag `História.Militar.WWII` como o valor:

- `tags:"=.História"` : Verdadeiro. `História` é um prefixo da tag.
- `tags:"=.História.Militar"` : Verdadeiro. `História.Militar` é um prefixo da tag.

[2] https://pt.wikipedia.org/wiki/Express%C3%A3o_regular

- `tags:"=.História.Militar.WWII"` : Verdadeiro. `História.Militar.WWII` é um prefixo da tag, embora impróprio.
- `tags:"=.Militar"` : Falso. `Militar` não é um prefixo da tag.
- `tags:"=.WWII"` : Falso. `WWII` não é um prefixo da tag.
- `tags:"=..História"` : Verdadeiro. A hierarquia contém o valor `História`.
- `tags:"=..Militar"` : Verdadeiro. A hierarquia contém o valor `Militar`.
- `tags:"=..WWII"` : Verdadeiro. A hierarquia contém o valor `WWII`.
- `tags:"=..Military.WWII"` : Falso. A pesquisa `..` procura valores únicos.

Pesquisas de 'Expressão regular'

As pesquisas de expressão regular são indicadas prefixando a string de pesquisa com um til (~). Qualquer expressão regular compatível com Python[3] pode ser usada. As barras invertidas usadas para escapar de caracteres especiais em expressões regulares devem ser duplicadas porque as barras invertidas simples serão removidas durante a análise da consulta. Por exemplo, para corresponder a um parêntese literal, você deve inserir `\\(` ou, alternativamente, usar *super-aspas* (veja abaixo). As pesquisas de expressão regular são pesquisas 'contém', a menos que a expressão seja ancorada. As variantes de caracteres são significativas: `~e` não corresponde a `é`.

Pesquisas de 'Variante de caractere'

As pesquisas de variante de caractere são indicadas prefixando a string de pesquisa com um acento circunflexo (^). Esta pesquisa é semelhante à pesquisa 'contém' (acima), exceto que:

- maiúsculas e minúsculas são sempre ignoradas.
- as variantes de caracteres sempre correspondem entre si.
- pontuação e espaços em branco são sempre significativos.

As opções de pesquisa *Caracteres sem acento correspondem a caracteres acentuados e a pontuação é ignorada* e *Pesquisa sensível a maiúsculas e minúsculas* são ignoradas. Elas não têm efeito sobre o comportamento desta pesquisa.

O seguinte compara esta pesquisa a uma pesquisa contém, assumindo que a opção *Caracteres sem acento correspondem…* esteja marcada (veja acima), dados os mesmos dois títulos de livros:

1. Grande, Incomodado e Mau
2. Grande Chatice

então essas pesquisas de variante de caractere encontram:

- `title:"^er"` corresponde a ambos ('e' corresponde a 'é' e 'e')
- `title:"^g"` corresponde a ambos
- `title:"^g "` corresponde a #2 porque o espaço é significativo
- `title:"^g,"` corresponde a #1 porque a vírgula é significativa
- `title:"^gb"` não corresponde a nada porque o espaço e a vírgula são significativos
- `title:"^g b"` corresponde a #2 porque a vírgula é significativa
- `title:"^db"` não corresponde a nada
- `title:"^,"` corresponde a #1 (em vez de todos os livros) porque a vírgula é significativa

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

Sintaxe da expressão de pesquisa

Uma *expressão de pesquisa* é uma sequência de *termos de pesquisa* opcionalmente separados pelos operadores `and` e `or`. Se dois termos de pesquisa ocorrerem sem um operador de separação, `and` será assumido. O operador `and` tem prioridade sobre o operador `or`; por exemplo, a expressão ``a or b and c é o mesmo que `a or (b and c)`. Você pode usar parênteses para alterar a prioridade; por exemplo `(a or b) and c` para fazer o `or` ser avaliado antes do `and`. Você pode usar o operador `not` para negar (inverter) o resultado da avaliação de uma expressão de pesquisa. Exemplos:

- `not tag:foo` encontra todos os livros que não contêm a tag `foo`
- `not (author:Asimov or author:Weber)` encontra todos os livros não escritos por Asimov ou Weber.

Os exemplos acima mostram exemplos de *termos de pesquisa*. Um *termo de pesquisa* básico é uma sequência de caracteres que não inclui espaços, aspas (`"`), barras invertidas (`\`) ou parênteses (`( )`). Pode ser opcionalmente precedido por um especificador de nome de coluna: o *nome de pesquisa* de uma coluna seguido por dois pontos (:), por exemplo `author:Asimov`. Se um termo de pesquisa deve conter um espaço, o termo inteiro deve ser colocado entre aspas, como em `title:"O Anel"`. Se o termo de pesquisa deve conter aspas, elas devem ser 'escapadas' com barras invertidas. Por exemplo, para pesquisar uma série chamada *A "Bola"* e *A "Cadeia"*, use:

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

Se você precisar de uma barra invertida real, algo que acontece frequentemente em pesquisas de *expressão regular*, use duas delas (`\\`).

Às vezes é difícil acertar todos os escapes para que o resultado seja o que você deseja, especialmente em pesquisas de *expressão regular* e *modelo*. Nesses casos, use a *super-aspas*: `"""sequência de caracteres"""`. Os caracteres entre super-aspas são usados sem alterações: nenhum processamento de escape é feito.

Mais Informações

Para pesquisar uma string que começa com um sinal de igual, til ou acento circunflexo; prefixe a string com uma barra invertida.

Coloque as strings de pesquisa entre aspas (") se a string contiver parênteses ou espaços. Por exemplo, para encontrar livros com a tag `Ficção Científica`, você deve pesquisar por `tag:"=ficção científica"`. Se você pesquisar por `tag:=ficção científica`, encontrará todos os livros com a tag `ciência` e a palavra `ficção` em qualquer metadado.

Você pode criar consultas de pesquisa avançada facilmente usando a caixa de diálogo *Pesquisa avançada* acessada clicando no botão à esquerda da caixa de pesquisa.

Os campos disponíveis para pesquisa são: `tag`, `title`, `author`, `publisher`, `series`, `series_index`, `rating`, `cover`, `comments`, `format`, `identifiers`, `date`, `pubdate`, `search`, `size`, `vl` e colunas personalizadas. Se um dispositivo estiver conectado, o campo `ondevice` ficará disponível ao pesquisar na visualização da biblioteca calibre. Para encontrar o nome de pesquisa (na verdade chamado de *nome de pesquisa*) para uma coluna personalizada, passe o mouse sobre o cabeçalho da coluna na visualização da biblioteca.

Datas

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
```

Se a data for ambígua, a localidade atual será usada para comparação de datas. Por exemplo, em uma localidade mm/dd/aaaa, 02/01/2009 é interpretado como 01 de fevereiro de 2009. Em uma localidade dd/mm/aaaa, é interpretado como 02 de janeiro de 2009. Algumas strings de data especiais estão disponíveis. A string `today` se traduz na data de hoje, seja ela qual for. As strings `yesterday` e `thismonth` (ou o equivalente traduzido no idioma atual) também

funcionam. Além disso, a string `daysago` (também traduzida) pode ser usada para comparar com uma data de alguns dias atrás. Por exemplo:

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

Para evitar possíveis problemas com strings traduzidas ao usar uma versão não inglesa do calibre, as strings `_today`, `_yesterday`, `_thismonth` e `_daysago` estão sempre disponíveis. Elas não são traduzidas.

Pesquisando datas e valores numéricos com comparações relacionais

Datas e campos numéricos suportam os operadores relacionais = (igual a), > (maior que), >= (maior ou igual a), < (menor que), <= (menor ou igual a) e `!=` (diferente de). Os campos de classificação são considerados numéricos. Por exemplo, a pesquisa `rating:>=3` encontrará todos os livros classificados com 3 ou mais.

Você pode pesquisar livros que têm um formato de um determinado tamanho assim:

- `size:>1.1M` encontrará livros com um formato maior que 1,1 MB
- `size:<=1K` encontrará livros com um formato menor ou igual a 1 KB

Você pode pesquisar o número de itens em campos de vários valores, como tags, usando o caractere `#` e, em seguida, usando a mesma sintaxe dos campos numéricos. Por exemplo, para encontrar todos os livros com mais de 4 tags, use `tags:#>4`. Para encontrar todos os livros com exatamente 10 tags, use `tags:#=10`.

Índices de séries

Os índices de séries são pesquisáveis. Para a série padrão, o nome de pesquisa é `series_index`. Para colunas de séries personalizadas, use o nome de pesquisa da coluna seguido por _index. Por exemplo, para pesquisar os índices de uma coluna de série personalizada chamada `#my_series`, você usaria o nome de pesquisa `#my_series_index`. Os índices de séries são números, então você pode usar os operadores relacionais descritos acima.

Pesquisas salvas

O campo especial `search` é usado para *pesquisas salvas* (página 18). Se você salvar uma pesquisa com o nome `Livros do meu cônjuge`, poderá inserir `search:"Livros do meu cônjuge"` na barra de pesquisa para reutilizar a pesquisa salva. Mais sobre como salvar pesquisas abaixo.

Bibliotecas virtuais

O campo especial `vl` é usado para pesquisar livros em uma biblioteca virtual. Por exemplo, `vl:Lidos` encontrará todos os livros na biblioteca virtual *Lidos*. A pesquisa `vl:Lidos and vl:"Ficção Científica"` encontrará todos os livros que estão nas bibliotecas virtuais *Lidos* e *Ficção Científica*. O valor seguindo `vl:` deve ser o nome de uma biblioteca virtual. Se o nome da biblioteca virtual contiver espaços, coloque-o entre aspas.

Se um campo tem um valor

Você pode pesquisar a ausência ou presença de um valor para um campo usando "true" e "false". Por exemplo:

- `cover:false` encontra todos os livros sem capa
- `series:true` encontra todos os livros que estão em uma série
- `series:false` encontra todos os livros que não estão em uma série
- `comments:false` encontra todos os livros com um comentário vazio
- `formats:false` encontra todos os livros sem arquivos de livro (registros vazios)

Colunas personalizadas Sim/não

Pesquisar colunas personalizadas Sim/não para `false`, `empty` ou `blank` encontrará todos os livros com valores indefinidos na coluna. Pesquisar por `true` encontrará todos os livros que não têm valores indefinidos na coluna. Pesquisar por `yes` ou `checked` encontrará todos os livros com `Sim` na coluna. Pesquisar por `no` ou `unchecked` encontrará todos os livros com `Não` na coluna. Observe que as palavras `yes`, `no`, `blank`, `empty`, `checked` e `unchecked` são traduzidas;

você pode usar a palavra equivalente do idioma atual ou a palavra em inglês. As palavras `true` e `false` e os valores especiais `_yes`, `_no` e `_empty` não são traduzidos.

Identificadores

Identificadores (por exemplo, ISBN, DOI, LCCN, etc.) usam uma sintaxe estendida. Um identificador tem o formato `tipo:valor`, como em `isbn:123456789`. A sintaxe estendida permite que você especifique independentemente o tipo e o valor a serem pesquisados. Tanto o tipo quanto as partes de valor da consulta podem usar qualquer um dos *tipos de pesquisa* (página 13). Exemplos:

- `identifiers:true` encontrará livros com qualquer identificador.
- `identifiers:false` encontrará livros sem identificador.
- `identifiers:123` pesquisará livros com qualquer tipo que tenha um valor contendo *123*.
- `identifiers:=123456789` pesquisará livros com qualquer tipo que tenha um valor igual a *123456789*.
- `identifiers:=isbn:` e `identifiers:isbn:true` encontrarão livros com um tipo igual a ISBN com qualquer valor
- `identifiers:=isbn:false` encontrará livros sem tipo igual a ISBN.
- `identifiers:=isbn:123` encontrará livros com um tipo igual a ISBN com um valor contendo *123*.
- `identifiers:=isbn:=123456789` encontrará livros com um tipo igual a ISBN com um valor igual a *123456789*.
- `identifiers:i:1` encontrará livros com um tipo contendo um *i* com um valor contendo um *1*.

Categorias visíveis no navegador de tags

A pesquisa `in_tag_browser:true` encontra todos os livros que estão em categorias (tags, autores, etc.) atualmente exibidas no *Navegador de tags*. Isso é útil se você definir as duas preferências *Preferências → Aparência → Navegador de tags → Ocultar categorias vazias* e *Localizar mostra todos os itens correspondentes*. Com essas duas preferências definidas, fazer um *Localizar* no *Navegador de tags* mostra apenas as categorias que contêm itens correspondentes ao *Localizar*. Então, a pesquisa `in_tag_browser:true` encontra adicionalmente livros com essas categorias/itens. Você pode executar esta pesquisa facilmente pressionando a tecla `Ctrl+Alt+Shift+F` ou clicando no botão configurar no *Navegador de tags* e escolhendo a entrada *Mostrar apenas livros que têm categorias visíveis*.

Pesquisar usando modelos

Você pode pesquisar usando um modelo em *A linguagem padrão do calibre* (página 160) em vez de um campo de metadados. Para fazer isso, você insere um modelo, um tipo de pesquisa e o valor a ser pesquisado. A sintaxe é:

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

O `modelo` é qualquer modelo de linguagem de modelo calibre válido. O `tipo de pesquisa` deve ser um de `t` (pesquisa de texto), `d` (pesquisa de data), `n` (pesquisa numérica) ou `b` (definido/não definido (booleano)). O `valor` é o que você quiser e pode usar os *tipos de pesquisa* (página 13) descritos acima para os vários tipos de pesquisa. Você deve colocar a string de pesquisa inteira entre aspas se houver espaços em qualquer lugar dela.

Exemplos:

- `template:"program: connected_device_name('main')#@#:t:kindle"` – é verdadeiro quando o dispositivo `kindle` está conectado.
- `template:"program: select(formats_sizes(), 'EPUB')#@#:n:>1000000"` – encontra livros com arquivos EPUB maiores que 1 MB.
- `template:"program: select(formats_modtimes('iso'), 'EPUB')#@#:d:>10daysago"` – encontra livros com arquivos EPUB mais recentes que 10 dias atrás.

- `template:"""program: book_count('tags:^"' & $series & '"', 0) != 0#@#:n:1"""` – encontra todos os livros que contêm o nome da série nas tags. Este exemplo usa super-aspas porque o modelo usa aspas simples (`'`) e aspas duplas (`"`) ao construir a expressão de pesquisa.

Você pode criar consultas de pesquisa de modelo facilmente usando a caixa de diálogo *Pesquisa avançada* acessada clicando no botão . Você pode testar modelos em livros específicos usando o *Testador de modelo* do calibre, que pode ser adicionado às barras de ferramentas ou menus por meio de *Preferências → Barras de ferramentas e menus*. Ele também pode ser atribuído a um atalho de teclado por meio de *Preferências → Atalhos*.

Caixa de diálogo de pesquisa avançada

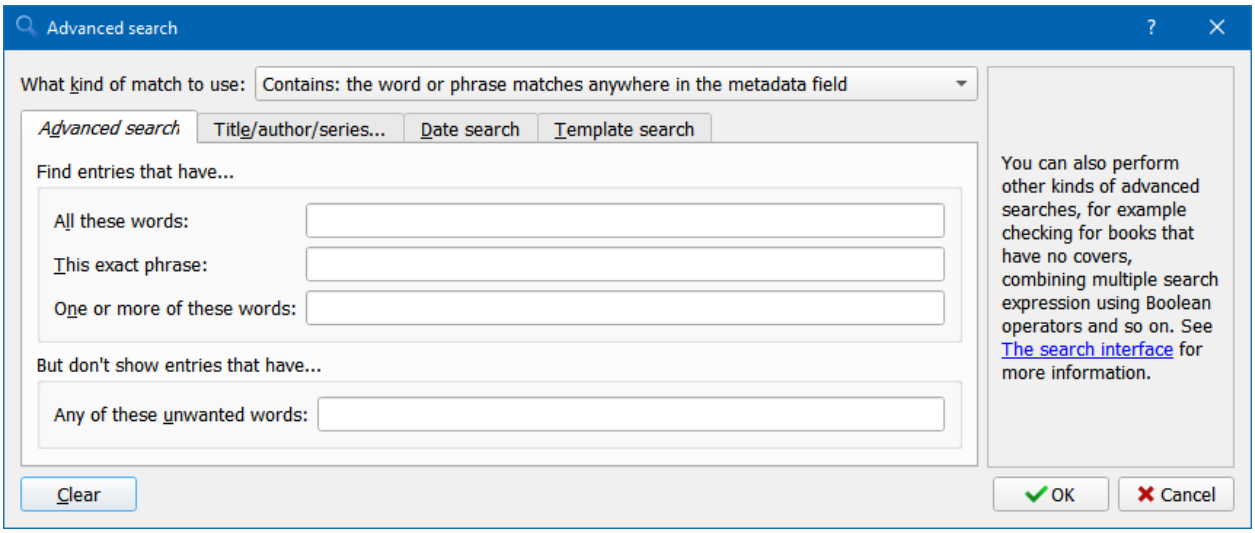


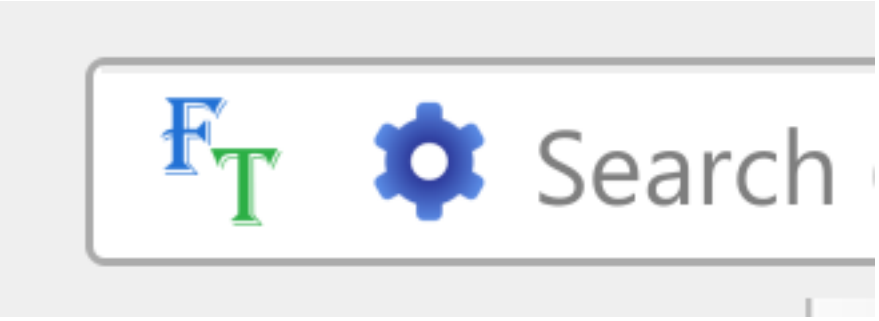
Figura1: *Caixa de diálogo de pesquisa avançada*

1.6 Salvando pesquisas

o calibre permite que você salve uma pesquisa usada com frequência com um nome especial e, em seguida, reutilize essa pesquisa com um único clique. Para fazer isso, crie sua pesquisa digitando-a na barra de pesquisa ou usando o navegador de tags. Em seguida, digite o nome que deseja dar à pesquisa na caixa Pesquisas salvas ao lado da barra de pesquisa. Clique no ícone de mais ao lado da caixa de pesquisas salvas para salvar a pesquisa.

Agora você pode acessar sua pesquisa salva no navegador de tags em *Pesquisas salvas*. Um único clique permitirá que você reutilize qualquer pesquisa arbitrariamente complexa facilmente, sem precisar recriá-la.

1.7 Pesquisando o texto completo de todos os livros



O calibre pode *opcionalmente* indexar o texto completo dos livros na biblioteca para permitir a pesquisa instantânea de palavras dentro de qualquer livro. Para usar esta funcionalidade, clique no botão *FT* na extremidade esquerda da barra de pesquisa.

Em seguida, ative a indexação para a biblioteca atual. Assim que a indexação for concluída, você poderá pesquisar todo o texto na biblioteca completa. Quando você adiciona novos livros à biblioteca, eles serão indexados automaticamente em segundo plano. Esta pesquisa permite que você encontre rapidamente todos os livros que contêm uma palavra ou combinação de palavras. Você pode até pesquisar palavras que ocorrem perto de outras palavras, conforme mostrado nos exemplos na janela pop-up de pesquisa. Observe que esta ferramenta de pesquisa encontrará apenas uma ocorrência da consulta de pesquisa em um livro específico, não listará todas as ocorrências, para isso é melhor pesquisar dentro do livro usando o *Visualizador de e-books* do calibre.

Você pode reindexar um livro individual clicando com o botão direito do mouse no *Painel de detalhes do livro* no calibre e escolhendo *Reindexar este livro para pesquisa de texto completo*.

1.8 Bibliotecas virtuais

Uma *biblioteca virtual* é uma maneira de fingir que sua biblioteca calibre tem apenas alguns livros em vez de sua coleção completa. Esta é uma excelente maneira de particionar sua grande coleção de livros em partes menores e gerenciáveis. Para aprender como criar e usar bibliotecas virtuais, consulte o tutorial: *Bibliotecas virtuais* (página 248).

1.9 Marcando livros temporariamente

Você pode marcar conjuntos arbitrários de livros temporariamente. Os livros marcados terão um alfinete sobre eles e podem ser encontrados com a pesquisa `marked:true`. Para marcar um livro, pressione `Ctrl+M` ou vá para *Preferências → Barras de ferramentas e menus* e adicione o botão *Marcar livros* à barra de ferramentas principal.

Você pode marcar livros com um rótulo de texto específico clicando com o botão direito do mouse no botão *Marcar livros* e escolhendo *Marcar livros com rótulo de texto*. Os livros marcados com rótulos de texto podem ser encontrados posteriormente usando a pesquisa `marked:"=o-texto-que-você-inseriu"`.

1.10 Adivinhando metadados a partir de nomes de arquivo

Normalmente, o calibre lê os metadados de dentro do arquivo do livro. No entanto, ele pode ser configurado para ler os metadados do nome do arquivo, por meio de *Preferências → Importar/exportar → Adicionar livros → Ler metadados do conteúdo do arquivo*.

Você também pode controlar como os metadados são lidos do nome do arquivo usando expressões regulares (consulte *Tudo sobre como usar expressões regulares no calibre* (página 215)). Na seção *Adicionando livros* da caixa de diálogo de configuração, você pode especificar uma expressão regular que o calibre usará para tentar adivinhar os metadados a partir dos nomes dos arquivos de e-book que você adiciona à biblioteca. A expressão regular padrão é:

```
title - author
```

ou seja, assume que todos os caracteres até o primeiro – são o título do livro e os caracteres subsequentes são o autor do livro. Por exemplo, o nome do arquivo:

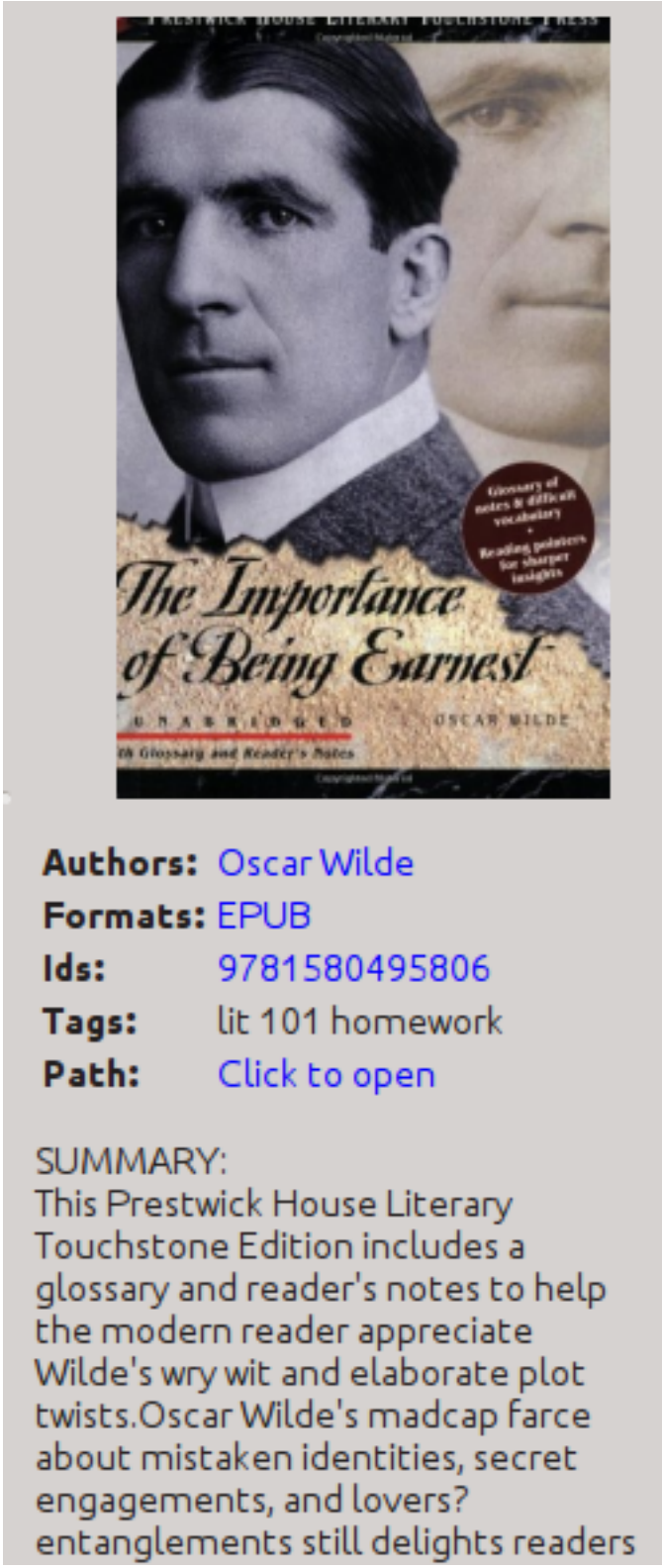
```
Foundation and Earth - Isaac Asimov.txt
```

será interpretado como tendo o título: Fundação e Terra e autor: Isaac Asimov

Dica

Se o nome do arquivo não contiver o hífen, a expressão regular acima falhará.

1.11 Detalhes do livro



A exibição Detalhes do livro mostra a capa e todos os metadados do livro atualmente selecionado. Ele pode ser ocultado

por meio do botão *Layout* no canto inferior direito da janela principal do calibre. Os nomes dos autores mostrados no painel Detalhes do livro podem ser clicados, eles o levarão por padrão para a página da Wikipedia do autor. Isso pode ser personalizado clicando com o botão direito do mouse no nome do autor e selecionando Gerenciar este autor.

Da mesma forma, se você baixar metadados para o livro, o painel Detalhes do livro mostrará automaticamente links apontando para as páginas da Web do livro na Amazon, WorldCat, etc. de onde os metadados foram baixados.

Você pode clicar com o botão direito do mouse em formatos individuais de e-book no painel Detalhes do livro para excluí-los, compará-los com suas versões originais, salvá-los no disco, abri-los com um programa externo, etc.

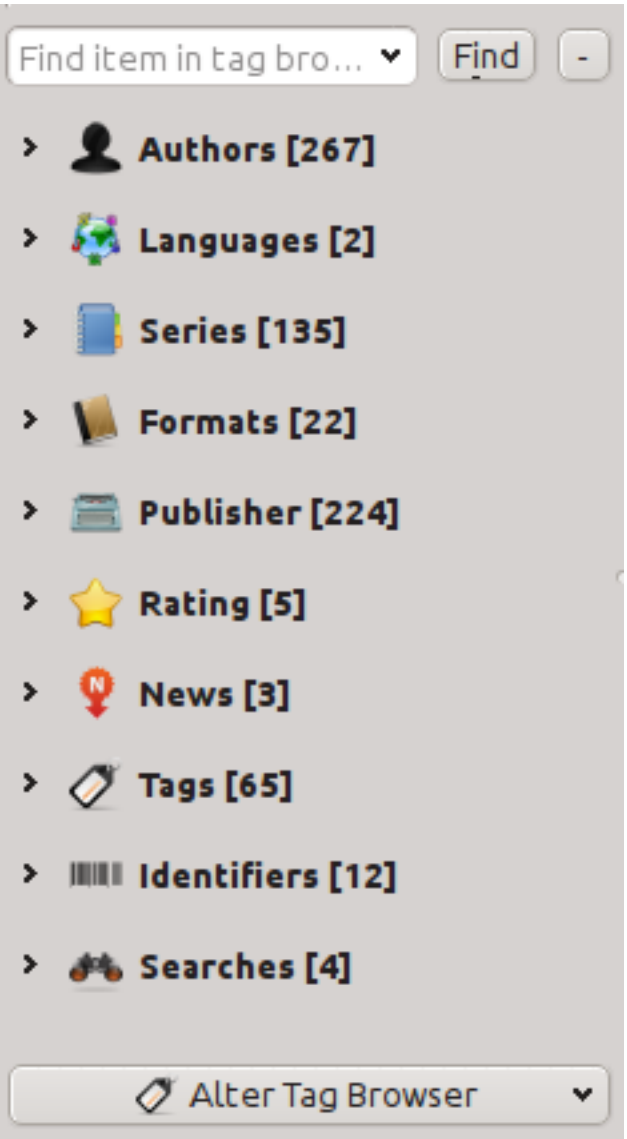
Você pode alterar a capa do livro simplesmente arrastando e soltando uma imagem no painel Detalhes do livro. Se desejar editar a imagem da capa em um programa externo, basta clicar com o botão direito do mouse nela e escolher *Abrir com*.

Você também pode adicionar arquivos de e-book ao livro atual arrastando e soltando os arquivos no painel Detalhes do livro.

Clicar duas vezes no painel Detalhes do livro abrirá em uma janela pop-up separada.

Por fim, você pode personalizar exatamente quais informações são exibidas no painel Detalhes do livro por meio de *Preferências → Interface → Aparência → Detalhes do livro*.

1.12 Navegador de tags



O Navegador de tags permite que você navegue facilmente por sua coleção por Autor/Tags/Série/etc. Se você clicar em qualquer item no Navegador de tags, por exemplo, o nome do autor Isaac Asimov, a lista de livros à direita será restrita a mostrar livros desse autor. Você também pode clicar em nomes de categorias. Por exemplo, clicar em "Série" mostrará todos os livros em qualquer série.

O primeiro clique em um item restringirá a lista de livros àqueles que contêm ou correspondem ao item. Continuando o exemplo acima, clicar em Isaac Asimov mostrará os livros desse autor. Clicar novamente no item mudará o que é mostrado, dependendo se o item tem filhos (consulte subcategorias e itens hierárquicos abaixo). Continuando o exemplo de Isaac Asimov, clicar novamente em Isaac Asimov restringirá a lista de livros àqueles que não são de Isaac Asimov. Um terceiro clique removerá a restrição, mostrando todos os livros. Se você mantiver pressionadas as teclas `Ctrl` ou `Shift` e clicar em vários itens, serão criadas restrições com base em vários itens. Por exemplo, você pode manter pressionada a tecla `Ctrl` e clicar nas tags História e Europa para encontrar livros sobre história europeia. O Navegador de tags funciona construindo expressões de pesquisa que são inseridas automaticamente na barra de pesquisa. Observar o que o Navegador de tags gera é uma boa maneira de aprender a construir expressões de pesquisa básicas.

Os itens no Navegador de tags têm seus ícones parcialmente coloridos. A quantidade de cor depende da classificação

média dos livros nessa categoria. Então, por exemplo, se os livros de Isaac Asimov tiverem uma média de quatro estrelas, o ícone de Isaac Asimov no Navegador de tags terá 4/5 coloridos. Você pode passar o mouse sobre o ícone para ver a classificação média.

Os itens de nível externo no *Navegador de tags*, como Autores e Séries, são chamados de categorias. Você pode criar suas próprias categorias, chamadas de *Categorias do usuário*, que são úteis para organizar itens. Por exemplo, você pode usar o *Editor de categorias do usuário* (clique no botão *Configurar* no canto inferior esquerdo do *Navegador de tags* e escolha *Gerenciar autores, tags, etc.* → *Categorias do usuário*) para criar uma categoria de usuário chamada `Autores favoritos`, em seguida, coloque os itens de seus favoritos na categoria. As categorias de usuário podem ter subcategorias. Por exemplo, a categoria de usuário `Favoritos.Autores` é uma subcategoria de `Favoritos`. Você também pode ter `Favoritos.Séries`, nesse caso haverá duas subcategorias em `Favoritos`. As subcategorias podem ser criadas clicando com o botão direito do mouse em uma categoria de usuário, escolhendo *Adicionar subcategoria a…* e inserindo o nome da subcategoria; ou usando o *Editor de categorias do usuário* inserindo nomes como o exemplo de Favoritos acima.

Você pode pesquisar categorias de usuário da mesma forma que categorias integradas, clicando nelas. Existem quatro pesquisas diferentes percorridas clicando:

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

Também é possível criar hierarquias dentro de algumas das categorias de texto, como tags, séries e colunas personalizadas. Essas hierarquias são mostradas com o pequeno triângulo, permitindo que os subitens sejam ocultados. Para usar hierarquias de itens em uma categoria, você deve primeiro ir para *Preferências* → *Interface* → *Aparência* e inserir o(s) nome(s) da categoria no campo "Categorias com itens hierárquicos". Feito isso, os itens dessa categoria que contêm pontos serão mostrados usando o pequeno triângulo. Por exemplo, suponha que você crie uma coluna personalizada chamada "Gênero" e indique que ela contém itens hierárquicos. Feito isso, itens como Mistério.Suspense e Mistério.Inglês serão exibidos como Mistério com o pequeno triângulo ao lado. Clicar no triângulo mostrará Suspense e Inglês como subitens. Consulte *Gerenciando subgrupos de livros, por exemplo "gênero"* (página 153) para obter mais informações.

Itens hierárquicos (itens com filhos) usam as mesmas quatro pesquisas 'clicar em' que as categorias de usuário. Os itens que não têm filhos usam duas das pesquisas: "tudo que corresponde" e "tudo que não corresponde".

Você pode arrastar e soltar itens no Navegador de tags em categorias de usuário para adicioná-los a essa categoria. Se a origem for uma categoria de usuário, manter pressionada a tecla `Shift` enquanto arrasta moverá o item para a nova categoria. Você também pode arrastar e soltar livros da lista de livros em itens no Navegador de tags; soltar um livro em um item faz com que esse item seja aplicado automaticamente aos livros soltos. Por exemplo, arrastar um livro para Isaac Asimov definirá o autor desse livro como Isaac Asimov. Soltá-lo na tag História adicionará a tag História às tags do livro.

Você pode encontrar facilmente qualquer item no Navegador de tags clicando no botão de pesquisa no canto inferior direito. Além disso, você pode clicar com o botão direito do mouse em qualquer item e escolher uma das várias operações. Alguns exemplos são ocultá-lo, renomeá-lo ou abrir uma caixa de diálogo "Gerenciar x" que permite gerenciar itens desse tipo. Por exemplo, a caixa de diálogo *Gerenciar autores* permite que você renomeie autores e controle como seus nomes são classificados.

Você pode controlar como os itens são classificados no Navegador de tags por meio do botão *Configurar* no canto inferior esquerdo do Navegador de tags. Você pode optar por classificar por nome, classificação média ou popularidade (popularidade é o número de livros com um item em sua biblioteca; por exemplo, a popularidade de Isaac Asimov é o número de livros em sua biblioteca de Isaac Asimov).

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

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

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

  Book metadata such as title is not available. Template database functions such as *book_count* (página 190) and *book_values* (página 191) will work, but the performance might not be acceptable. The following template functions will work in the GUI but won't work in the content server: *connected_device_name* (página 204), *connected_device_uuid* (página 204), *current_virtual_library_name* (página 204), *is_marked* (página 206), and *virtual_libraries* (página 207).

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

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

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

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

The icon is chosen using the following hierarchy:

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

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

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

1.13 Grade de capas

Você pode fazer com que o calibre exiba uma grade de capas de livros em vez de uma lista de livros, se preferir navegar por sua coleção por capas. A *Grade de capas* é ativada clicando no botão *Layout* no canto inferior direito da janela principal do calibre. Você pode personalizar os tamanhos das capas e o plano de fundo da *Grade de capas* por meio de *Preferências → Interface → Aparência → Grade de capas*. Você pode até fazer com que o calibre exiba qualquer campo especificado sob as capas, como título ou autores ou classificação ou uma coluna personalizada de sua própria criação.

1.14 Navegador de capas

Além da *Grade de capas* descrita acima, você também pode fazer com que o calibre exiba capas em uma única linha. Isso é ativado por meio do botão *Layout* no canto inferior direito da janela principal. Em *Preferências → Interface → Aparência → Navegador de capas* você pode alterar o número de capas exibidas e até mesmo fazer com que o *Navegador de capas* seja exibido em uma janela pop-up separada.

1.15 Adicionando notas para autores, séries, etc.

Você pode adicionar notas para um autor/série/tag/editora/etc. à sua biblioteca calibre. Para fazer isso, clique com o botão direito do mouse no nome do autor no *Navegador de tags* à esquerda ou no painel *Detalhes do livro* à direita e escolha *Criar nota* ou *Editar nota*.

Uma janela pop-up simples permitirá que você insira suas notas usando formatação básica e suportando links e imagens. Depois que uma nota para um autor é criada, ela pode ser visualizada facilmente no painel *Detalhes do livro* clicando no pequeno ícone de lápis ao lado do nome do autor.

Você pode pesquisar em todas as notas em sua biblioteca usando a ferramenta *Navegar pelas notas* pressionando `Ctrl+Shift+N` ou adicionando-a à barra de ferramentas por meio de *Preferências → Barras de ferramentas e menus*.

1.16 Visualização rápida

Às vezes, você deseja selecionar um livro e obter rapidamente uma lista de livros com o mesmo valor em alguma categoria (autores, tags, editora, série, etc.) que o livro atualmente selecionado, mas sem alterar a visualização atual da biblioteca. Você pode fazer isso com a Visualização rápida. A Visualização rápida abre uma segunda janela ou um painel na lista de livros, mostrando a lista de livros correspondentes ao valor de interesse. Por exemplo, suponha que você queira ver uma lista de todos os livros com um ou mais dos autores do livro atualmente selecionado. Clique na célula do autor na qual você está interessado e pressione a tecla 'Q' ou clique no ícone *Visualização rápida* na seção *Layout* da janela calibre. Uma janela ou painel será aberto com todos os autores desse livro à esquerda e todos os livros do autor selecionado à direita.

Alguns exemplos de uso da Visualização rápida: ver rapidamente quais outros livros:

- têm alguma(s) tag(s) aplicada(s) ao livro atualmente selecionado,
- estão na mesma série que o livro atual
- têm os mesmos valores em uma coluna personalizada que o livro atual
- são escritos por um dos mesmos autores do livro atual
- compartilham valores em uma coluna personalizada

Existem duas opções para onde as informações da Visualização rápida aparecem:

1. Ele pode abrir "desancorado": sobre a janela calibre e permanecerá aberto até que você o feche explicitamente.
2. Ele pode abrir "ancorado": como um painel na seção da lista de livros da janela principal do calibre.

Você pode mover a janela de ancorada para desancorada conforme desejado usando o botão "Ancorar/Desancorar".

O painel Visualização rápida pode ser deixado aberto permanentemente, caso em que segue os movimentos na lista de livros. Por exemplo, se você clicar na visualização da biblioteca calibre em uma coluna de categoria (tags, série, editora, autores, etc.) para um livro, o conteúdo da janela Visualização rápida mudará para mostrar a você no painel do lado esquerdo os valores nessa categoria para o livro selecionado (por exemplo, as tags para esse livro). O primeiro item dessa lista será selecionado e a Visualização rápida mostrará no painel do lado direito todos os livros em sua biblioteca que usam esse valor. Clique em um valor diferente no painel do lado esquerdo para ver os livros com esse valor diferente.

Clique duas vezes em um livro na janela Visualização rápida para selecionar esse livro na visualização da biblioteca. Isso também mudará a exibição dos itens na janela Visualização rápida (o painel do lado esquerdo) para mostrar os itens no livro recém-selecionado.

Clique duas vezes em um livro na janela Visualização rápida com `Shift-` ou `Ctrl-` pressionado para abrir a caixa de diálogo editar metadados desse livro na janela calibre. O livro editado será visualizado rapidamente quando você fechar a caixa de diálogo editar metadados.

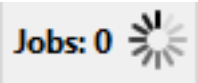
Você pode ver se uma coluna pode ser visualizada rapidamente passando o mouse sobre o cabeçalho da coluna e olhando a dica de ferramenta para esse cabeçalho. Você também pode saber clicando com o botão direito do mouse no cabeçalho da coluna para ver se a opção "Visualização rápida" é mostrada no menu, nesse caso, escolher essa opção Visualização rápida é equivalente a pressionar 'Q' na célula atual.

Opções (em *Preferências → Aparência → Visualização rápida*):

- Respeitar (ou não) a biblioteca virtual atual. Se marcado, a Visualização rápida mostra apenas os livros na biblioteca virtual atual. Padrão: respeitar bibliotecas virtuais
- Alterar o conteúdo da janela Visualização rápida quando a coluna for alterada na lista de livros usando as teclas do cursor. Padrão: não seguir alterações feitas com as teclas do cursor
- Alterar a coluna sendo "visualizada rapidamente" quando uma célula na janela Visualização rápida for clicada duas vezes. Caso contrário, o livro é alterado, mas a coluna que está sendo examinada não. Padrão: alterar a coluna

- Alterar a coluna sendo "visualizada rapidamente" para a coluna atual quando a tecla Enter for pressionada no painel Visualização rápida. Caso contrário, o livro é alterado, mas a coluna que está sendo examinada não. Padrão: alterar a coluna
- Escolha quais colunas são mostradas na janela/painel Visualização rápida.

1.17 Tarefas



O painel Tarefas mostra o número de tarefas em execução no momento. As tarefas são processos que são executados em um processo separado. Eles incluem a conversão de e-books e a comunicação com o dispositivo do leitor. Você pode clicar no painel de tarefas para acessar a lista de tarefas. Depois que uma tarefa for concluída, você pode ver um log detalhado dessa tarefa clicando duas vezes nela na lista. Isso é útil para depurar tarefas que podem não ter sido concluídas com sucesso.

1.18 Atalhos de teclado

o calibre possui vários atalhos de teclado para economizar tempo e movimento do mouse. Esses atalhos estão ativos na visualização da lista de livros (quando você não está editando os detalhes de um livro específico) e a maioria deles afeta o título que você selecionou. O Visualizador de e-books do calibre *possui seus próprios atalhos* (página 57) que podem ser personalizados nas *Preferências* do visualizador.

> **Nota**
>
> Observação: os atalhos de teclado do calibre não exigem uma tecla modificadora (Command, Option, Control, etc.), a menos que seja especificamente observado. Você só precisa pressionar a tecla da letra, por exemplo, E para editar.

Tabela1: Atalhos de teclado para o programa principal do calibre

Atalho de teclado	Ação
`F2 (Enter para macOS)`	Editar os metadados do campo atualmente selecionado na lista de livros.
`A`	Adicionar livros
`Shift+A`	Adicionar formatos aos livros selecionados
`C`	Converter livros selecionados
`D`	Enviar para o dispositivo
`Del`	Remover livros selecionados
`E`	Editar metadados de livros selecionados
`G`	Obter livros
`I`	Mostrar detalhes do livro
`K`	Editar Índice
`M`	Mesclar registros selecionados

continua na próxima página

Tabela 1 – continuação da página anterior

Atalho de teclado	Ação
Alt+M	Mesclar registros selecionados, mantendo os originais
O	Abrir conteúdo da pasta
P	Polir livros
S	Salvar no disco
T	Editar livro
V	Visualizar
Shift+V	Visualizar último livro lido
Alt+V/ Cmd+V para macOS	Visualizar formato específico
Alt+Shi	Alternar lista de tarefas
Alt+Shi	Alternar Navegador de capas
Alt+Shi	Alternar painel de detalhes do livro
Alt+Shi	Alternar Navegador de tags
Alt+Shi	Alternar Grade de capas
Alt+A	Mostrar livros do mesmo autor do livro atual
Alt+T	Mostrar livros com as mesmas tags do livro atual
Alt+P	Mostrar livros da mesma editora do livro atual
Alt+Shi	Mostrar livros da mesma série do livro atual
/, Ctrl+F	Focar na barra de pesquisa
Shift+C	Abrir a caixa de diálogo de pesquisa avançada
Shift+A	Alternar a barra de pesquisa
Esc	Limpar a busca atual
Shift+E	Focar a lista de livros
Ctrl+Es	Limpar a biblioteca virtual
Alt+Esc	Limpar a restrição adicional
Ctrl+*	Criar uma biblioteca virtual temporária com base na pesquisa atual
Ctrl+Ri	Selecionar a próxima guia da biblioteca virtual
Ctrl+Le	Selecionar a guia anterior da biblioteca virtual
N ou F3	Encontrar o próximo livro que corresponde à pesquisa atual (só funciona se o realce da pesquisa estiver ativado nas preferências de pesquisa)
Shift+N ou Shift+F	Encontrar o livro anterior que corresponde à pesquisa atual (só funciona se o realce da pesquisa estiver ativado nas preferências de pesquisa)
Ctrl+D	Baixar metadados e capas
Ctrl+R	Reiniciar o calibre
Ctrl+Sh	Reiniciar o calibre em modo de depuração
Shift+C	Adicionar livros vazios para o calibre
Ctrl+M	Alternar status marcado/desmarcado em livros selecionados
Ctrl+/ ou Ctrl+Al	Abrir o pop-up para pesquisar o texto completo de todos os livros da biblioteca
Q	Abrir o pop-up Visualização rápida para visualizar livros em séries/tags/etc. relacionados.
Shift+Q	Focar no painel Visualização rápida aberto
Shift+S	Realizar uma pesquisa no painel Visualização rápida
F5	Reaplicar a classificação atual

continua na próxima página

Tabela 1 – continuação da página anterior

Atalho de teclado	Ação
`Ctrl+Q`	Encerrar o calibre
`X`	Alternar rolagem automática da lista de livros
`Ctrl+Al`	Restringir os livros exibidos apenas aos livros que estão em uma categoria atualmente exibida no *Navegador de tags*
`B`	Navegar pelas notas (destaques e marcações) feitas pelo visualizador de E-book do calibre para todos os livros na biblioteca
`Ctrl+Sh`	Navegar pelas notas associadas a autores/séries/tags/etc.
`Alt+Shi`	Alternar o layout entre visualizações amplas e estreitas

CAPÍTULO 2

Adicionar seus sites de notícia favoritos ao calibre

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 o framework, seguir os exemplos na ordem listada abaixo:

- *Download totalmente automático.* (página 33)
  - *O blog calibre* (página 34)
  - *bbc.co.uk* (página 35)
- *Personalizando o processo de busca* (página 35)
  - *Usando a versão de impressão do bbc.co.uk* (página 35)
  - *Trocando os estilos do artigo* (página 36)
  - *Dividindo e separando* (página 37)
  - *Exemplo real* (página 37)
- *Dicas para desenvolver novas receitas* (página 40)
- *Leituras recomendadas* (página 41)
- *Documentação da API* (página 41)

2.1 Download totalmente automático.

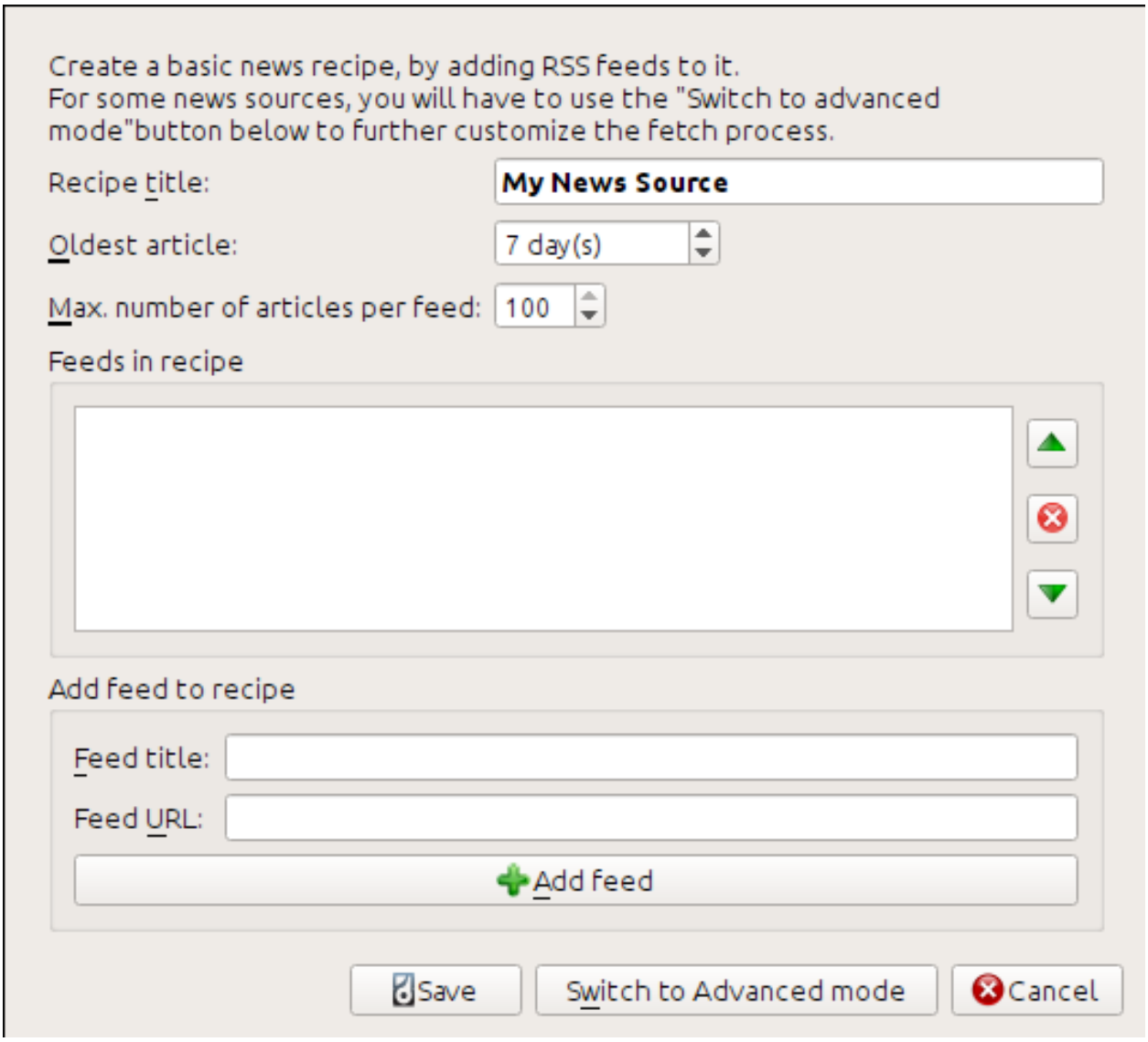
Se a sua fonte de notícias é bastante simples, calibre pode muito bem ser capaz de buscá-la por completo automaticamente, tudo que você precisa fazer é fornecer a URL. calibre reúne todas as informações necessárias para baixar uma fonte de notícias em um *recipe*. Para dizer ao calibre sobre uma fonte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

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

2.2 Personalizando o processo de busca

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 Usando a versão de impressão do bbc.co.uk

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

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

URL da versão de impressão
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):

Recipe source code (python)

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

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

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

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

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

Recipe source code (python)

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

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

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

In the above, `def print_version(self, url)` defines a *method* that is called by calibre for every article. `url` is the URL of the original article. What `print_version` does is take that url and replace it with the new URL that points to the print version of the article. To learn about Python[5] see the tutorial[6].

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

2.2.2 Trocando os estilos do 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:

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

```
no_stylesheets = True
```

A receita agora se parece com:

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 bom. Se você é um perfeccionista, você vai querer ler a próxima seção, que trata realmente de modificar o conteúdo baixado.

2.2.3 Dividindo e separando

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 37) recipe again. Looking at the source code (*HTML*) of a couple of articles (print version), we see that they have a footer that contains no useful information, contained in

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

Este poderá ser removido pela adição:

```
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
```

(continua na próxima página)

(continuação da página anterior)

```
class NYTimes(BasicNewsRecipe):

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

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

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

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

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

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

             elif ''.join(div['class']) in ['story', 'story headline']:
                 a = div.find('a', href=True)
                 if not a:
                     continue
                 url = re.sub(r'\?.*', '', a['href'])
```

(continua na próxima página)

(continuação da página anterior)

```
                url += '?pagewanted=all'
                title = self.tag_to_string(a, use_alt=True).strip()
                description = ''
                pubdate = strftime('%a, %d %b')
                summary = div.find(True, attrs={'class':'summary'})
                if summary:
                    description = self.tag_to_string(summary, use_alt=False)

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

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

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 51).

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

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

These remove everything before the first `<h1>` tag and everything after the first tag whose id is `footer`. See *remove_tags* (página 50), *remove_tags_before* (página 50), *remove_tags_after* (página 50).

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 43) actually does the login into the NYT website. Once logged in, calibre will use the same, logged in, browser instance to fetch all content. See mechanize[7] to understand the code in `get_browser`.

The next new feature is the *calibre.web.feeds.news.BasicNewsRecipe.parse_index()* (página 44) method. Its job is to go to https://www.nytimes.com/pages/todayspaper/index.html and fetch the list of articles that appear in *todays* paper. While more complex than simply using *RSS*, the recipe creates an e-book that corresponds very closely to the days paper. `parse_index` makes heavy use of BeautifulSoup[8] to parse the daily paper webpage. You can also use other, more modern parsers if you dislike BeautifulSoup. calibre comes with lxml[9] and html5lib[10], which are the recommended parsers. To use them, replace the call to `index_to_soup()` with the following:

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

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

2.3 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 334) 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[11] para a partilhar com outros utilizadores do Calibre.

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

Nota

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

Ver também

generated/pt/ebook-convert
The command line interface for all e-book conversion.

2.4 Leituras recomendadas

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 41)**
Documentação da classe "BasicNewsRecipe" e todos os seus métodos e campos importantes.

BasicNewsRecipe[12]
O código fonte do "BasicNewsRecipe"

Built-in recipes[13]
The source code for the built-in recipes that come with calibre

O fórum de receitas do Calibre[14]
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 :class:'BasicNewsRecipe'

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 seus sites de notícia favoritos ao calibre* (página 33).

classmethod adeify_images(*soup*)

Se sua receita quando convertida em EPUB eve problemas com imagens quando visualizado no Adobe Digital Editions, este método do within `postprocess_html()` (página 45).

classmethod image_url_processor(*baseurl*, *url*)

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

classmethod print_version(*url*)

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

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

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

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

Convenience method to take a BeautifulSoup[15] `Tag` and extract the text from it recursively, including any CDATA sections and alt tag attributes. Return a possibly empty Unicode string.

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

tag: BeautifulSoup[16] `Tag`

abort_article(*msg=None*)

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.

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.

Parâmetros

is_link – É Verdadeiro se a URL está vindo de um link interno em um arquivo HTML. Falso se a URL é a URL usada para baixar um artigo.

cleanup()

Chamado após todos artigos serem baixados. Use isso para fazer uma limpeza como deslogar de sites inscritos, 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[17]. By default it looks for the original link (for feeds syndicated via a service like FeedBurner or Pheedo) and if found, returns that or else returns article.link[18].

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

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

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

```
def get_browser(self, *a, **kw):
    br = super().get_browser(*a, **kw)
    # Add some headers
    br.addheaders += [
        ('My-Header', 'one'),
        ('My-Header2', 'two'),
    ]
    # Set some cookies
    br.set_cookie('name', 'value')
    br.set_cookie('name2', 'value2', domain='.mydomain.com')
    # Make a POST request with some data
    br.open('https://someurl.com', {'username': 'def', 'password': 'pwd'}).
↪read()
    # Do a login via a simple web form (only supported with mechanize␣
↪browsers)
    if self.username is not None and self.password is not None:
        br.open('https://www.nytimes.com/auth/login')
        br.select_form(name='login')
        br['USERID']   = self.username
        br['PASSWORD'] = self.password
        br.submit()
    return br
```

get_cover_url()

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

get_extra_css()

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

get_feeds()

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

get_masthead_title()

Override in subclass to use something other than the recipe title

get_masthead_url()

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

get_obfuscated_article(*url*)

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

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

get_url_specific_delay(*url*)

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

Retorna
A floating point number, the delay in seconds.

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

Convenience method that takes an URL to the index page and returns a BeautifulSoup[20] of it.

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

is_link_wanted(*url*, *tag*)

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

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 43). Return a list of `Feed` objects.

parse_index()

This method should be implemented in recipes that parse a website instead of feeds to generate a list of articles. Typical uses are for news sources that have a "Print Edition" webpage that lists all the articles in the current print edition. If this function is implemented, it will be used in preference to `BasicNewsRecipe.parse_feeds()` (página 44).

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

Each list of articles must contain dictionaries of the form:

```
{
'title'       : article title,
'url'         : URL of print version,
'date'        : The publication date of the article as a string,
'description' : A summary of the article
```

(continua na próxima página)

(continuação da página anterior)

```
'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 42).

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

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

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[21] 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[22] instance containing the downloaded *HTML*.

preprocess_image(*img_data*, *image_url*)

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

preprocess_raw_html(*raw_html*, *url*)

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

Note that this method acts *before* preprocess_regexps.

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

publication_date()

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

skip_ad_pages(*soup*)

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

soup: A BeautifulSoup[23] instance containing the downloaded *HTML*.

sort_index_by(*index*, *weights*)

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

index: 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.

articles_are_obfuscated = False

Set to True and implement *get_obfuscated_article()* (página 44) to handle websites that try to make it difficult to scrape content.

auto_cleanup = False

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

auto_cleanup_keep = None

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

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

browser_type = 'mechanize'

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

center_navbar = True

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

compress_news_images = False

Set this to False to ignore all scaling and compression parameters and pass images through unmodified. If True and the other compression parameters are left at their default values, images will be scaled to fit in the

screen dimensions set by the output profile and compressed to size at most (w * h)/16 where w x h are the scaled image dimensions.

compress_news_images_auto_size = 16

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

compress_news_images_max_size = None

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

conversion_options = {}

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

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

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

By default, the cover image returned by get_cover_url() will be used as the cover for the periodical. Overriding this in your recipe instructs calibre to render the downloaded cover into a frame whose width and height are expressed as a percentage of the downloaded cover. cover_margins = (10, 15, '#ffffff') pads the cover with a white margin 10px on the left and right, 15px on the top and bottom. Color names are defined here[24]. Note that for some reason, white does not always work in Windows. Use #ffffff instead

delay = 0

The default delay between consecutive downloads in seconds. The argument may be a floating point number to indicate a more precise time. See `get_url_specific_delay()` (página 44) to implement per URL delays.

description = ''

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

encoding = None

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

extra_css = None

Specify any extra *CSS* that should be added to downloaded *HTML* files. It will be inserted into *<style>* tags, just before the closing *</head>* tag thereby overriding all *CSS* except that which is declared using the style attribute on individual *HTML* tags. Note that if you want to programmatically generate the extra_css override the `get_extra_css()` (página 43) method instead. For example:

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

feeds = None

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

filter_regexps = []

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

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

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

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

handle_gzip = True

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

ignore_duplicate_articles = None

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

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

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 50). If this list is not empty, then the *<body>* tag will be emptied and re-filled with the tags that match the entries in this list. For example:

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

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

language = 'und'

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

masthead_url = None

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

match_regexps = []

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

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

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

Only one of `BasicNewsRecipe.match_regexps` (página 48) or `BasicNewsRecipe.filter_regexps` (página 48) should be defined.

max_articles_per_feed = 100

Maximum number of articles to download from each feed. This is primarily useful for feeds that don't have article dates. For most feeds, you should use `BasicNewsRecipe.oldest_article` (página 49)

needs_subscription = False

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

no_stylesheets = False

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

oldest_article = 7.0

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

preprocess_regexps = []

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

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

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

publication_type = 'unknown'

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

recipe_disabled = None

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

recipe_specific_options = None

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

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

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

recursions = 0

Number of levels of links to follow on article webpages

remove_attributes = []

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

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

remove_empty_feeds = False

If True empty feeds are removed from the output. This option has no effect if parse_index is overridden in the sub class. It is meant only for recipes that return a list of feeds using *feeds* or `get_feeds()` (página 43). It is also used if you use the ignore_duplicate_articles option.

remove_javascript = True

Convenient flag to strip all JavaScript tags from the downloaded HTML

remove_tags = []

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': 'advertisement'}
}
```

All keys are optional. For a full explanation of the search criteria, see Beautiful Soup[25] A common example:

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

Isto removerá todos as etiquetas * * e respetivos descendentes 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 50). 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 50). For example:

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

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

requires_version = (0, 6, 0)

Minimum calibre version needed to use this recipe

resolve_internal_links = False

If set to True then links in downloaded articles that point to other downloaded articles are changed to point to the downloaded copy of the article rather than its original web URL. If you set this to True, you might also need to implement `canonicalize_internal_url()` (página 42) to work with the URL scheme of your particular website.

reverse_article_order = False

Reverse the order of articles in each feed

scale_news_images = None

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

scale_news_images_to_device = True

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

simultaneous_downloads = 5

Number of simultaneous downloads. Set to 1 if the server is picky. Automatically reduced to 1 if `BasicNewsRecipe.delay` (página 47) > 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.

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

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 54)
- *Navegar por um livro digital* (página 54)
- *Realce de texto* (página 55)
- *Ler em voz alta* (página 55)
- *Searching the text* (página 55)
- *Following links using only the keyboard* (página 56)
- *Personalizando a aparência e a sensação de sua experiência de leitura* (página 56)
- *Pesquisa no dicionário* (página 56)
- *Copiando texto e imagens* (página 56)
- *Ampliar em imagens* (página 57)
- *Syncing with a paper edition of the current book* (página 57)
- *Atalhos do teclado* (página 57)
- *Non re-flowable content* (página 59)
- *Designing your book to work well with the calibre E-book viewer* (página 59)

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 Navegando 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 referência

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

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**
>
> Support for text-to-speech in browsers is very incomplete and bug-ridden so how well *Read aloud* will work in the in-browser viewer is dependent on how well the underlying browser supports text-to-speech.

3.5 Searching the text

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

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.

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

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 215).

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 Personalizando a aparência e a sensação de 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.

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 Pesquisa no 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 Copiando 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.

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

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 Syncing with a paper edition of the current book

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

3.12 Atalhos do teclado

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

Tabela1: Keyboard shortcuts for the calibre E-book viewer

Atalho de teclado	Ação
`Home`, `Ctrl+Ar` `Ctrl+Ar`	Scroll to the start of the current file in a multi file book
`Ctrl+Hc`	Rolar até o início do livro
`Ctrl+En`	Rolar até o fim do livro
`End`, `Ctrl+Ar` `Ctrl+Ar`	Scroll to the end of the current file in a multi file book
`ArrowUp`	Rolar para trás, suavemente no modo fluxo e por tela inteira no modo paginado
`ArrowDown`	Rolar para frente, suavemente no modo fluxo e por tela inteira no modo paginado
`ArrowLeft`	Rolar para a esquerda um pouco no modo fluxo e uma página no modo paginado
`ArrowRight`	Rolar para a direita um pouco no modo fluxo e uma página no modo paginado
`PageUp`, `Shift+S`	Rolar para trás por tela inteira
`PageDow` `Spacebar`	Rolar para frente por tela inteira
`Ctrl+Pa`	Rolar para a seção anterior
`Ctrl+Pa`	Rolar para a seção seguinte
`Alt+Arr`	Voltar
`Alt+Arr`	Avançar

continua na próxima página

Tabela 1 – continuação da página anterior

Atalho de teclado	Ação
Ctrl+T	Toggle Table of Contents
Ctrl+S	Ler em voz alta
Alt+P	Change settings quickly by creating and switching to *profiles*
Alt+f	Seguir links com o teclado
Ctrl+C	Copiar para a área de transferência
Alt+C	Copiar a localização atual para a área de transferência
Ctrl+Sh	Copiar a localização atual como um URL calibre:// para a área de transferência
/, Ctrl+f, Cmd+f	Iniciar busca
F3, Enter	Buscar seguinte
Shift+F Shift+E	Localizar anterior
Ctrl+Pl Meta+Pl	Aumentar tamanho da fonte
Ctrl+Mi Meta+Mi	Diminuir tamanho da fonte
Ctrl+0	Restaurar o tamanho padrão de fonte
Ctrl+]	Aumentar o número de páginas por tela
Ctrl+[	Diminuir o número de páginas por tela
Ctrl+Al	Tornar o número de páginas por tela automático
F11, Ctrl+Sh	Alternar em tela cheia
:kbd:Ctrl·	Alternar entre Modo Paginado e Modo Fluxo para o layout do texto
Ctrl+W	Ativar a barra de rolagem
Ctrl+X	Ativar o Modo de Referência
Ctrl+B	Mostrar/ocultar marcadores
Ctrl+Al	Novo marcador
Ctrl+N, Ctrl+E	Exibir metadados do livro
Ctrl+Al Ctrl+Al	Recarregar livro
Ctrl+Sh	Alterar a seleção atual em uma palavra para frente
Ctrl+Sh	Alterar a seleção atual em uma palavra para trás
Shift+A	Alterar a seleção atual em um caractere para frente
Shift+A	Alterar a seleção atual em um caractere para trás
Shift+A	Alterar a seleção atual em uma linha para frente
Shift+H	Extender a seleção atual para o início da linha
Shift+E	Extender a seleção atual para o final da linha
Ctrl+A	Selecionar tudo
Shift+A	Alterar a seleção atual em uma linha para trás
Ctrl+Sh	Alterar a seleção atual em um parágrafo para frente

continua na próxima página

Tabela 1 – continuação da página anterior

Atalho de teclado	Ação
Ctrl+Sh	Alterar a seleção atual em um parágrafo para trás
Esc, Menu-Key	Mostrar os controles do Visualizador de e-books
Ctrl+Cc Ctrl+Es Meta+Es Meta+Cc	Mostrar as preferências do Visualizador de e-books
Ctrl+G, ;, :	Ir para um local ou posição específica do livro
Ctrl+Sp	Ativar auto rolagem
Alt+Arr	Auto rolar mais rápido
Alt+Arr	Auto rolar mais devagar
Ctrl+I	Mostrar/ocultar o Inspetor
Ctrl+L	Mostrar/ocultar o painel de pesquisa de palavras
Ctrl+Q (Cmd+Q on macOS)	Encerrar
Ctrl+P	Imprimir livro para PDF
Ctrl+F1	Ativar a barra de ferramentas
Ctrl+H	Ativar o painel de destaques
Ctrl+D	Editar este livro

3.13 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 }
```

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

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

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

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

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

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

`body.calibre-footnote-container`
Set when displaying a popup footnote

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

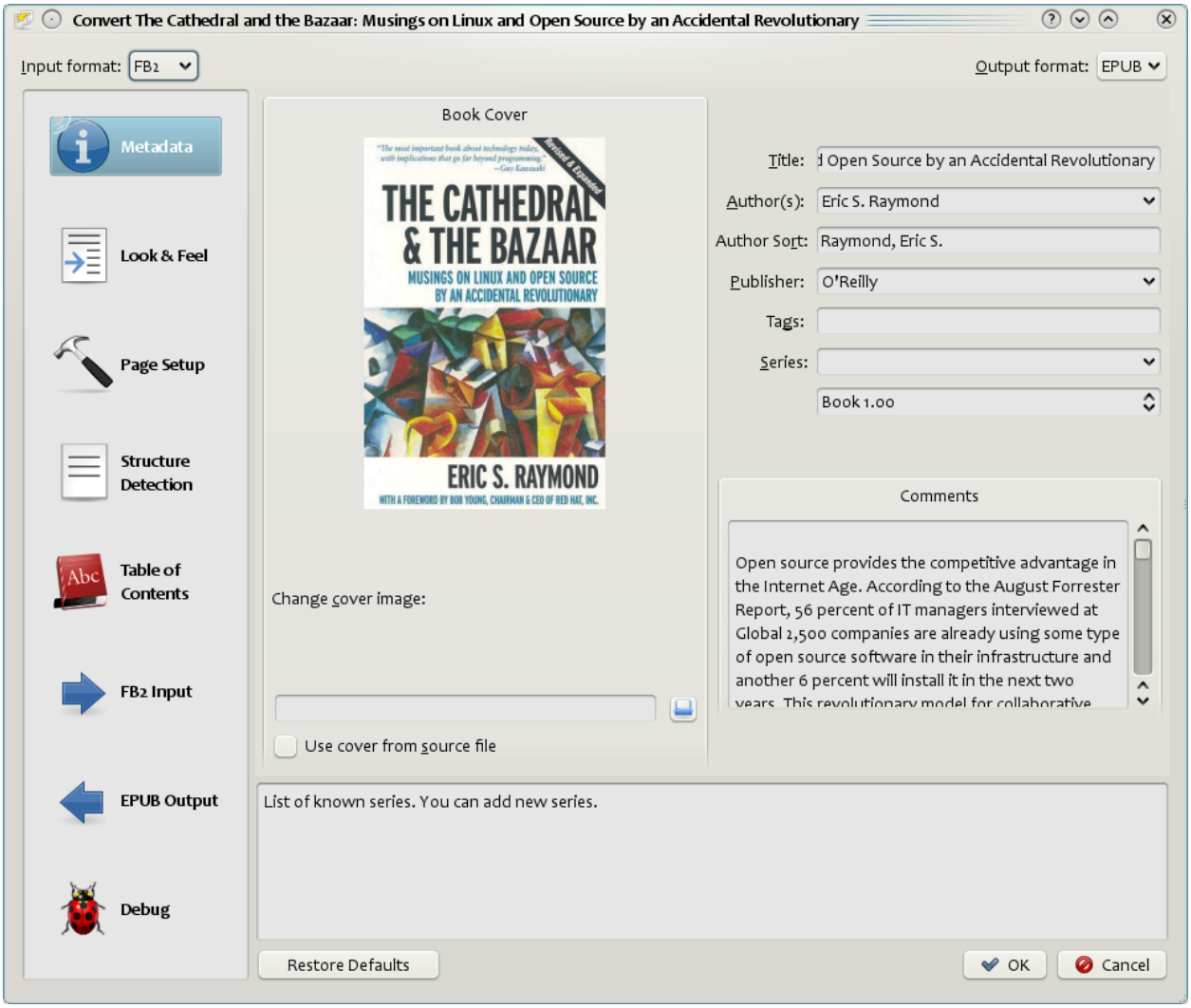
[28] https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties

CAPÍTULO 4

Conversão de E-book

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 *ebook-convert* (página 325). Dentro do calibre, você pode obter ajuda sobre qualquer configuração individual, mantendo seu mouse sobre ela. Uma dica irá aparecer, descrevendo a configuração.

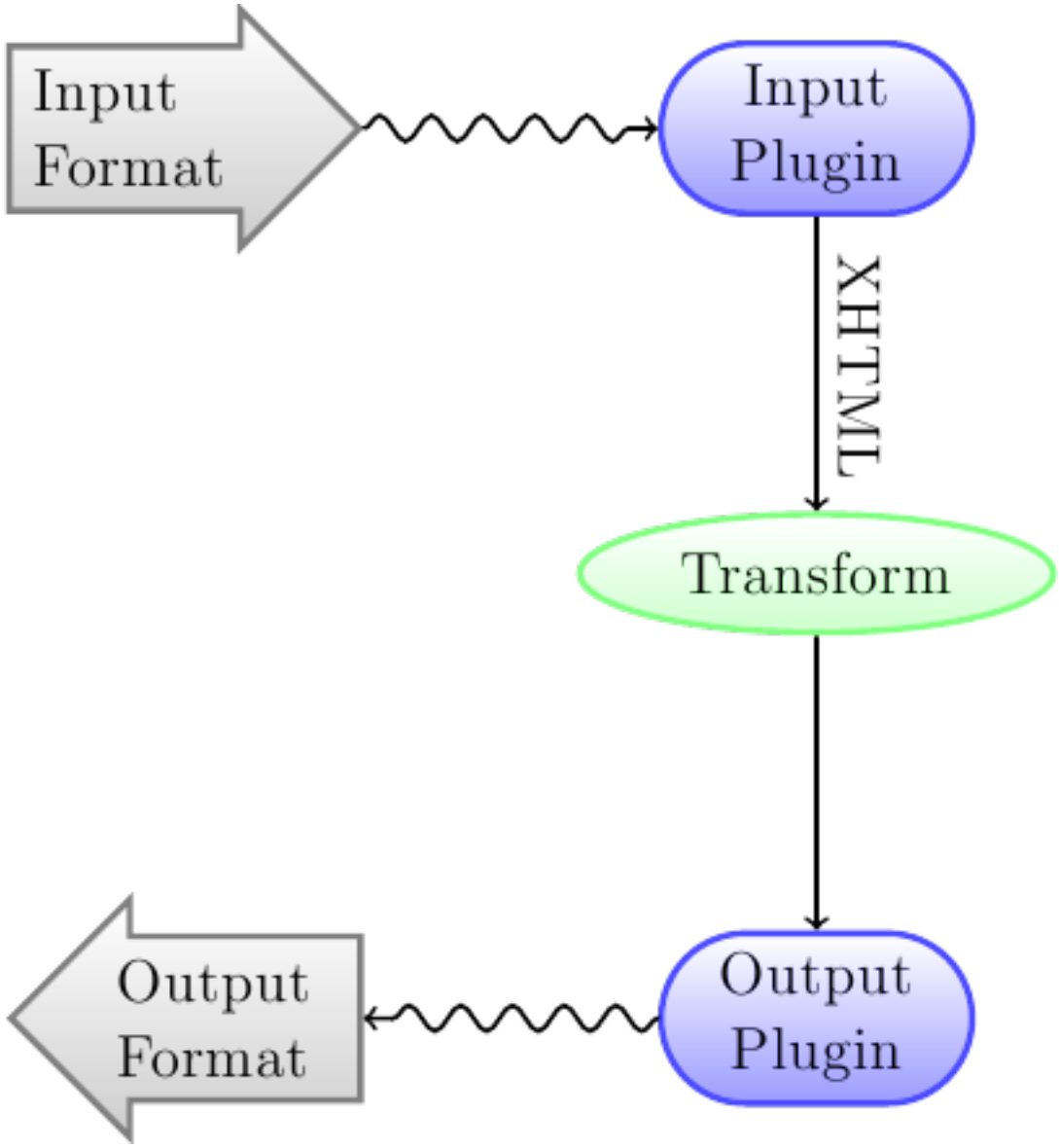


Conteúdo

- *Introdução* (página 63)
- *Aparência* (página 64)
- *Configurar Página* (página 67)
- *Processamento heurístico* (página 67)
- *Buscar & Substituir* (página 69)
- *Detecção da Estrutura* (página 69)
- *Índice* (página 70)
- *Usando imagens como títulos de capítulos ao converter documentos de entrada HTML* (página 72)
- *Usando atributos de tag para fornecer o texto para entradas no Sumário* (página 72)
- *Modo como as opções são definidas/guardadas para a conversão* (página 73)
- *Formatar dicas específicas* (página 73)

4.1 Introdução

A primeira coisa a entender sobre o sistema de conversão é que ele é projetado como um pipeline e 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 130).

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.

Você pode ver esse processo em ação usando a opção de depuração . Basta especificar o caminho para uma pasta para a saída de depuração. Durante a conversão, o calibre colocará o XHTML gerado pelos vários estágios do pipeline de conversão em diferentes subpastas. As quatro subpastas são:

Tabela1: Etapas do processo de conversão

Pasta	Descrição
entrada	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 detecção de estrutira, mas antes do achatamento de CSS e conversão de tamanho de fonte. Utilizado para depurar a conversão de fonte 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.

Se você quiser editar o documento de entrada um pouco antes de convertê-lo, a melhor coisa a fazer é editar os arquivos na subpasta :file:'input', compactá-lo e usar o arquivo ZIP como formato de entrada para conversões subsequentes. Para fazer isso, use a caixa de diálogo :guilabel:'Editar meta informações' para adicionar o arquivo ZIP como um formato para o livro e, em seguida, no canto superior esquerdo da caixa de diálogo de conversão, selecione ZIP como o formato de entrada.

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

Conteúdo

- *Fontes* (página 65)
- *Texto* (página 66)
- *Layout* (página 66)
- *Estilização* (página 66)
- *Transformar estilos* (página 67)
- *Transformar HTML* (página 67)

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 *Configurar Página* (página 67)). 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 de origem que estamos convertendo foi produzido por alguém com uma visão excelente e tem uma tamanho básico de fonte de 8pt. Isso significa que a maior parte do texto no documento está dimensionado em 8pts, enquanto os títulos são um pouco maior (digamos 10 e 12 pt) e as notas de rodapé um pouco menor em 6pt. Agora se nós usamos 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 66).

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 Layout

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; }
```

Em seguida, no documento de origem, marque os parágrafos que precisam de espaçamento com 'class="spacious"'. Se o documento de entrada não estiver em HTML, use a opção Depurar, descrita na Introdução para obter HTML (use a subpasta :file:'input').

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 Estilização

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 você quiser trocar para a o entalhe 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 a transformar estilos, mas permite fazer alterações no conteúdo HTML do livro. Você pode substituir uma tapa por outra, adicionar classes ou outros atribuir às tags com base em seu conteúdo, etc.

4.3 Configurar 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.

Se você sabe que o arquivo que está convertendo se destina a ser usado em um determinado dispositivo/plataforma de software, escolha o perfil de entrada correspondente, caso contrário, basta escolher o perfil de entrada padrão. Se você sabe que os arquivos que está produzindo são destinados a um tipo de dispositivo, escolha o perfil de saída correspondente. Caso contrário, escolhe um dos perfis de saída Genéricos. Se você está estiver convertendo para MOBI ou AZW3, então você quase sempre vai querer escolher um dos perfis de saída para Kindle. Fora isso, sua melhor aposta para dispositivos modernos de leitura de Ebook é escolher o perfil de saída: *Generic e-ink HD*.

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'
Essa opção controla o calibre do algoritmo usado para remover quebras de linha rígidas. Por exemplo, se o valor dessa opção for 0,4, isso significa que o calibre removerá quebras de linha rígidas do final de linhas cujos comprimentos são menores que o comprimento de 40% de todas as linhas no documento. Se o documento tiver apenas algumas quebras de linha que precisam de correção, esse valor deve ser reduzido para algo entre 0,1 e 0,2.

:guilabel:'Detectar e formatar capítulos não formatados e subtítulos'
Se o documento não tiver títulos de capítulo e títulos formatados de forma diferente do restante do texto, o calibre poderá usar essa opção para tentar detectá-los e cercá-los com tags de cabeçalho. <h2> são usadas para títulos de capítulos; <h3> são usadas para todos os títulos detectados.

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 as sequências de etiquetas <h1> ou <h2>
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'
Essa opção fará com que o calibre analise as linhas em branco incluídas no documento. Se cada parágrafo for intercalado com uma linha em branco, o calibre removerá todos esses parágrafos em branco. Sequências de várias linhas em branco serão consideradas quebras de cena e mantidas como um único parágrafo. Essa opção difere da opção :guilabel:'Remove paragraph espaçamento' em :guilabel:'Look and Feel' porque ela realmente modifica o conteúdo HTML, enquanto a outra opção modifica os estilos do documento. Essa opção também pode remover parágrafos que foram inseridos usando a opção :guilabel:'Inserir linha em branco' do calibre.

: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 irá substituir os marcadores de quebra de cena que encontrar com o texto de substituição especificado pelo usuário. 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 tag de imagem (coloque a imagem em uma pasta 'Imagens' dentro do EPUB após a conversão):

Exemplo de régua horizontal com estilos:
1

:guilabel:' Remover hífens 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 Buscar & 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 Detecção da 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.

Por padrão, o Calibre usa a seguinte expressão para detectar capítulos:

```
//*[((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 padrão para a detecção de quebra 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**
>
> A expressão padrão pode depender do formato de entrada que você está convertendo.

4.6.2 Extras

Existem algumas outras opções para esta seção.

: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*.

Se for detectado um número inferior ao número de capítulos :guilabel:'Chapter threshold', o calibre adicionará quaisquer hiperlinks encontrados no documento de entrada ao Índice. Isso geralmente funciona bem: muitos documentos de entrada incluem um Sumário com hiperlink logo no início. A opção :guilabel:'Número de links' pode ser usada para controlar esse comportamento. Se definido como zero, nenhum link será adicionado. Se definido como um número maior que zero, no máximo esse número de links será adicionado.

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

Então, nós colocamos a opção como:

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

Isto vai resultar em um sumário automaticamente gerado que se parece como:

```
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 Usando imagens como títulos de capítulos ao converter documentos de entrada HTML

Suponha que você queira usar uma imagem como título do capítulo, mas ainda queira que o calibre seja capaz de gerar automaticamente um Índice para você a partir dos títulos dos capítulos. Use a seguinte marcação HTML para conseguir isso:

```
<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 Usando atributos de tag para fornecer o texto para entradas no Sumário

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>
```

(continua na próxima página)

(continuação da página anterior)

```
    </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):

- Dos padrões definidos 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.
- Nas configurações definidas na caixa de diálogo Conversão em massa

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 Formatar dicas específicas

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 do Microsoft Word

O calibre pode converter automaticamente arquivos `` .docx`` criados pelo Microsoft Word 2007 e mais recentes. Basta adicionar o arquivo ao calibre e clicar em converter.

> **Nota**
>
> Existe um **:download_arquivo:`demo .docx arquivo <demos/demo.docx>`** que demonstra as capacidades do mecanismo de conversão do Calibre. Basta baixá-lo e convertê-lo para EPUB ou AZW3 para ver o que o Calibre pode fazer.

O calibre gerará automaticamente um Sumário com base nos títulos, se você os marcar com os estilos `` Título 1 ", `` Título 2 " etc. no Microsoft Word. Abra o e-book de saída no visualizador de e-books de calibre e clique no botão: guilabel: *Table of Contents* para visualizar o Table of Contents gerado.

Arquivos .doc mais 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`.

Outra alternativa é usar o LibreOffice gratuito. Abra seu arquivo .doc no LibreOffice e salve-o como .docx, que pode ser convertido diretamente no Calibre.

4.11.2 converter documentos em 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 entrada TXT suporta várias opções para diferenciar como os parágrafos são detectados.

: 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 que uma ou mais linhas em branco são um limite de parágrafo:

```
This is the first.

This is the
second paragraph.
```

: guilabel: ***Estilo de parágrafo: Único***
Assume que toda 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 formatação especial ao texto, o documento é convertido em HTML sem outras alterações.

4.11.3 Converter documentos PDF

Documentos PDF são um dos piores formatos para converter. Eles são um tamanho de página fixo e formato de posicionamento de texto. Ou seja, é muito difícil determinar onde termina um parágrafo e começa outro. O calibre tentará desembrulhar parágrafos usando um fator configurável, :guilabel:'Line un-wrapping factor'. Esta é uma escala usada para determinar o comprimento no qual uma linha deve ser desembrulhada. Os valores válidos são decimais entre 0 e 1. O padrão é 0,45, logo abaixo do comprimento médio da linha. Diminua esse valor para incluir mais texto na desquebragem. Aumentar para incluir menos. Você pode ajustar esse valor nas configurações de conversão em :guilabel:'PDF Input'.

Além disso, eles geralmente têm cabeçalhos e rodapés como parte do documento que será incluído com o texto. Use o painel :guilabel:'Pesquisar e substituir' para remover cabeçalhos e rodapés para atenuar esse problema. Se os cabeçalhos e rodapés não forem removidos do texto, ele poderá descartar a desquebra do parágrafo. Para saber como usar as opções de remoção de cabeçalho e rodapé, leia :ref:'regexptutorial'.

Algumas limitações do processamento de PDF são:

- Documentos complexos, multi colunares e 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.
- Links e Índices não são suportados
- PDFs que usam fontes não-Unicode incorporadas para representar caracteres que não estejam em inglês resultarão em saída distorcida para esses caracteres

- 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 quadrinhos

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 arquivo .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 Exemplo de formatação avançada de um EPUB

Várias formatações avançadas para arquivos EPUB são demonstradas neste :d ownload_file:'demo file <demos/demo.epub>'. O arquivo foi criado a partir de HTML codificado à mão usando calibre e destina-se a ser usado como um modelo para seus próprios esforços de criação de EPUB.

O HTML de origem a partir do qual ele foi criado está disponível :d ownload_file:'demo.zip <demos/demo.zip>'. As configurações usadas para criar o EPUB a partir do arquivo ZIP são:

```
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 documentos 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.

Para permitir a detecção automática de capítulos, você precisa marcá-los com os estilos internos chamados :guilabel:'Heading 1', :guilabel:'Heading 2', …, :guilabel:'Heading 6' (:guilabel:'Heading 1' equivale à tag HTML "<h1>", :guilabel:'Posições 2' a "<h2>", etc). Quando você converte em calibre, você pode inserir qual estilo você usou na caixa :guilabel:'Detectar capítulos em'. Exemplo:

- If you mark Chapters with style *Heading 2*, you have to set the 'Detect chapters at' box to `//h:h2`
- Para um sumário aninhado com Seções marcadas com :guilabel:'Título 2' e os Capítulos marcados com :guilabel:'Título 3', você precisa digitar "//h:h2|//h:h3". Na página Converter - sumário, defina a caixa :guilabel:'Level 1 TOC' como "//h:h2" e a caixa :guilabel:'Level 2 TOC' como "//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!

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 Convertendo para PDF

A primeira configuração, mais importante, para decidir ao converter para PDF é o tamanho da página. Por padrão, o calibre usa um tamanho de página de "U.S. Letter". Você pode alterar isso para outro tamanho de página padrão ou um tamanho completamente personalizado na seção :guilabel:'PDF Output' da caixa de diálogo de conversão. Se você estiver gerando um PDF para ser usado em um dispositivo específico, poderá ativar a opção para usar o tamanho da página a partir do perfil :guilabel:'output profile'. Portanto, se o seu perfil de saída estiver definido como Kindle, o calibre criará um PDF com tamanho de página adequado para visualização na pequena tela do Kindle.

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>
```

(continua na próxima página)

(continuação da página anterior)

```
    <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>
```

Isso exibirá o título à esquerda e o autor à direita, em um tamanho de fonte menor que o texto principal.

Você também pode usar a seção atual nos modelos, 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
- `_WIDTH_PIXELS_` - a largura da área de cabeçalho/rodapé em pixels
- `_HEIGHT_PIXELS_` - a altura da área de cabeçalho/rodapé em pixels

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>
```

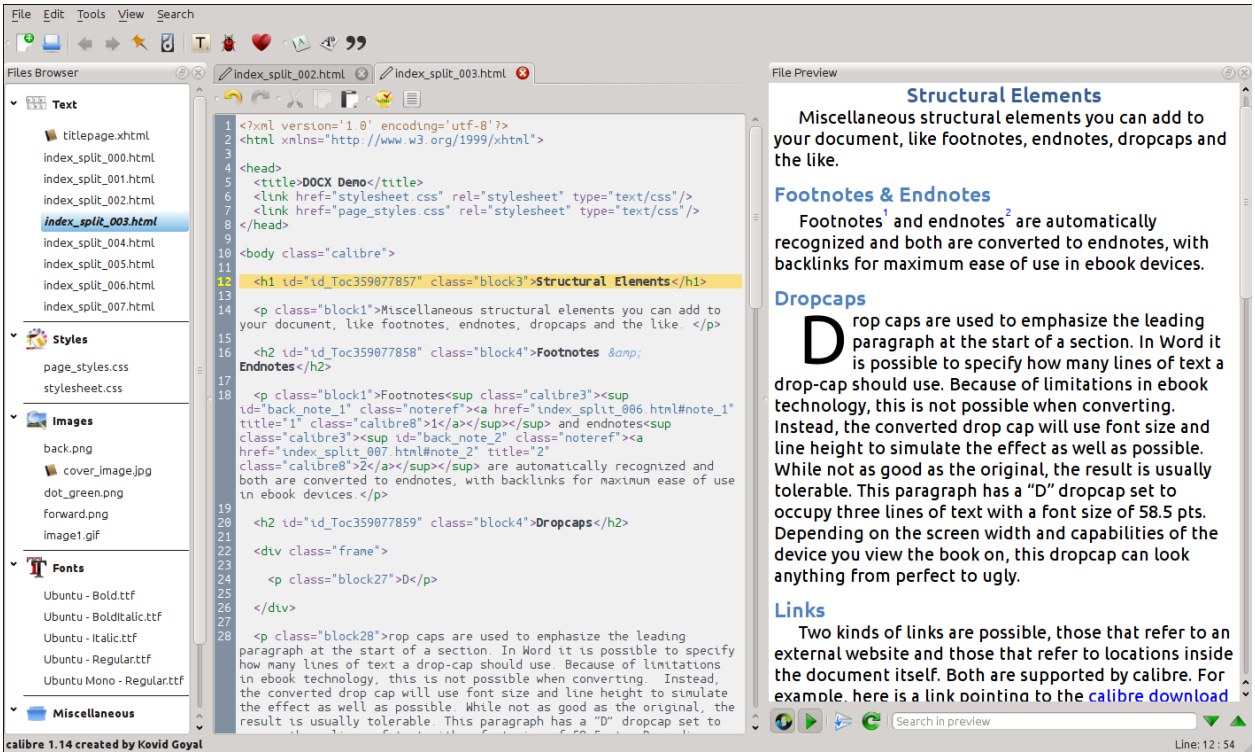
Em seguida, na seção de saída em PDF da caixa de diálogo de conversão, ative a opção para: guilabel: *Use as margens da página do documento que está sendo convertido*. Agora todas as páginas geradas a partir deste arquivo HTML terão margens `` 10pt``.

CAPÍTULO 5

Editando e-books

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

Você pode utilizar o editor clicando com botão direito em qualquer ebook no Calibre e selecionar *Editar Ebook*.



Conteúdo

- *Fluxo de trabalho básico* (página 83)
- *O navegador de ficheiros* (página 85)
  - *Renomear arquivos* (página 86)
  - *Mesclar arquivos* (página 86)
  - *Alterando a Ordem de arquivo de texto* (página 86)
  - *Marcando a capa* (página 87)
  - *Deletando arquivos* (página 87)
  - *Exportar arquivos* (página 87)
  - *Adicionar novas imagens / fonts / etc, ou a criar novos arquivos em branco* (página 87)
  - *Substituindo arquivos* (página 87)
  - *Ligar eficientemente folhas de estilo a ficheiros HTML* (página 87)
- *Buscar & Substituir* (página 87)
  - *Buscas salvas* (página 88)
  - *Modo de função* (página 88)
  - *Pesquisar ignorando etiquetas HTML* (página 88)
- *Ferramentas automatizadas* (página 88)
  - *Editando a Tabela de Conteúdos* (página 89)
  - *A verificar o livro* (página 90)
  - *Adicionando uma capa* (página 91)
  - *Incorporando fontes referenciadas* (página 91)
  - *Efetuando subconjunto de fontes embutidas* (página 91)
  - *Pontuação inteligente* (página 91)
  - *Transformar as propriedades CSS* (página 91)
  - *Removendo regras CSS não usadas* (página 92)
  - *Corrigir HTML* (página 92)
  - *Embelezando arquivos* (página 92)
  - *Inserir índice em linha* (página 92)
  - *Definir a semântica* (página 92)
  - *Filtrando informações do estilo* (página 93)
  - *Atualização das estruturas internas do livro* (página 93)
- *Pontos de verificação* (página 93)
- *O painel de pré-visualização em tempo real* (página 95)
  - *Dividindo arquivos HTML* (página 96)
- *O Painel CSS em tempo real* (página 97)

- *Ferramentas diversas* (página 98)
  - *A vista de índice* (página 98)
  - *Verificando a ortografia das palavras no livro* (página 98)
  - *Inserindo caracteres especiais* (página 99)
  - *Vista de inspeção de código* (página 100)
  - *Checando ligações externas* (página 100)
  - *Descarregar recursos externos* (página 100)
  - *Organizar ficheiros em pastas por tipo* (página 100)
  - *Importing files in other e-book formats as EPUB* (página 100)
  - *A ferramenta de relatórios* (página 110)
- *Características especiais no editor de código* (página 111)
  - *Realçar sintaxe* (página 111)
  - *Ajuda de conteúdo sensível* (página 111)
  - *Auto-completar* (página 112)
  - *Trechos* (página 112)

5.1 Fluxo de trabalho básico

Nota

Está disponível um vídeo contendo uma visita guiada ao editor de ebooks do Calibre aqui[29].

When you first open a book with the Edit book tool, you will be presented with a list of files on the left. These are the individual HTML files, stylesheets, images, etc. that make up the content of the book. Simply double click on a file to start editing it. Note that if you want to do anything more sophisticated than making a few small tweaks, you will need to know HTML Tutorial[30] and CSS Tutorial[31].

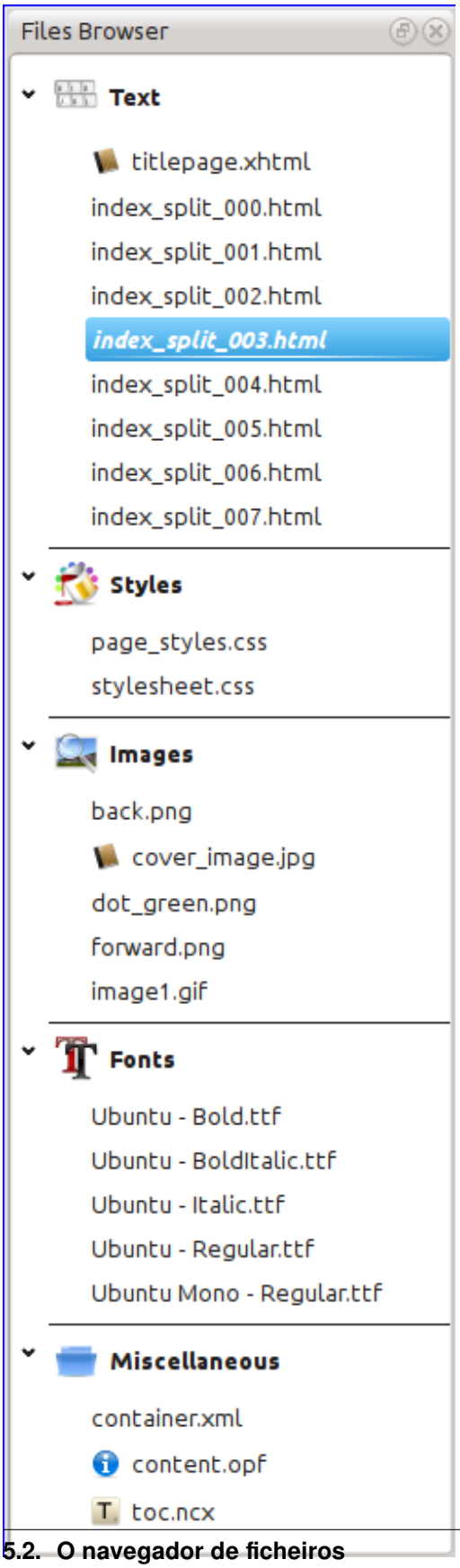
Conforme vai fazendo alterações ao HTML ou ao CSS no editor, as mudanças serão pré-visualizadas, no momento, no painel de pré-visualização à direita. Quando estiver satisfeito com o modo como se apresentam as mudanças que efetuou, clique no botão *Guardar* ou use *Ficheiro → Guardar* para guardar as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

[29] https://calibre-ebook.com/pt/demo#tutorials
[30] https://www.w3schools.com/Html/default.asp
[31] https://www.w3schools.com/css/default.asp

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.

Você pode executar diversas ações em arquivos individualmente, clicando com o botão direito neles.

5.2.1 Renomear arquivos

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 Mesclar arquivos

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 Alterando a Ordem de arquivo 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 Marcando 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 Deletando arquivos

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 arquivos

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 / fonts / etc, ou a criar novos arquivos em branco

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 Substituindo arquivos

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.

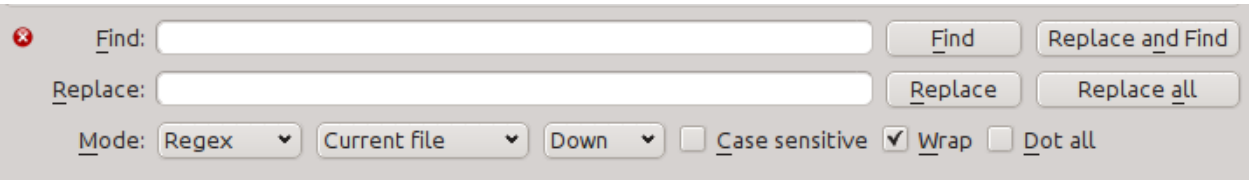
5.2.9 Ligar eficientemente folhas de estilo a ficheiros HTML

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 Buscar & 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 como usar expressões regulares no calibre* (página 215).



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 como usar expressões regulares no calibre* (página 215).

5.3.1 Buscas salvas

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 100).

5.3.3 Pesquisar ignorando etiquetas HTML

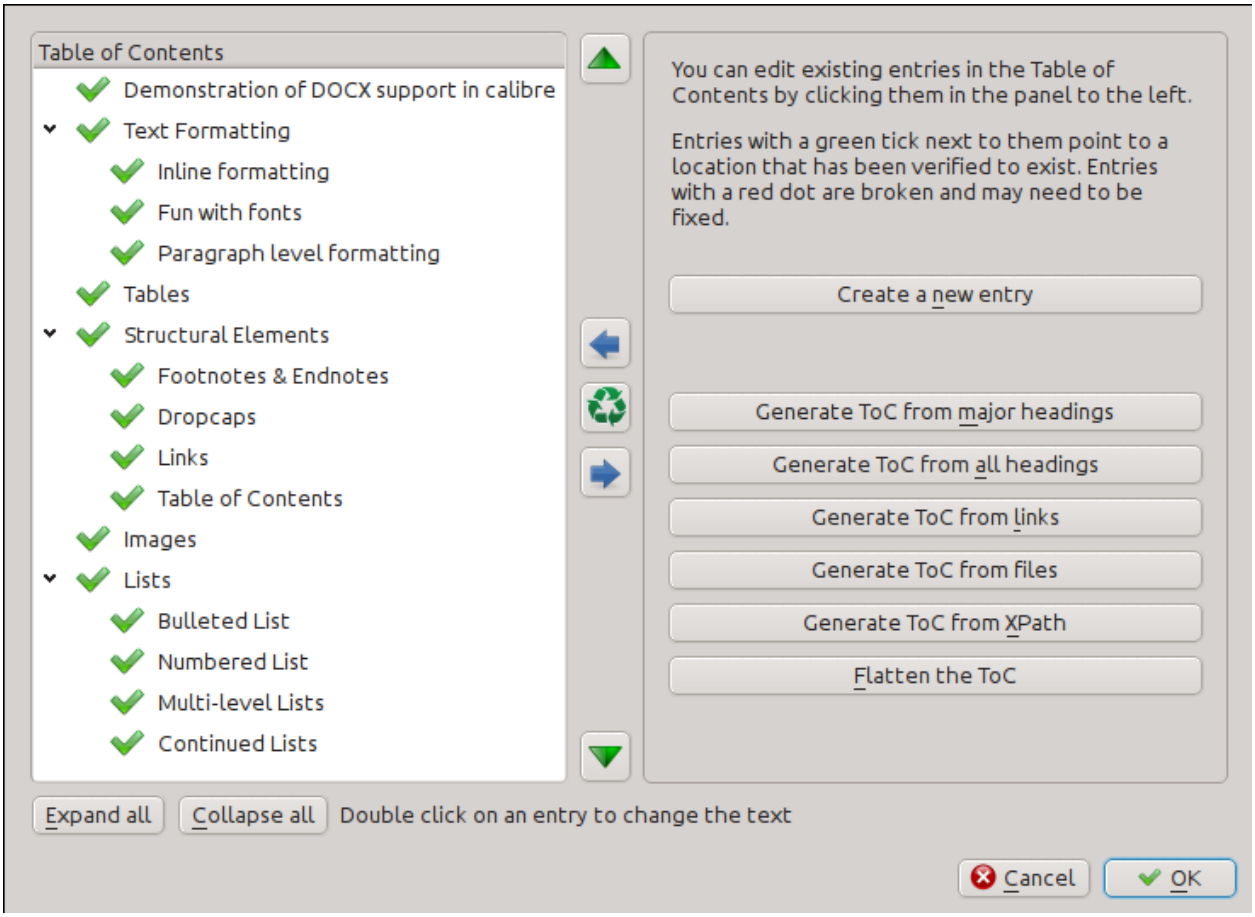
There is also a dedicated tool for searching for text, ignoring any HTML tags in between. For example, if the book has the HTML `Empahisis on a <i>word</i>.` you can search for `on a word` and it will be found even though there is an `<i>` tag in the middle. Use this tool via the *Search → Search ignoring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

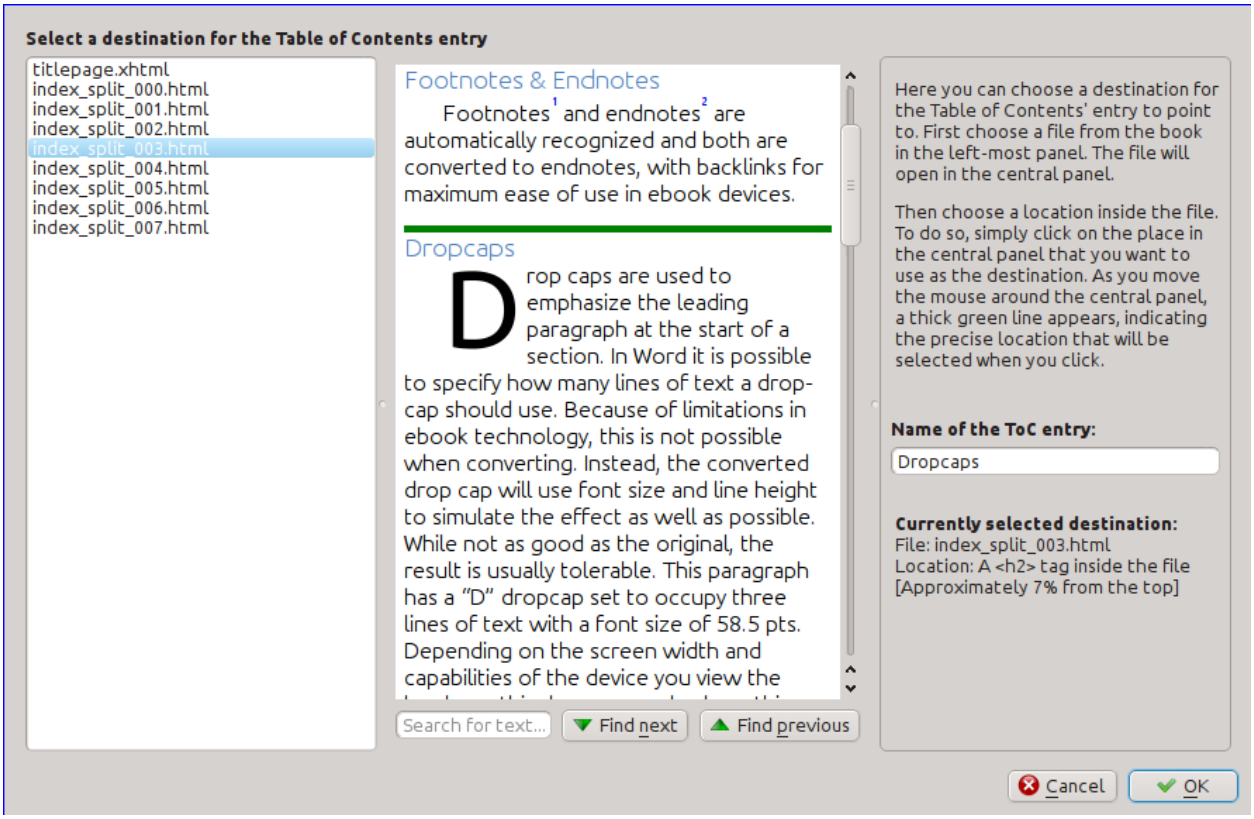
Existe uma ferramenta dedicada para facilitar a edição do índice. Invoque-a com *Ferramentas → Índice → Editar índice*.



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.

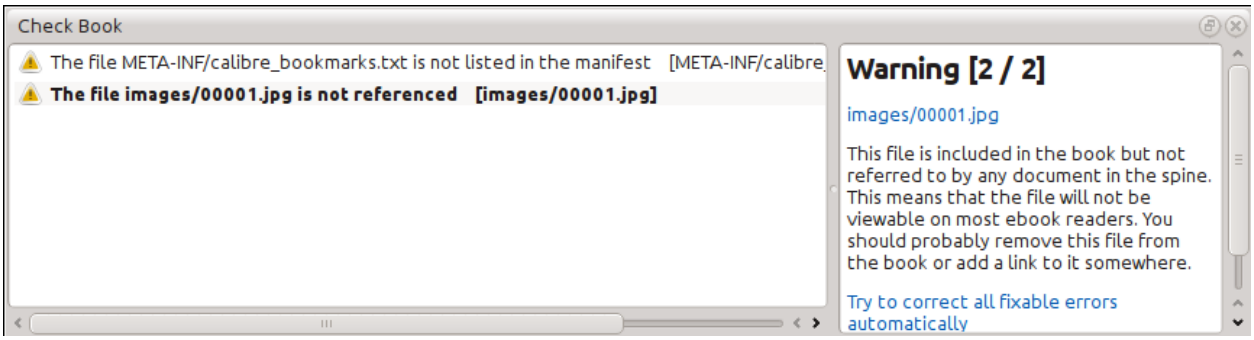
For books that do not have a pre-existing Table of Contents, the tool gives you various options to auto-generate a Table of Contents from the text. You can generate from the headings in the document, from links, from individual files and so on.

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 Adicionando uma 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 Incorporando fontes referenciadas

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 Efetuando subconjunto de fontes embutidas

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 Pontuação inteligente

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 Removendo regras CSS não usadas

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 Embelezando arquivos

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[32].

5.4.11 Inserir índice em linha

Normalmente em ebooks, o índice está separado do texto principal e é acedido, no dispositivo de leitura de ebooks, através de um botão/menu especial de índice. Também pode fazer com que o Calibre gere automaticamente um índice *incorporado* que se torna parte do texto do livro. Será gerado com base no índice atualmente definido.

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 a 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*.

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

5.4.13 Filtrando informações do estilo

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 verificação

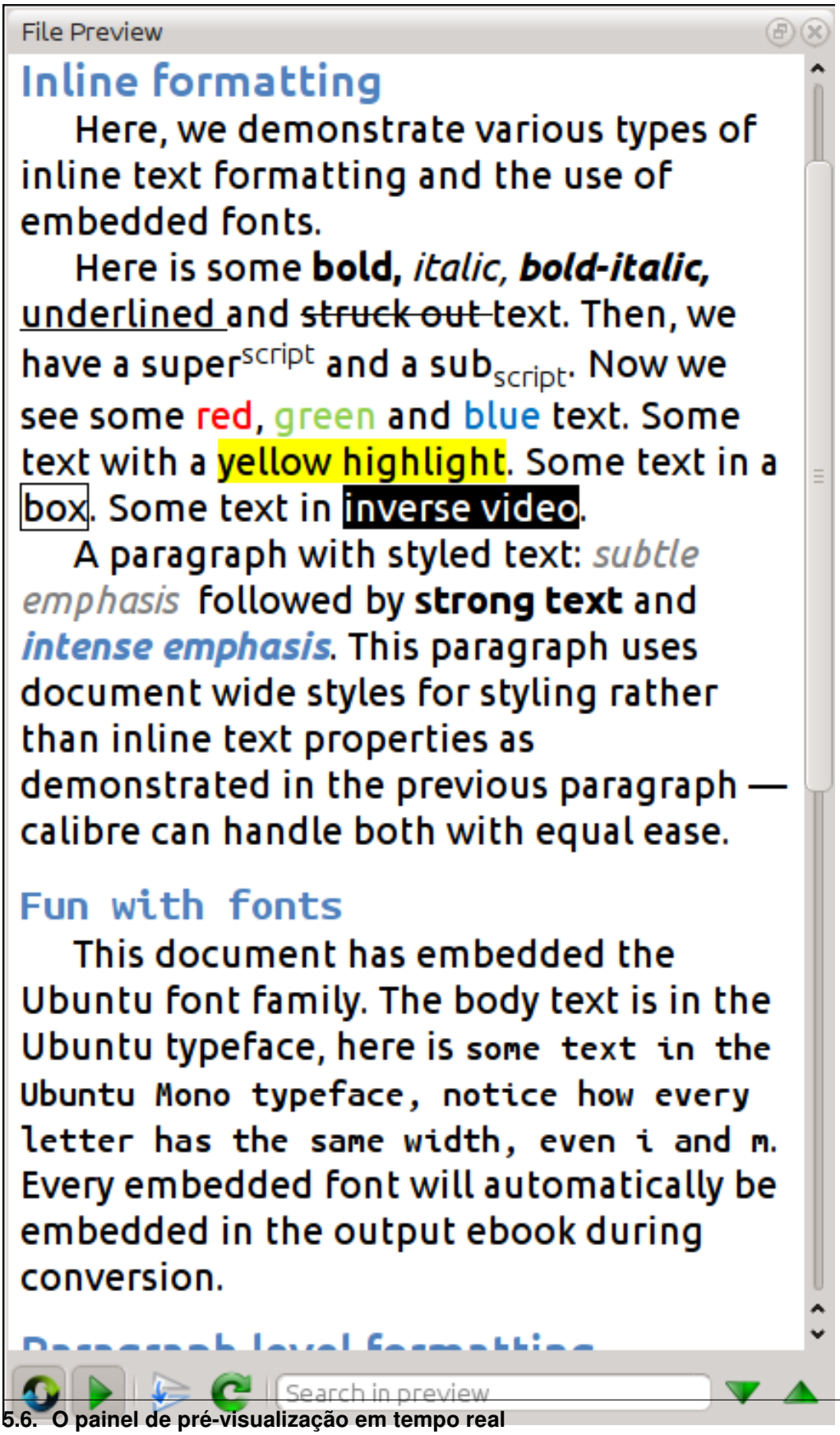
Checkpoints are a way to mark the current state of the book as "special". You can then go on to do whatever changes you want to the book and if you don't like the results, return to the checkpointed state. Checkpoints are automatically created every time you run any of the automated tools described in 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 *Comparando e-books* (página 121) tool – by selecting the checkpoint of interest and clicking the *Compare* button. The *Revert to* button restores the book to the selected checkpoint, undoing all changes since that checkpoint was created.

5.6 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.

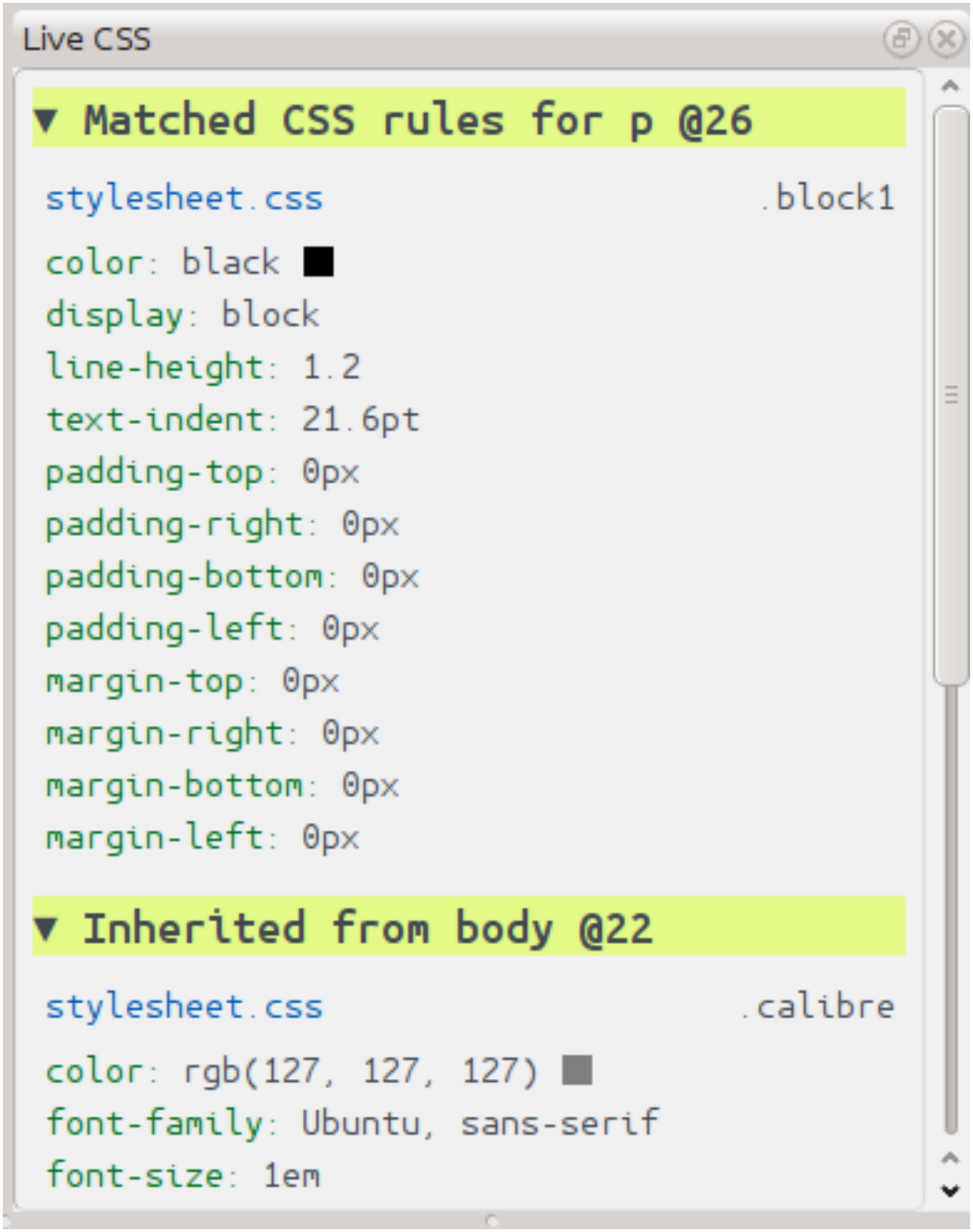
5.6.1 Dividindo arquivos HTML

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 O Painel CSS em tempo real



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

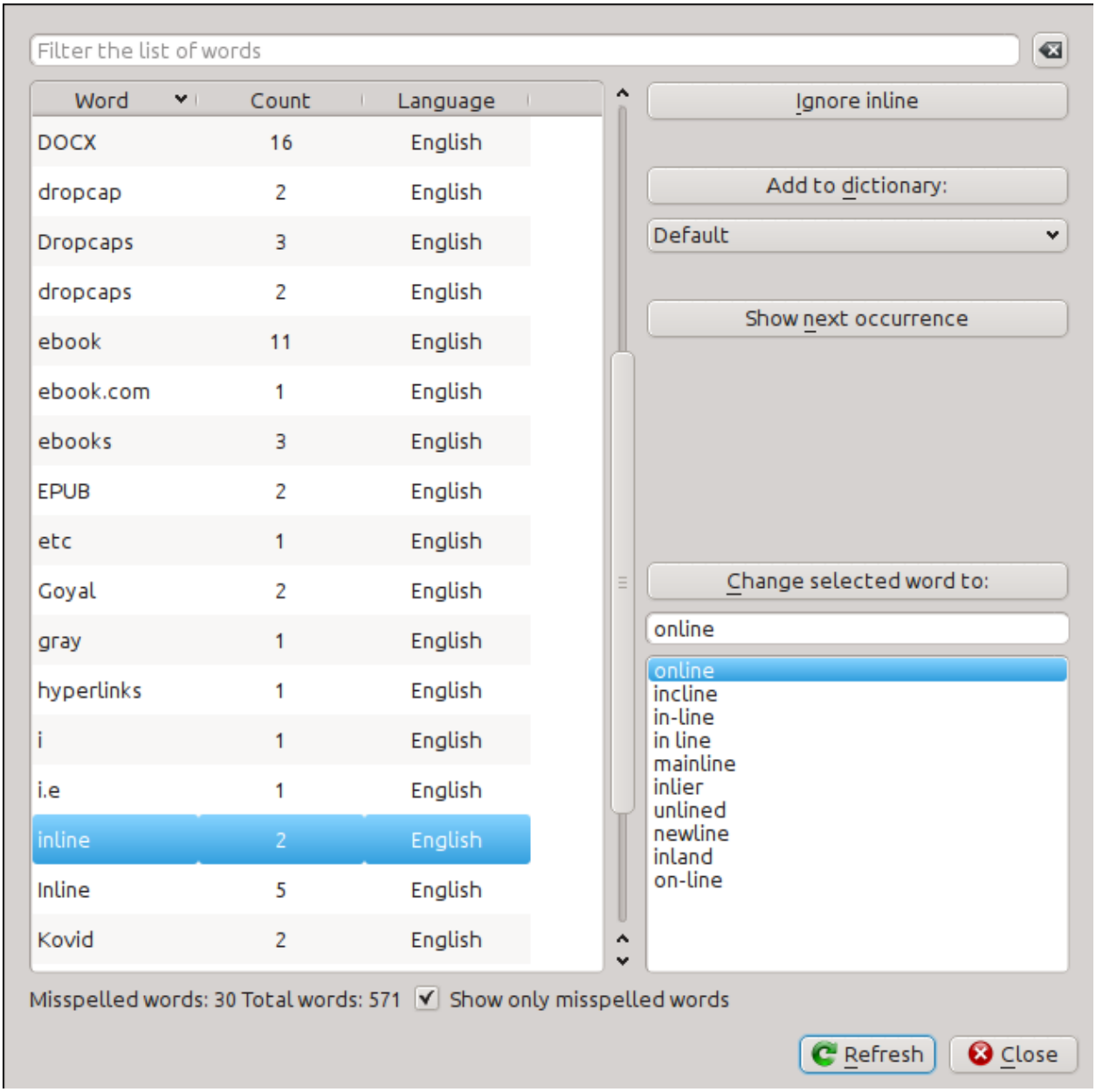
Há mais algumas ferramentas que podem ser úteis durante a edição do livro.

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 Verificando a ortografia das 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 -->
```

Adicionando novos 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[33].

5.8.3 Inserindo 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.

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

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.

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 Checando 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*.

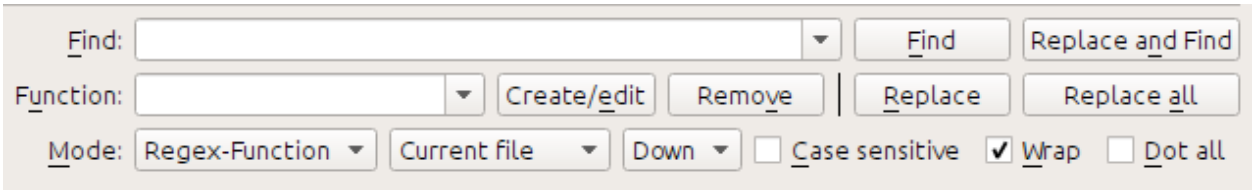
Modo de função para Procurar & Substituir no Editor

The *Search & replace* tool in the editor support a *function mode*. In this mode, you can combine regular expressions (see *Tudo sobre como usar expressões regulares no calibre* (página 215)) 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[34]. This allows you to do lots of things that are not possible with simple templates.

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

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.



Correção automática de caracteres nos 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>
```

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

Sua primeira função personalizada - hifens espertos

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[35]. `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.

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.

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

```
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 106) for details.

Auto criar uma lista de conteúdo

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

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

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

# Ensure that we are called once after the last match is found so we can
# output the ToC
replace.call_after_last_match = True
# Ensure that when running over multiple files, this function is called,
# the files are processed in the order in which they appear in the book
replace.file_order = 'spine'
```

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[36]. 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.

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 211). Useful attributes include, `title`, `authors` (a list of authors) and `language` (the language code).

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

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 = {}

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[37].

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 112). 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

É possível incluir trechos de texto nos campos de entrada na tela *Localizar & substituir*, apesar de que placeholders (usando `Control+J` para navegação) não irão funcionar.

Os trechos embutidos

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]

O primeiro trecho embutido, e o mais simples, é usado para inserir texto de preenchimento em um documento. O texto de preenchimento é tirado de De finibus bonorum et malorum[38] uma obra filosófica da autoria de Cicero (traduzido para o Inglês). Para usá-lo, apenas digite `Lorem` em um arquivo HTML e pressione `Control+J`. O texto será substituído por dois parágrafos do texto de preenchimento.

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.

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

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 (aquele com o próximo número superior).

Inserindo uma tag de link 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 112). 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
```

Existem alguns recursos novos aqui. Primeiro o marcador `$1` ficou mais complexo. Agora ele inclui um *texto padrão* (a palavra `filename`). Se um marcador contiver o texto padrão, o texto padrão é substituído pelo marcador quando o trecho for expandido. Além disso, quando se pula para um marcador com o texto padrão usando `Control+J`, o texto padrão é selecionado. Assim, é possível usar o texto padrão para atuar como um lembrete para incluir partes importantes do modelo. Pode-se especificar o texto padrão para um marcador usando a sintaxe: `${<number>:texto padrão}`.

O outro recurso novo é que o segundo marcador tem um asterisco depois dele (`${2*}`). Isto quer dizer que qualquer texto selecionado antes da expansão do modelo é substituído pelo marcador. Para ver como este recurso funciona, selecione algum texto no editor, pressione `Control+J`, digite `<a` e pressione `Control+J` novamente, o modelo será expandido para

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

Inserindo uma etiqueta HTML de imagem [<i]

Isto é bem semelhante a inserir um link HTML, como se viu acima. Ela permite que se coloque rapidamente uma `<img src="filename" alt="description" />` etiqueta e coloca entre os atributos `src` e `alt`:

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

Inserir uma etiqueta HTML aleatória [<<]

Isto permite que se insira uma etiqueta HTML completa (ou termine um texto previamente selecionado na etiqueta). Para usar, basta digitar `<<` e pressionar `Control+J`. O editor vai expandir para:

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

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

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

Repare que a etiqueta de fechamento foi automaticamente preenchida com `span`. Isto é possível com mais um tipo de placeholder, *mirroring*. Mirroring (espelhamento) significa simplesmente que se se especificar o placeholder de exemplo mais de uma vez no modelo, a segunda e todas as posições subsequentes serão automaticamente preenchidas com o que for especificado na primeira posição, quando se pressionar `Control+J`. A definição para este trecho é:

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

Como é possível perceber, o primeiro marcador ($1) foi especificado duas vezes, a segunda vez na tag de fechamento, que simplesmente copiará o que for digitado na tag de abertura.

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

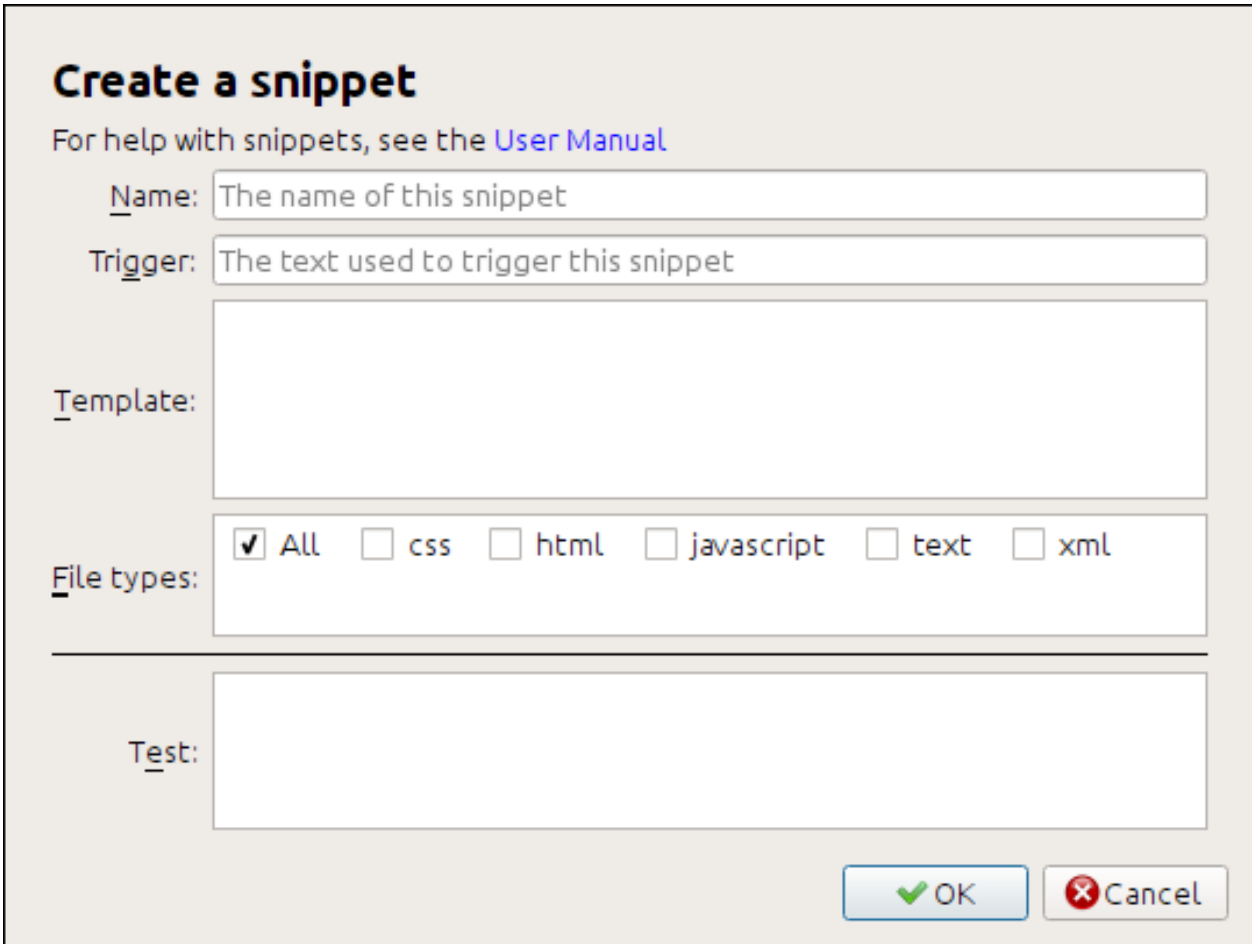
Isto é semelhante ao exemplo de inserção arbitrária de etiqueta acima, exceto que se presume que se deseja especificar uma classe para a etiqueta:

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

Isto irá permitir que se digite o nome da etiqueta primeiro, pressione `Control+J`, digite o nome da classe, pressione `Control+J`, digite o conteúdo da etiqueta e pressione `Control+J` uma última vez para sair da etiqueta. O fechamento da etiqueta irá ser auto-preenchido.

Criando seus próprios trechos

Trechos são bastantes versáteis pois é possível criar um que seja adequado ao seu estilo de edição. Para criar seus trechos, vá para *Editar → Opções → Ajustes do editor → Gerenciar trechos* no editor. Irá aparecer uma tela fácil de itilizar, para ajudar a criar os seus trechos. Clique no botão *Adicionar trecho* e aparecerá uma tela semelhante a:



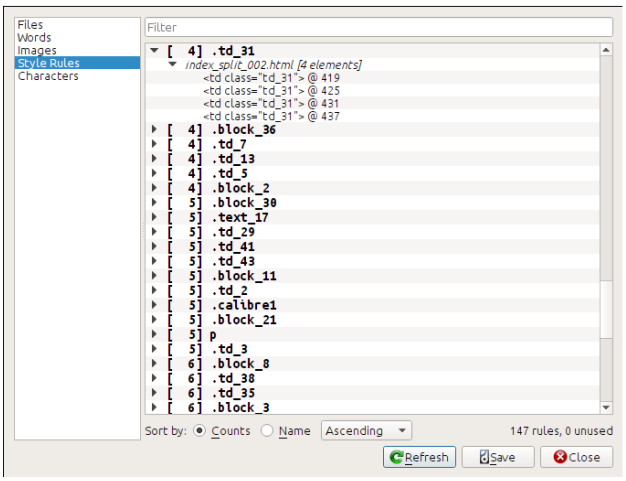
Primeiro dê um nome ao seu trecho, algo descritivo para identificá-lo no futuro. Depois especifique o *gatilho*. Um gatilho é simplesmente o texto que é necessário digitar no editor antes de pressionar `Control+J` para expandir o trecho.

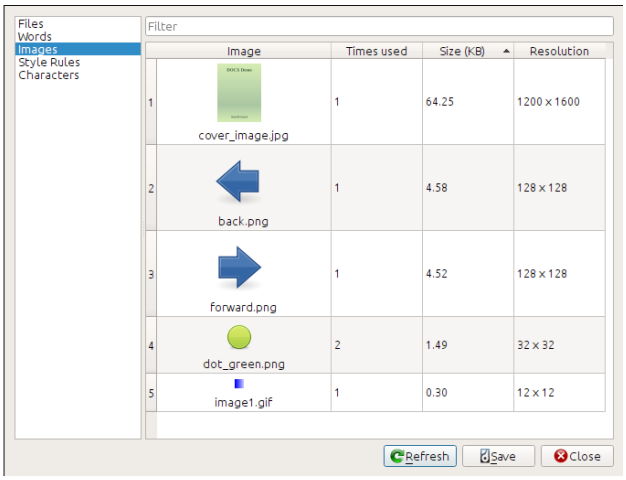
Depois especifique o modelo do trecho. É recomendável começar com um dos exemplos acima e modificá-lo para se adequar às suas necessidades. Finalmente, especifique para quais tipos de arquivos você quer que o trecho esteja ativo. Assim, é possível ter vários trechos com o mesmo texto de gatilho que funcionarão de formas diferentes em tipos de arquivos diferentes.

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 107).

CAPÍTULO 6

O servidor de Conteúdo do calibre

O calibre :guidelabel: "servidor de Conteúdo" te permite acessar suas bibliotecas calibre e ler livros diretamente do navegador do seu tablet ou telefone favoritos. Assim, você não precisará instalar nenhum aplicativo de leitura/gerenciador no seu aparelho. Apenas utilizar o navegador. O servidor baixa e armazena o livro que você está lendo em um cache offline, tornando a leitura possível mesmo sem conexão com a internet.

Conteúdo

- *Acessando o servidor de Conteúdo a partir de outros dispositivos* (página 114)
  - *Acessando o servidor a partir de dispositivos na sua rede local* (página 114)
  - *Aceder ao servidor a partir de qualquer local na Internet* (página 115)
- *A interface do servidor* (página 115)
  - *A lista de livros* (página 116)
  - *O visualizador de livros digitais* (página 116)
- *Browser support* (página 116)
- *Enabling offline support* (página 116)
- *Managing user accounts from the command-line only* (página 116)
- *Integrating the calibre Content server into other servers* (página 117)
  - *Using a full virtual host* (página 117)
  - *Usar um prefixo de URL* (página 118)
- *Criando uma tarefa para o servidor caibre em um sistema Linux moderno* (página 119)

Para iniciar o servidor, clique no botão :guidelabel:"Conectar/compartilhar" e escolha a opção :guidelabel:"Iniciar o servidor de Conteúdo". Você poderá receber uma notificação do firewall ou anti-vírus do seu computador perguntando se ACEITA abrir "calibre.exe". Clique no botão "OK" ou "Permitir". Abra o navegador (de preferência o Chrome ou

Firefox) em seu computador e digite o seguinte endereço:

http://127.0.0.1:8080

This will open a page in the browser showing you your calibre libraries, click on any one and browse the books in it. Click on a book, and it will show you all the metadata about the book, along with buttons to *Read* and *Download* the book. Click the *Read* button to start reading the book.

Nota

O endereço utilizado "http://127.0.0.1:8080" irá funcionar apenas se o computador estiver com o calibre aberto. Para acessar o servidor por outros computadores/telefones/tablets/tec. você precisará seguir outros passos, como descritos na próxima seção.

6.1 Acessando o servidor de Conteúdo a partir de outros dispositivos

Existem dois tipos de acesso remoto a dispositivos que normalmente irá precisar. O primeiro, e mais simples, é utilizando de sua rede doméstica. Se o calibre está rodando em um computador na sua rede doméstica e você também possui seus outros aparelhos conectados nessa mesma rede, então será fácil acessar o servidor nesses outros aparelhos.

6.1.1 Acessando o servidor a partir de dispositivos na sua rede local

Após iniciar o servidor no calibre como infromados anteriormente, clique no botão :guidelabel:'Conectar/compartilhar' novamente. Ao invés da opção :guidelabel:'Iniciar o servidor de Conteúdo', você deve ver a opção :guidelabel:'Parar o servidor de Conteúdo'. À direita dessa opção estará listado um endereço de IP e um número de porta. Esses números devem ser uma série separados por pontos. Exemplo:

```
Stop Content server [192.168.1.5, port 8080]
```

Esses números te informam qual endereço usar para se conectar com o servidor no seu dispositivo. Seguindo o exemplo anterior, temos:

```
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[39].

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[40] 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[41] 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.

[39] https://www.mobileread.com/forums/forumdisplay.php?f=166
[40] https://www.whatismyip.com/
[41] 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 visualizador de livros digitais

You can read any book in your calibre library by simply tapping on it and then tapping the *Read* button. The book viewer is very simple to operate. You can both tap and swipe to turn pages. Swiping up/down skips between chapters. Tapping the top quarter of the screen gets you the detailed controls and viewer preferences.

If you leave the Content server running, you can even open the same book on multiple devices and it will remember your last read position. If it does not you can force a sync by tapping in the top quarter and choosing *Sync*.

6.3 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 ApplicationCache, which was what was used in calibre for offline support. As a result now-a-days sadly, offline mode works only as long as you keep the browser tab open. In addition, in Firefox on Android, you will need to type `about:config` and create a preference called `browser.tabs.useCache` and set it to `true`.

6.5 Managing user accounts from the command-line only

The calibre program has a nice section in *Preferences* to allow you to manage user accounts for the server. However, if you want to run the standalone server and cannot run the main calibre program on the same computer/user account, you can also manage users using just the command-line.

You can manage user accounts using the `--manage-users` option to the standalone `calibre-server` program. Suppose you want to store the user database in the folder `/srv/calibre`, then you create it by running:

```
calibre-server --userdb /srv/calibre/users.sqlite --manage-users
```

Just follow the prompts to create user accounts, set their permission, etc. Once you are done, you can run the server as:

```
calibre-server --userdb /srv/calibre/users.sqlite --enable-auth
```

It will use the user accounts you created in the previous step.

6.6 Integrating the calibre Content server into other servers

Here, we will show you how to integrate the calibre Content server into another server. The most common reason for this is to make use of SSL or to serve the calibre library as part of a larger site. The basic technique is to run the calibre server and setup a reverse proxy to it from the main server.

A reverse proxy is when your normal server accepts incoming requests and passes them onto the calibre server. It then reads the response from the calibre server and forwards it to the client. This means that you can simply run the calibre server as normal without trying to integrate it closely with your main server.

6.6.1 Using a full virtual host

The simplest configuration is to dedicate a full virtual host to the calibre server. In this case, run the calibre server as:

```
calibre-server
```

Now setup the virtual host in your main server, for example, for nginx:

```
http {
    client_max_body_size 64M;  # needed to upload large books
}

server {
    listen [::]:80;
    server_name myserver.example.com;

    location / {
        proxy_pass http://127.0.0.1:8080;
    }
}
```

Or, for Apache:

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

<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/
```

Isso é tudo, você agora poderá acessar o servidor de Conteúdo do calibre na URL "/calibre" no seu servidor principal. As restrições citadas serão aceitas pelo "/calibre" quando o servidor estiver na porta 8080 e, através da opção "–url-prefix", o servidor se ocupará delas de maneira transparente.

> **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 Criando uma tarefa para o servidor caibre em um sistema Linux moderno

You can easily create a service to run calibre at boot on a modern (systemd[42]) 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.

[42] https://www.freedesktop.org/wiki/Software/systemd/

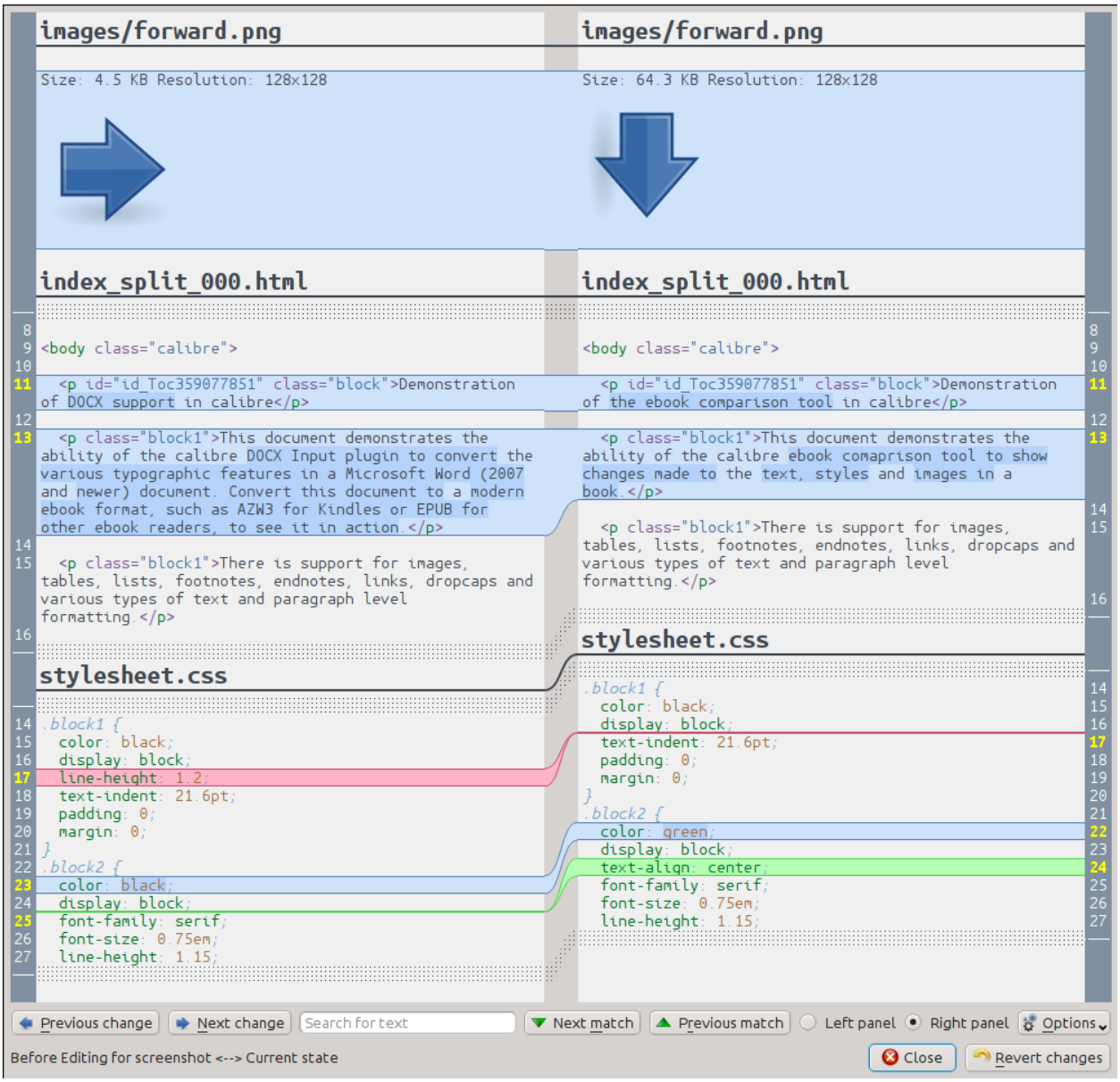
CAPÍTULO 7

Comparando e-books

O calibre possui uma ferramenta integrada de comparação de livros que mostra o que mudou em um e-book (nos formatos EPUB e AZW3) depois de ter sido editado ou convertido.

Para usá-la, abra o e-book na ferramenta para *Editando e-books* (página 81) e então clique *File → Compare to other book* ou use o painel de *Detalhes do livro* (página 21). Se você converter um EPUB para EPUB, o arquivo original do EPUB será salvo como ORIGINAL_EPUB. Simplesmente clique com o botão direito na entrada ORIGINAL_EPUB no painel de Detalhes do Livro e escolha *Compare to EPUB format*.

A ferramenta de comparação que abrirá se parece com a imagem abaixo. Ela mostrará as diferenças no texto, estilos e imagens nos livros escolhidos.



7.1 Entendendo a visualização de comparação.

Como pode ser visto na imagem acima, a tela de comparação mostra a diferença entre os dois livros lado a lado. Somente as diferenças, com algumas linhas de contexto ao redor são apresentadas. Isto torna mais fácil visualizar somente o que mudou dentro de um grande documento como um livro.

Textos adicionados são mostrados com um fundo verde, textos removidos com um fundo vermelho e textos alterados com um fundo azul.

O número das linhas de todo texto modificado é apresentado ao lado, tornando fácil ir a uma alteração específica no 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 quando se compara livros é dizer à ferramenta de comparação para melhorar o texto e os arquivos de estilo antes de calcular as diferenças. Isto pode resultar em mais facilidade e clareza na hora de procurar as diferenças. Para isso, clique no botão *Options* no canto inferior direito e escolha *Beautify files before comparing*. Note que melhorar

o texto pode trazer efeitos indesejados, bem como fazer com que marcações inválidas sejam alteradas para se tornarem válidas. Você também pode alterar o número de linhas de contexto apresentadas em torno das diferenças através do botão guilabel:*Options*.

Você pode procurar por qualquer texto nas diferenças através da barra de busca na parte inferior da tela. Você vai precisar especificar em qual painel será feita a busca, o *Left* ou o *Right*.

7.2 Executando a ferramenta

A ferramenta de comparação é muito útil quando se tem duas versões do mesmo livro e você deseja ver quais as diferenças entre eles. Para este fim, existem muitas maneiras de iniciar a ferramenta.

7.2.1 Comparando dois e-books

Abra o primeiro arquivo na ferramenta *Editando e-books* (página 81). Agora clique em *File → Compare to another book* e escolha o segundo arquivo (este deve estar no mesmo formato que o primeiro). A tela de comparação abrirá com o arquivo sendo editado à direita e o segundo arquivo à esquerda.

7.2.2 Comparando o ORIGINAL_FMT e o FMT.

Quando você faz uma conversão no calibre de um FMT para ele mesmo, o arquivo original é salvo como ORIGINAL_FMT. Você pode ver o que mudou com a conversão, clicando com o botão direito na entrada do ORIGINAL_FMT no painel *Detalhes do livro* (página 21) do calibre e selecionando :guilabel:`Compare to FMT. A tela de comparação vai abrir com o ORIGINAL_FMT à esquerda e o FMT à direita.

7.2.3 Comparando um ponto de verificação para o estado atual do livro enquanto é editado.

A ferramenta *Editando e-books* (página 81) tem uma função bastante útil, chamada *Pontos de verificação* (página 93). Ela permite que você salve o estado atual do livro como um *ponto de verificação*, com ele você pode reverter as alterações efetuadas no documento desde quando foi criado o ponto de verificação. Pontos de verificação são criados automaticamente quando você efetua ações automatizadas no editor. Você também pode ver a lista de pontos de verificação indo em *View → Checkpoints* e então usar o botão *Compare* para comparar o livro no ponto de verificação selecionado com o estado atual. A ferramenta de comparação apresentará o ponto de verificação ao lado esquerdo e o estado atual ao lado direito.

CAPÍTULO 8

Editando metadados de e-books

Conteúdo

- *Editar os metadados de um livro de cada vez* (página 125)
  - *Descarregar metadados* (página 126)
  - *Gerir formatos de ebooks* (página 126)
  - *Tudo sobre capas* (página 126)
- *Editar os metadados de vários livros de uma vez* (página 126)
  - *Buscar e Substituir* (página 126)
  - *Descarregamento de metadados em lote* (página 127)
- *Adding extra data files to a book* (página 127)

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 os metadados de um livro de cada vez

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:

- Você pode clicar no botão entre o título e o autor para inverte-los 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 os metadados de vários livros de uma vez

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 Buscar 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 como usar expressões regulares no calibre* (página 215).

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 como usar expressões regulares no calibre* (página 215) 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.

8.3 Adding extra data files to a book

calibre can store any number of extra data files associated to a book. These can be alternate covers, supplementary material, etc. They cannot be viewed directly or used as conversion sources. Nor are they indexed by the Full text search engine in calibre. To view/add/delete them select the book and right click the *Edit metadata* button and choose *Manage data files*. This will pop-up a window where you can perform operations on these files. Alternately, you can right click the *Add books* button and choose *Add data files to selected book records* to more quickly add data files.

CAPÍTULO 9

Perguntas e Respostas Frequentes

Conteúdo

- *Conversão de formato de e-book* (página 129)
- *Integração de Dispositivo* (página 132)
- *Library management* (página 139)
- *Extras* (página 143)

9.1 Conversão de formato de e-book

Conteúdo

- *Quais os formatos de conversão o calibre de suportar para/de?* (página 130)
- *Qual o melhor formato fonte para converter?* (página 130)
- *Eu converti um arquivo PDF, mas o resultado teve vários problemas?* (página 130)
- *Como converto meu arquivo que contém caracteres que não são da língua inglesa, ou que contém aspas inteligentes?* (página 130)
- *Qual é o problema com o Índice em arquivos MOBI?* (página 130)
- *Como converto uma coleção de arquivos HTML em uma ordem específica?* (página 131)
- *O EPUB que produzi com o calibre não é válido?* (página 132)
- *Como faço para usar alguns dos recursos avançados das ferramentas de conversão?* (página 132)

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, KEPUB, LIT, LRF, MOBI, ODT, PDF, PRC, PDB, PML, RB, RTF, SNB, TCR, TXT, TXTZ

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

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 Qual o melhor formato fonte para converter?

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

9.1.3 Eu converti um arquivo PDF, mas o resultado teve vários problemas?

PDF é um formato terrível para se converter. Para uma lista dos vários problemas que você encontrará convertendo PDF, veja:*Converter documentos PDF* (página 75).

9.1.4 Como converto meu arquivo que contém caracteres que não são da língua inglesa, ou que contém aspas inteligentes?

Há dois aspectos para este 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 Go to 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/>
        .
        .
        .
     </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 produzi 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[43].

9.2 Integração de Dispositivo

Conteúdo

- *Quais dispositivos o calibre suporta?* (página 133)
- *Como posso ajudar para que meu dispositivo seja suportado pelo calibre?* (página 133)
- *Meu dispositivo não é detectado pelo calibre.* (página 133)
- *Meu dispositivo não é padrão ou é incomum. O que posso fazer para me conectar a ele?* (página 134)
- *How do I use calibre with my iPad/iPhone/iPod touch?* (página 134)
- *How do I use calibre with my Android phone/tablet or Kindle Fire?* (página 134)
- *Can I access my calibre books using the web browser in my Kindle or other reading device?* (página 136)
- *I cannot send emails using calibre?* (página 136)
- *My device is getting mounted read-only in Linux, so calibre cannot connect to it?* (página 137)
- *Why does calibre not support collections on the Kindle or shelves on the Nook?* (página 137)
- *I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?* (página 137)
- *Covers for books I send to my e-ink Kindle show up momentarily and then are replaced by a generic cover?* (página 138)

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

- *Covers for books sent to my Kindle ColorSoft and newer do not show up in the book list?* (página 138)
- *As capas dos meus arquivos MOBI deixaram de aparecer no Kindle para PC/Android/iPad, etc.* (página 138)
- *I transferred some books to my Kindle using calibre and they did not show up?* (página 139)

9.2.1 Quais dispositivos 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 e-book que seu dispositivo suporta.
- 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[44].

9.2.3 Meu dispositivo não é detectado 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 2024 Kindle or newer or an Android device, and are on macOS or Linux, read the note under *Usar um cabo USB* (página 135).
- On macOS if you get permission errors when connecting a device to calibre, you can fix that by looking under *System Preferences > Security and Privacy > Privacy > Files and Folders*.
- Make sure you are running the latest version of calibre (currently 8.7.0). The latest version can always be downloaded from the calibre website[45]. 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

[44] https://calibre-ebook.com/pt/bugs
[45] https://calibre-ebook.com/pt/download

- 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[46].

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 Conteúdo do calibre* (página 113), which makes your collection available over the net. First perform the following steps in calibre

- Set the Preferred Output Format in calibre to EPUB (The output format can be set under *Preferences → Interface → Behavior*)
- Set the output profile to iPad (this will work for iPhone/iPods as well), under *Preferences → Conversion → Common options → Page setup*
- Convert the books you want to read on your iDevice to EPUB format by selecting them and clicking the *Convert* button.
- 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.

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 Conteúdo do calibre* (página 113) for details on running the server and finding out the right IP address to use.

You will see a list of books in Safari, tap on any book and you will be given the option to either download it, or read it in the browser itself. If you choose to download it, Safari will ask you if you want to open it with iBooks.

Many reading apps support browsing the calibre library directly via its OPDS support. In such apps you can go to the online catalog screen and add the IP address of the calibre server to browse and download books from your calibre library within the app.

9.2.6 How do I use calibre with my Android phone/tablet or Kindle Fire?

There are two ways that you can connect your Android device to calibre. Using a USB cable – or wirelessly, over the air. The first step to using an Android device is installing an e-book reading application on it. There are many free and paid

[46] https://bugs.launchpad.net/calibre

e-book reading applications for Android: Some examples (in no particular order): FBReader[47], Moon+[48], Mantano[49], Aldiko[50], Kindle[51].

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.

Sem fios

calibre has a builtin web server, the *calibre Content server* (página 113). It makes your calibre collection available over the net. You can browse it on your device using a simple browser or a dedicated application. First perform the following steps in calibre:

- Set the *Preferred Output Format* in calibre to EPUB for normal Android devices or MOBI for Kindles (The output format can be set under *Preferences → Interface → Behavior*)
- Convert the books you want to read on your device to EPUB/MOBI format by selecting them and clicking the *Convert* button.
- Turn on the *Content server* in calibre's preferences and leave calibre running.

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 Conteúdo do calibre* (página 113) for details on running the server and finding out the right IP address to use.

You can now browse your book collection and download books from calibre to your device to open with whatever e-book reading software you have on your Android device.

[47] https://play.google.com/store/apps/details?id=org.geometerplus.zlibrary.ui.android&hl=en
[48] https://play.google.com/store/apps/details?id=com.flyersoft.moonreader&hl=en
[49] https://play.google.com/store/apps/details?id=com.mantano.reader.android.lite&hl=en
[50] https://play.google.com/store/apps/details?id=com.aldiko.android&hl=en
[51] https://play.google.com/store/apps/details?id=com.amazon.kindle&feature=related_apps

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 Conteúdo do calibre* (página 113) for details.

Some devices, like the Kindle (1/2/DX), do not allow you to access port 8080 (the default port on which the content server runs). In that case, change the port in the calibre Preferences to 80. (On some operating systems, you may not be able to run the server on a port number less than 1024 because of security settings. In this case the simplest solution is to adjust your router to forward requests on port 80 to port 8080).

Also some devices do not have browsers advanced enough to run the app-like interface used by the Content server. For such devices, you can simply add `/mobile` to the server URL to get a simplified, non-JavaScript interface.

9.2.8 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[52].
- Go to *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*)
- 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

[52] https://www.gmx.com

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[53].

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

[53] https://www.mobileread.com/forums/showthread.php?t=244202

- 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[54]. 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.

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

Amazon deliberately broke this functionality in their ColorSoft and newer devices in order to discourage you from reading non Amazon books on their devices. See this forum thread[55] for details. The only known workaround is to send the books as "Personal documents" to the Kindle which will fix the covers not showing up but break other features such as Whispersync and the books will show up under "Personal documents" not "Books" on the device. To enable this in calibre go to *Preferences → Output options → MOBI output* and enable the check box that says *Enable sharing of book content*. This will cause all future books sent to the Kindle by calibre to be marked as personal documents.

9.2.14 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

[54] https://www.mobileread.com/forums/showthread.php?t=329945
[55] https://www.mobileread.com/forums/showthread.php?t=364350

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.

9.2.15 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

Conteúdo

- *Onde é que os ficheiros dos livros são guardados?* (página 139)
- *How does calibre manage author names and sorting?* (página 140)
- *Why doesn't calibre let me store books in my own folder structure?* (página 141)
- *Why doesn't calibre have a column for foo?* (página 142)
- *Posso ter uma coluna para mostrar os formatos ou o ISBN?* (página 142)
- *Como mover os meus dados Calibre de um computador para outro?* (página 142)
- *A lista de livros do* Calibre *está em branco!* (página 143)
- *I am getting errors with my calibre library on a networked drive/NAS?* (página 143)

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[56] 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*

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.

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

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[57] to learn how to create your own columns, or read this blog post[58].

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 padrão do calibre* (página 160).

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.

Simply copy the calibre library folder from the old to the new computer. You can find out what the library folder is by clicking the calibre icon in the toolbar. Choose the *Switch/create calibre library* action and you will see the path to the current calibre library.

Now on the new computer, start calibre for the first time. It will run the *Welcome wizard* asking you for the location of the calibre library. Point it to the previously copied folder. If the computer you are transferring to already has a calibre installation, then the *Welcome wizard* won't run. In that case, right-click the calibre icon in the toolbar and point it to the newly copied folder. You will now have two calibre libraries on your computer and you can switch between them by clicking the calibre icon on the toolbar. Transferring your library in this manner preserves all your metadata, tags, custom columns, etc.

[57] https://calibre-ebook.com/pt/demo#tutorials
[58] https://blog.calibre-ebook.com/calibre-custom-columns/

9.3.7 A lista de livros do *Calibre* está em branco!

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 do calibre mudou seu local. Isto pode acontecer se ela estava em um disco externo e a letra do drive para esse disco mudou. Ou se você acidentalmente moveu a pasta. Neste caso, o calibre não encontra sua biblioteca e então inicia com uma biblioteca vazia no lugar. Para reparar isso, clique com o botão direito no ícone do calibre na barra de ferramentas do calibre e selecione Alternar/criar biblioteca. Clique no pequeno ícone azul para selecionar o novo local da sua biblioteca do calibre e clique OK. Se você não sabe o novo local, procure no seu computador pelo arquivo `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[59] for details.

9.4 Extras

Conteúdo

- *Amazon is stopping email delivery of MOBI files?* (página 144)
- *I want calibre to download news from my favorite news website.* (página 145)
- *Porque escolheu o nome Calibre?* (página 145)
- *Why does calibre show only some of my fonts on macOS?* (página 145)
- *O Calibre não arranca em Windows?* (página 145)

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

- *calibre freezes/crashes occasionally?* (página 146)
- *The calibre E-book viewer and Edit book tools do not work on Windows?* (página 146)
- *Using the viewer or doing any conversions results in a permission denied error on Windows* (página 147)
- *calibre is not starting/crashing on macOS?* (página 147)
- *I get only a black or white screen when running the calibre E-book viewer?* (página 147)
- *Eu baixei o instalador, mas ele não está funcionando?* (página 147)
- *Meu programa antivírus afirma que o Calibre é um vírus/espião?* (página 148)
- *Como eu faço backup do Calibre?* (página 148)
- *How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?* (página 148)
- *I am getting a "Permission Denied" error?* (página 149)
- *Can I have the comment metadata show up on my reader?* (página 149)
- *How do I get calibre to use my HTTP proxy?* (página 149)
- *I want some feature added to calibre. What can I do?* (página 149)
- *Why doesn't calibre have an automatic update?* (página 150)
- *Como o calibre é licenciado?* (página 150)
- *How do I run calibre from my USB stick?* (página 150)
- *How do I run parts of calibre like news download and the Content server on my own Linux server?* (página 150)

9.4.1 Amazon is stopping email delivery of MOBI files?

Amazon have announced[60] 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

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.

[60] https://blog.the-ebook-reader.com/2022/05/03/amazon-dropping-mobi-support-on-send-to-kindle-apps/

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 seus sites de notícia favoritos ao calibre* (página 33).

Otherwise, you can request a particular news site by posting in the calibre Recipes forum[61].

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 no errors but the calibre window does not appear, it has probably just appeared off screen. You can gather all windows onto the current screen using one of the techniques described here[62].
- Some software has been known to interfere with calibre, try rebooting in Safe mode and see if it works. A known culprit is the Sunshine[63] screen sharing software.
- 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)
  - 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.

[61] https://www.mobileread.com/forums/forumdisplay.php?f=228
[62] https://www.wikihow.com/Bring-an-Off-Screen-Window-Back-on-Windows
[63] https://github.com/LizardByte/Sunshine

- 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.

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,

[64] https://www.mobileread.com/forums/forumdisplay.php?f=166
[65] https://www.nirsoft.net/utils/shexview.html

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 and Default Folder X.

You can obtain debug output about why calibre is not starting by running *Console.app*. Debug output will be printed to it. If the debug output contains a line that looks like:

```
Qt: internal: -108: Error ATSUMeasureTextImage text/qfontengine_mac.mm
```

then the problem is probably a corrupted font cache. You can clear the cache by following these instructions[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".

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 287) for how to change environment variables.

9.4.11 Eu baixei o instalador, mas ele não está funcionando?

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.

[66] https://doc.qt.io/qt-6/qtwebengine-debugging.html
[67] https://typefaceapp.com/help/troubleshooting/macos-font-cache
[68] https://github.com/kovidgoyal/calibre/releases/latest

- Similarly, if the installer is failing/rolling back and you have Microsoft PowerToys running, quit it.
- If you have installed to a non-standard location, try running the installer as Administrator
- Try rebooting your computer and running a registry cleaner like Wise registry cleaner[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 Meu programa antivírus afirma que o Calibre é um vírus/espião?

A primeira coisa a se certificar é de que você baixou o Calibre site oficial[71]. Certifique-se de que você está clicando nos links de download à esquerda, não nas propagandas à direita. O Calibre é um programa bem popular e pessoas inescrupulosas tentam criar sites falsos oferecendo o download para enganar usuários desavisados.

Se você fez o download do site oficial e ainda assim o seu antivírus afirma que o Calibre está com vírus, seu programa antivírus está errado. Programas antivírus usam heurística, padrões de código que "parecem suspeitos" para detectar vírus. É algo parecido com determinação de perfil racial. O Calibre é um produto totalmente código aberto. Você pode acessar o código fonte (ou contratar alguém para fazê-lo por você) para se certificar de que não contém vírus. Por favor, informe o alarme falso para o fabricante do seu antivírus. Se o antivírus impede o download ou a instalação do Calibre, desabilite-o temporariamente, instale o Calibre e depois habilite o antivírus novamente.

9.4.13 Como eu faço backup do Calibre?

O mais importante é fazer backup da pasta de biblioteca do Calibre, que contém todos os seus livros e metadados. É a pasta que você escolhe como pasta de biblioteca do Calibre quando você o executa pela primeira vez. É possível saber o caminho clicando no ícone do Calibre na barra de ferramentas principal. Você deve fazer backup da pasta inteira, com todos os seus arquivos e subpastas.

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 373). 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

[69] https://www.wisecleaner.com
[70] https://calibre-ebook.com/pt/download_portable
[71] https://calibre-ebook.com/pt/download

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].

[72] https://calibre-ebook.com/pt/get-involved

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 *Crie seus próprios plugins para aumentar as funcionalidades do Calibre* (página 225).

9.4.19 Why doesn't calibre have an automatic update?

Por muitas razões:

- *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 343).
- Existem atualizadores automáticos de terceiros para o Calibre criados por utilizadores Calibre no fórum do Calibre[74].
- Additionally, some third-party updaters such as Norton or Avast may update software behind the user's back. If you find calibre has updated unexpectedly, check for the presence of one.

9.4.20 Como o calibre é 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.

[73] https://calibre-ebook.com/pt/bugs
[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

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-cursor0`, `libxcb-xinerama0`, `libegl1`, `libopengl0`.

You can run the calibre server via the command:

```
/opt/calibre/calibre-server /path/to/the/library/you/want/to/share
```

You can download news and convert it into an e-book with the command:

```
/opt/calibre/ebook-convert "Title of news source.recipe" 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:

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

Remember to read the *Interface da Linha de Comando* (página 303) section of the calibre User Manual to learn more about these, and other commands.

CAPÍTULO 10

Tutoriais

Aqui você encontra tutoriais para que possa começar a usar os recursos mais avançados do Iappl, como XPath e modelos

10.1 Gerenciar subgrupos de livros, por exemplo "gênero"

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.

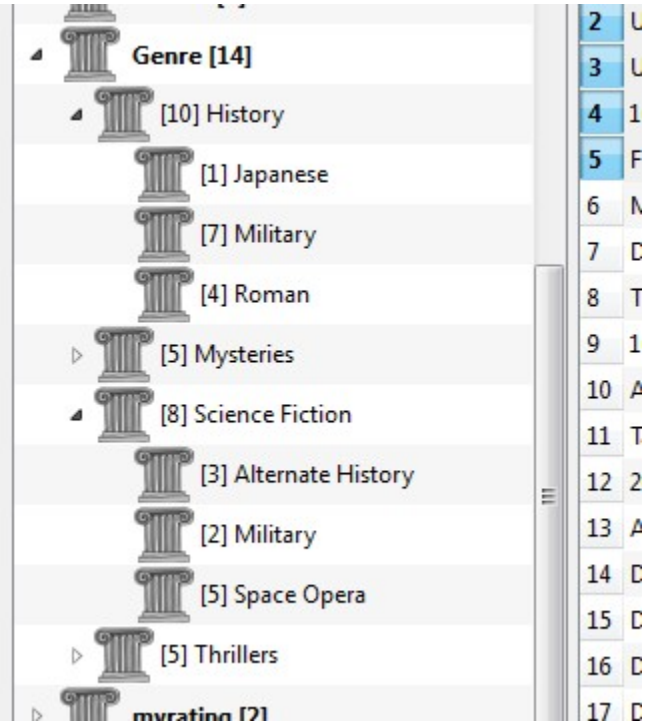
Before going on, please note that we are not talking about folders on the hard disk. Subgroups are not file folders. Books will not be copied anywhere. calibre's library file structure is not affected. Instead, we are presenting a way to organize and display subgroups of books within a calibre library.

- *Ajustes* (página 155)
- *Buscando* (página 157)
- *Restrições* (página 157)
- *Funções úteis de modelos* (página 158)

Os requisitos geralmente fornecidos para subgrupos, como gêneros, 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 ter múltiplos subgrupos (gêneros). A distinção dos subgrupos dos arquivos e pastas físicas
- Subgrupos (gêneros) devem formar uma hierarquia; Subgrupos contém 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,

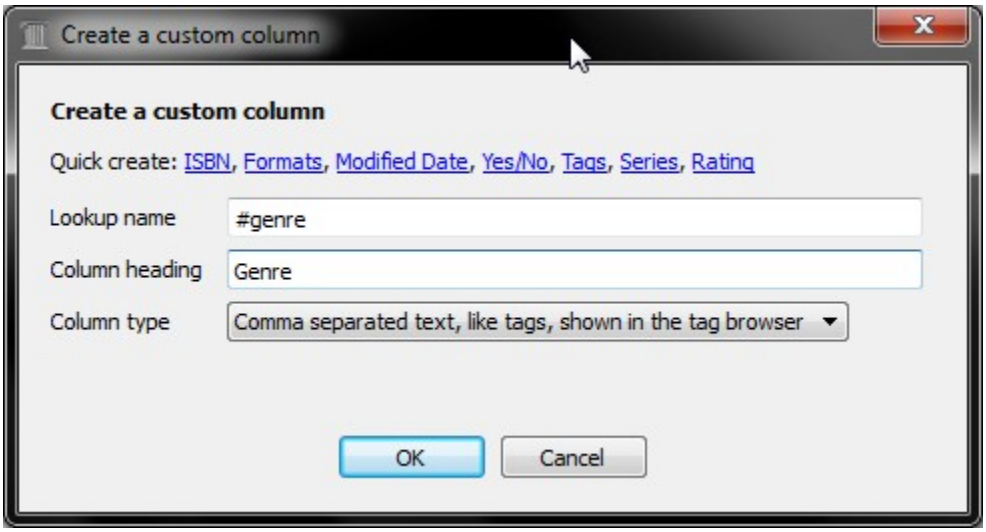
or (doubtfully in this case) both. For example, the books in Winston Churchill's "The Second World War" could be in "History.Military". David Weber's Honor Harrington books could be in "Science Fiction.Military", and for that matter also in "Science Fiction.Space Opera."

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 Ajustes

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.

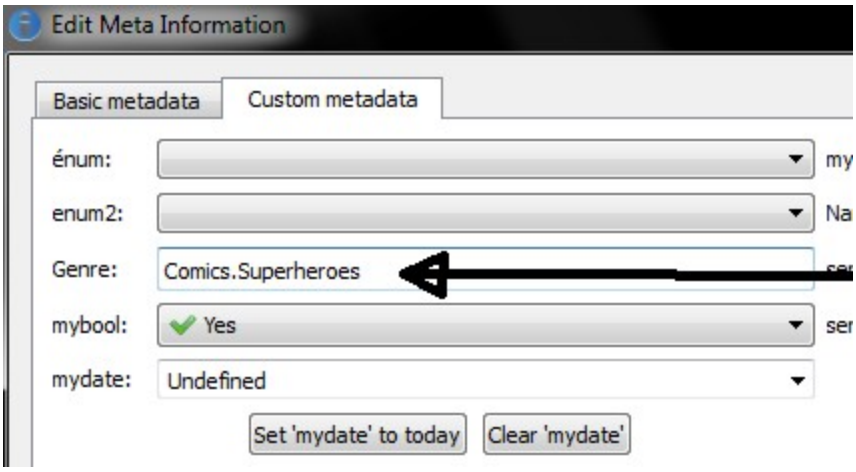
You create the custom column in the usual way, using Preferences -> Add your own columns. This example uses "#genre" as the lookup name and "Genre" as the column heading. It is important that the column type is set to *Comma-separated text, like tags, shown in the Tag browser*.



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

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 Buscando

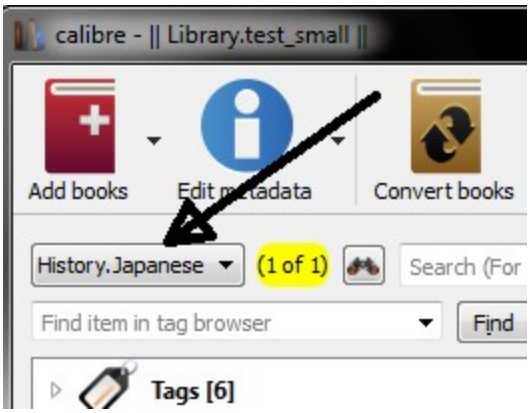
The easiest way to search for genres is using the Tag browser, clicking on the genre you wish to see. Clicking on a genre with children will show you books with that genre and all child genres. However, this might bring up a question. Just because a genre has children doesn't mean that it isn't a genre 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 160) 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.

Conteúdo

- *Seleção pelo nome da tag* (página 159)
- *Selecionar por atributos* (página 159)
- *Selecionar por conteúdo da tag* (página 159)
- *Amostra de E-book* (página 160)
- *Funções internas do XPath* (página 160)

[78] https://en.wikipedia.org/wiki/XPath

10.2.1 Seleção pelo nome da tag

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 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 160) para executar correspondências mais sofisticadas em valores de atributos.

10.2.3 Selecionar por conteúdo da tag

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 Amostra de E-book

```
<html>
    <head>
        <title>A very short e-book</title>
        <meta name="charset" value="utf-8" />
    </head>
    <body>
        <h1 class="bookTitle">A very short e-book</h1>
        <p style="text-align:right">Written by Kovid Goyal</p>
        <div class="introduction">
            <p>A very short e-book to demonstrate the use of XPath.</p>
        </div>

        <h2 class="chapter">Chapter One</h2>
        <p>This is a truly fascinating chapter.</p>

        <h2 class="chapter">Chapter Two</h2>
        <p>A worthy continuation of a fine tradition.</p>
    </body>
</html>
```

10.2.5 Funções internas do XPath

name()
O nome da tag atual.

contains()
`contains(s1, s2)` retorna *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 padrão 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.

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

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
```

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

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

example, to display the ISBN, create the column and enter `{identifiers:select(isbn)}` in the template box. To display a column containing the values of two series custom columns, separated by a comma, use `{#series1:||, }{#series2}`.

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.

10.3.4 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.

Plugboards are quite flexible and can be written in Single Function Mode, Template Program Mode, General Program Mode, or Python Template mode.

When a plugboard might apply (Content server, save to disk, or send to device), calibre searches the defined plugboards to choose the correct one for the given format and device. For example, to find the appropriate plugboard for an EPUB book being sent to an ANDROID device, calibre searches the plugboards using the following search order:

- a plugboard with an exact match on format and device, e.g., `EPUB` and `ANDROID`
- a plugboard with an exact match on format and the special `any device` choice, e.g., `EPUB` and `any device`
- a plugboard with the special `any format` choice and an exact match on device, e.g., `any format` and `ANDROID`
- 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.

10.3.5 Using functions in templates - Single Function Mode

Suppose you want to display the value of a field in upper case when that field is normally in title case. You can do this using *template functions*. For example, to display the title in upper case use the `uppercase` function, as in `{title:uppercase()}`. To display it in title case, use `{title:titlecase()}`.

Functions go into the format part of the template, after the `:` and before the first `|` or the closing `}` if no prefix/suffix is used. If you have both a format and a function reference, the function comes after a second `:`. Functions return the value of the column specified in the template, suitably modified.

The syntax for using functions is one of:

```
{lookup_name:function(arguments)}
{lookup_name:format:function(arguments)}
{lookup_name:function(arguments)|prefix|suffix}
{lookup_name:format:function(arguments)|prefix|suffix}
```

Function names must always be followed by opening and closing parentheses. Some functions require extra values (arguments), and these go inside the parentheses. Arguments are separated by commas. Literal commas (commas as text, not argument separators) must be preceded by a backslash (`\`) . The last (or only) argument cannot contain a textual closing parenthesis.

Functions are evaluated before format specifications and the prefix/suffix. See further down for an example of using both a format and a function.

Important: If you have programming experience, please note that the syntax in *Single Function Mode* is not what you expect. Strings are not quoted and spaces are significant. All arguments are considered to be constants; there are no expressions.

Do not use subtemplates (`{ … }`) as function arguments. Instead, use *Template Program Mode* (página 172) and *General Program Mode* (página 167).

Notes on calling functions in Single Function Mode:

- When functions are used in Single Function Mode, the first parameter, `value`, is automatically replaced by the content of the field specified in the template. For example, when the template `{title:capitalize()}` is processed, the content of the `title` field is passed as the parameter `value` to the capitalize function.
- In the function documentation, the notation `[something]*` means that `something` can be repeated zero or more times. The notation `[something]+` means that the `something` is repeated one or more times (must exist at least one time).
- Some functions use regular expressions. In the template language regular expression matching is case-insensitive.

Functions are documented in *Referência das funções de modelo* (página 181). The documentation tells you what arguments the functions require and what the functions do. For example, here is the documentation of the *ifempty* (página 201) function.

- `ifempty(value, text_if_empty)` – se o `value` não estiver vazio, então retorne esse `value`, caso contrário, retorne `text_if_empty`.

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

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

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

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

- *capitalize* (página 185) `(value)` – devolve o valor de `value` com a primeira letra em maísculas e as restantes em minúsculas.
- *ceiling* (página 185) `(value)` — devolve o menor número inteiro que seja maior que ou igual a `value`.
- *cmp* (página 210) `(value, y, lt, eq, gt)` — faz uma comparação de `value` e `y` depois de converter tudo para números`.
- *contains* (página 200) `(value, pattern, text_if_match, text_if_not_match)` – verifica se o valor é correspondido pela expressão regular `pattern`

- *date_arithmetic* (página 193) `(value, calc_spec, fmt)` – Calcula uma nova data a partir de `value` usando `calc_spec`.
- *encode_for_url* (página 194) `(value, use_plus)` – retorna o `value` codificado para uso em uma URL conforme especificado por `use_plus`. O valor é primeiro codificado em URL. Em seguida, se `use_plus` for `0` então os espaços são substituídos por `'+'` (sinais de mais). Se for `1` então os espaços são substituídos por `%20`.
- *floor* (página 185) `(value)` — devolve o maior número inteiro que seja menor que ou igual a `value`.
- *format_date* (página 187)``(value, format_string) — formata o value, que deve ser uma string de data, utilizando o format_string, retornando uma string. É preferível que a data esteja no formato ISO, pois o uso de outros formatos pode causar erros, já que o valor da data não pode ser determinado de forma inequívoca. Observe que a função format_date_field() é mais rápida e confiável. Os códigos de formatação são: [LIST] []d : o dia como número sem zero à esquerda (1 a 31) []dd : o dia como número com zero à esquerda (01 a 31) []ddd : o nome abreviado do dia, localizado (ex: "Seg" a "Dom") []dddd : o nome completo do dia, localizado (ex: "Segunda-feira" a "Domingo") []M : o mês como número sem zero à esquerda (1 a 12) []MM : o mês como número com zero à esquerda (01 a 12) []MMM : o nome abreviado do mês, localizado (ex: "Jan" a "Dez") []MMMM : o nome completo do mês, localizado (ex: "Janeiro" a "Dezembro") []yy : o ano com dois dígitos (00 a 99) []yyyy : o ano com quatro dígitos []h : as horas sem zero à esquerda (0 a 11 ou 0 a 23, dependendo de am/pm) []hh : as horas com zero à esquerda (00 a 11 ou 00 a 23, dependendo de am/pm) []m : os minutos sem zero à esquerda (0 a 59) []mm : os minutos com zero à esquerda (00 a 59) []s : os segundos sem zero à esquerda (0 a 59) []ss : os segundos com zero à esquerda (00 a 59) []ap : usa o relógio de 12 horas em vez do de 24, com 'ap' substituído pela string localizada minúscula para am ou pm []AP : usa o relógio de 12 horas em vez do de 24, com 'AP' substituído pela string localizada em maiúsculas para AM ou PM []aP : usa o relógio de 12 horas em vez do de 24, com 'aP' substituído pela string localizada para AM ou PM []Ap : usa o relógio de 12 horas em vez do de 24, com 'Ap' substituído pela string localizada para AM ou PM []iso : a data com hora e fuso horário. Deve ser o único formato presente []to_number : converte a data e hora em um número de ponto flutuante (um timestamp) [*]from_number : converte um número de ponto flutuante (um timestamp) em uma data formatada em ISO. Se desejar um formato diferente, adicione a string de formatação desejada após from_number e dois-pontos (:). Exemplo: [CODE] format_date(val, 'from_number:MMM dd yyyy') [/CODE] [/LIST] Você pode obter resultados inesperados se a data que estiver sendo formatada contiver nomes de meses localizados, o que pode ocorrer se o formato da data incluir MMMM. Utilizar format_date_field() evita esse problema.
- *format_duration* (página 188) `(value, template, [largest_unit])` – format the value, a number of seconds, into a string showing weeks, days, hours, minutes, and seconds. If the value is a float then it is rounded to the nearest integer.
- *format_number* (página 190) `(value, template)` – interpreta o `value` como um número e formata esse número usando um modelo de formatação Python como `{0:5.2f}` ou `{0:,d}` ou `${0:5,.2f}`.
- *fractional_part* (página 186) `(value)` — devolve o valor existente depois da casa decimal.
- *human_readable* (página 190) `(value)` – espera que o `value` seja um número e retorna uma string representando esse número em KB, MB, GB, etc.
- *ifempty* (página 201) `(value, text_if_empty)` – se o `value` não estiver vazio, então retorne esse `value`, caso contrário, retorne `text_if_empty`.
- *language_strings* (página 206) `(value, localize)` – retorna os nomes de idioma para os códigos de idioma (veja aqui para nomes e códigos[81]) passados em `value`.
- *list_contains* (página 208) `(value, separator, [ pattern, found_val, ]* not_found_val)` – interpreta o `value` como uma lista de itens separados por `separator`, verificando o `pattern` contra cada item na lista.
- *list_count* (página 196) `(value, separator)` – interpreta o valor como uma lista de itens separados por `separator` e retorna o número de itens na lista.

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

- *list_count_matching* (página 197) `(value, pattern, separator)` – interpreta `value` como uma lista de itens separados por `separator`, retornando o número de itens na lista que correspondem à expressão regular `pattern`.
- *list_item* (página 209) `(value, index, separator)` – interpreta o `value` como uma lista de itens separados por `separator`, retornando o 'index'ésimo item.
- *list_sort* (página 198) `(value, direction, separator)` – retorna `value` classificado usando uma classificação lexical sem distinção entre maiúsculas e minúsculas.
- *lookup* (página 211) `(value, [ pattern, key, ]* else_key)` – Os padrões serão verificados em relação ao `value` em ordem
- *lowercase* (página 185) `(value)` – devolve o valor de `value` em minúsculas.
- *mod* (página 186) `(value, y)` — devolve o `floor` do remanascente de `value / y`.
- *rating_to_stars* (página 190) `(value, use_half_stars)` – Retorna o `value` como uma sequência de caracteres de estrela (★).
- *re* (página 201) `(value, pattern, replacement)` – retorna o `value` após aplicar a expressão regular.
- *re_group* (página 201) `(value, pattern [, template_for_group]*)` – retorna uma string feita aplicando a expressão regular `pattern` a `value` e substituindo cada instância correspondente
- *round* (página 186) `(value)` — devolve o número inteiro mais próximo de `value`.
- *select* (página 209) `(value, key)` – interpreta o `value` como uma lista de itens separados por vírgulas com cada item tendo o formato `id:id_value` (o formato `identifier` do calibre).
- *shorten* (página 201) `(value, left_chars, middle_text, right_chars)` – Retorna uma versão encurtada do `value`
- *str_in_list* (página 209) `(value, separator, [ string, found_val, ]+ not_found_val)` – interpreta o `value` como uma lista de itens separados por `separator` e então compara `string` com cada valor na lista.
- *subitems* (página 199) `(value, start_index, end_index)` – Esta função separa listas de itens hierárquicos semelhantes a tags, como gêneros.
- *sublist* (página 200) `(value, start_index, end_index, separator)` – interpreta o `value` como uma lista de itens separados por `separator`, retornando uma nova lista feita a partir dos itens de `start_index` a `end_index`.
- *substr* (página 202) `(value, start, end)` – retorna o `start`'ésimo até o `end`'ésimo caracteres de `value`
- *swap_around_articles* (página 202) `(value, separator)` – retorna o `value` com os artigos movidos para o final, separados por ponto e vírgula.
- *swap_around_comma* (página 202) `(value)` – dado um `value` do formato `B, A`, retorna `A B`.
- *switch* (página 211) `(value, [patternN, valueN,]+ else_value)` – para cada par `patternN, valueN`, verifica se o `value` corresponde à expressão regular `patternN`
- *test* (página 202) `(value, text_if_not_empty, text_if_empty)` — devolve *text_if_not_empty*` se o valor não estiver vazio, caso contrário devolve ``text_if_empty`.
- *titlecase* (página 185) `(value)` – devolve o valor de `value` no formato caixa de título.
- *transliterate* (página 203) `(value)` – Retorna uma string em um alfabeto latino formada pela aproximação do som das palavras em `value`.
- *uppercase* (página 185) `(value)` – devolve o valor de `value` em maísculas.

Using functions and formatting in the same template

Suppose you have an integer custom column `#myint` that you want displayed with leading zeros, as in `003`. One way to do this is to use a format of `0>3s`. However, by default if a number (integer or float) equals zero then the value is

displayed as the empty string so zero values will produce the empty string, not `000`. If you want to see `000` values then you use both the format string and the `ifempty` function to change the empty value back to a zero. The template would be:

```
{#myint:0>3s:ifempty(0)}
```

Note that you can use the prefix and suffix as well. If you want the number to appear as `[003]` or `[000]`, then use the template:

```
{#myint:0>3s:ifempty(0)|[|]}
```

10.3.6 General Program Mode

General Program Mode (*GPM*) replaces *template expressions* with a program written in the *template language*. The syntax of the language is defined by the following grammar:

```
program         ::= 'program:' expression_list
expression_list ::= top_expression [ ';' top_expression ]*
top_expression  ::= or_expression
or_expression   ::= and_expression [ '||' and_expression ]*
and_expression  ::= not_expression [ '&&' not_expression ]*
not_expression  ::= [ '!' not_expression ]* | concatenate_expr
concatenate_expr::= compare_expr [ '&' compare_expr ]*
compare_expr    ::= add_sub_expr [ compare_op add_sub_expr ]
compare_op      ::= '==' | '!=' | '>=' | '>' | '<=' | '<' |
                    'in' | 'inlist' | 'inlist_field' |
                    '==#' | '!=#' | '>=#' | '>#' | '<=#' | '<#'
add_sub_expr    ::= times_div_expr [ add_sub_op times_div_expr ]*
add_sub_op      ::= '+' | '-'
times_div_expr  ::= unary_op_expr [ times_div_op unary_op_expr ]*
times_div_op    ::= '*' | '/'
unary_op_expr   ::= [ add_sub_op unary_op_expr ]* | expression
expression      ::= identifier | constant | function | assignment | field_reference |
                    if_expr | for_expr | break_expr | continue_expr |
                    '(' expression_list ')' | function_def
field_reference ::= '$' [ '$' ] [ '#' ] identifier
identifier      ::= id_start [ id_rest ]*
id_start        ::= letter | underscore
id_rest         ::= id_start | digit
constant        ::= " string " | ' string ' | number
function        ::= identifier '(' expression_list [ ',' expression_list ]* ')'
function_def    ::= 'def' identifier '(' top_expression [ ',' top_expression ]* ')' ':
↪'
                    expression_list 'fed'
assignment      ::= identifier '=' top_expression
if_expr         ::= 'if' condition 'then' expression_list
                    [ elif_expr ] [ 'else' expression_list ] 'fi'
condition       ::= top_expression
elif_expr       ::= 'elif' condition 'then' expression_list elif_expr | ''
for_expr        ::= for_list | for_range
for_list        ::= 'for' identifier 'in' list_expr
                    [ 'separator' separator_expr ] ':' expression_list 'rof'
for_range       ::= 'for' identifier 'in' range_expr ':' expression_list 'rof'
```

(continua na próxima página)

(continuação da página anterior)

```
range_expr      ::= 'range' '(' [ start_expr ',' ] stop_expr
                    [ ',' step_expr [ ',' limit_expr ] ] ')'
list_expr       ::= top_expression
break_expr      ::= 'break'
continue_expr   ::= 'continue'
separator_expr  ::= top_expression
start_expr      ::= top_expression
stop_expr       ::= top_expression
step_expr       ::= top_expression
limit_expr      ::= top_expression
```

Notes:

- a `top_expression` always has a value. The value of an `expression_list` is the value of the last `top_expression` in the list. For example, the value of the expression list `1;2;'foobar';3` is `3`.
- In a logical context, any non-empty value is `True`
- In a logical context, the empty value is `False`
- Strings and numbers can be used interchangeably. For example, `10` and `'10'` are the same thing.
- Comments are lines starting with a '#' character, possibly preceded by blanks or tabs.

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[82].
- 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[83]. It is an *inclusive or*, returning `'1'` if both the left- and right-hand expressions are True.

Referência de campos

[82] https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html
[83] https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html

A `field_reference` evaluates to the value of the metadata field named by lookup name that follows the `$` or `$$`. Using `$` is equivalent to using the *field* (página 204) function. Using `$$` is equivalent to using the *raw_field* (página 206) function. Examples:

```
* $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
```

For example, this program returns the value of the `series` column if the book has a series, otherwise the value of the `title` column:

```
program: field(if field('series') then 'series' else 'title' fi)
```

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* (página 199) generates a list of numbers. If the result is a valid `lookup name` then the field's value is fetched and the separator specified for that field type is used. If the result isn't a valid lookup name then it is assumed to be a list of values. The list is assumed to be separated by commas unless the optional keyword `separator` is supplied, in which case the list values must be separated by the result of evaluating the `separator_expr`. A separator cannot be used if the list is generated by `range()`. Each value in the list is assigned to the specified variable then the `expression_list` is evaluated. You can use `break` to jump out of the loop, and `continue` to jump to the beginning of the loop for the next iteration.

Example: This template removes the first hierarchical name for each value in Genre (`#genre`), constructing a list with the new names:

```
program:
  new_tags = '';
  for i in '#genre':
    j = re(i, '^.*?\.(.*)$', '\1');
    new_tags = list_union(new_tags, j, ',')
  rof;
  new_tags
```

If the original Genre is *History.Military, Science Fiction.Alternate History, ReadMe* then the template returns *Military, Alternate History, ReadMe*. You could use this template in calibre's *Edit metadata in bulk* → *Search & replace* with *Search for* set to `template` to strip off the first level of the hierarchy and assign the resulting value to Genre.

Note: the last line in the template, `new_tags`, isn't strictly necessary in this case because `for` returns the value of the last top_expression in the expression list. The value of an assignment is the value of its expression, so the value of the `for` statement is what was assigned to `new_tags`.

Function definition

If you have repeated code in a template then you can put that code into a local function. The `def` keyword starts the definition. It is followed by the function name, the argument list, then the code in the function. The function definition ends with the `fed` keyword.

Arguments are positional. When a function is called the supplied arguments are matched left to right against the defined parameters, with the value of the argument assigned to the parameter. It is an error to provide more arguments than defined parameters. Parameters can have default values, such as `a = 25`. If an argument is not supplied for that parameter then the default value is used, otherwise the parameter is set to the empty string.

The `return` statement can be used in a local function.

A function must be defined before it can be used.

Example: This template computes an approximate duration in years, months, and days from a number of days. The function `to_plural()` formats the computed values. Note that the example also uses the `&` operator:

```
program:
      days = 2112;
      years = floor(days/360);
      months = floor(mod(days, 360)/30);
      days = days - ((years*360) + (months * 30));

      def to_plural(v, str):
              if v == 0 then return '' fi;
              return v & ' ' & (if v == 1 then str else str & 's' fi) & ' '
      fed;
```

(continua na próxima página)

(continuação da página anterior)

```
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`, `inlist`, and `inlist_field`. For the `in`, `inlist`, and `inlist_field` operators, the result of the left hand expression is interpreted as a regular expression pattern. They are true if the value of left-hand regular expression matches the value of the right hand expression. The regular expressions are case-insensitive.

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

The numeric comparison operators are `==#`, `!=#`, `<#`, `<=#`, `>#`, `>=#`. The left and right expressions must evaluate to numeric values with two exceptions: both the string value "None" (undefined field) and the empty string evaluate to the value zero.

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 $#genre` returns `'1'` if any of the values retrieved from the book's genres match the regular expression `science`, e.g., *Science*, *History of Science*, *Science Fiction* etc., otherwise `''`.
- `program: '^science$' inlist $#genre` returns `'1'` if any of the book's genres exactly match the regular expression `^science$`, e.g., *Science*, otherwise `''`. The genres *History of Science* and *Science Fiction* don't match.
- `program: 'asimov' inlist $authors` returns `'1'` if any author matches the regular expression `asimov`, e.g., *Asimov, Isaac* or *Isaac Asimov*, otherwise `''`.
- `program: 'asimov' inlist_field 'authors'` returns `'1'` if any author matches the regular expression `asimov`, e.g., *Asimov, Isaac* or *Isaac Asimov*, otherwise `''`.
- `program: 'asimov$' inlist_field 'authors'` returns `'1'` if any author matches the regular expression `asimov$`, e.g., *Isaac Asimov*, otherwise `''`. It doesn't match *Asimov, Isaac* because of the `$` anchor in the regular expression.
- `program: if field('series') != 'foo' then 'bar' else 'mumble' fi` returns `'bar'` if the book's series is not *foo*. Otherwise it returns `'mumble'`.
- `program: if field('series') == 'foo' || field('series') == '1632' then 'yes' else 'no' fi` returns `'yes'` if series is either *foo* or *1632*, otherwise `'no'`.
- `program: if '^(foo|1632)$' in field('series') then 'yes' else 'no' fi` returns `'yes'` if series is either *foo* or *1632*, otherwise `'no'`.
- `program: if 11 > 2 then 'yes' else 'no' fi` returns `'no'` because the `>` operator does a lexical comparison.

- `program: if 11 ># 2 then 'yes' else 'no' fi` returns `'yes'` because the `>#` operator does a numeric comparison.

Functions in General Program Mode

See *Referência das funções de modelo* (página 181) for the list of functions built into the template language.

Notes:

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

10.3.7 More complex programs in template expressions - Template Program Mode

Template Program Mode (*TPM*) is a blend of *General Program Mode* (página 167) and *Single Function Mode* (página 163). *TPM* differs from Single Function Mode in that it permits writing template expressions that refer to other metadata fields, use nested functions, modify variables, and do arithmetic. It differs from *General Program Mode* in that the template is contained between `{` and `}` characters and doesn't begin with the word `program:`. The program portion of the template is a General Program Mode expression list.

Example: assume you want a template to show the series for a book if it has one, otherwise show the value of a custom field #genre. You cannot do this in the *Single Function Mode* (página 163) because you cannot make reference to another metadata field within a template expression. In *TPM* you can, as the following expression demonstrates:

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

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 163).

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

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

- Functions must be given all their arguments. For example, the standard built-in functions must be given the initial parameter `value`.
- The variable `$` is usable as the `value` argument and stands for the value of the field named in the template, `series` in this case.
- white space is ignored and can be used anywhere within the expression.
- constant strings are enclosed in matching quotes, either `'` or `"`.

In *TPM*, using `{` and `}` characters in string literals can lead to errors or unexpected results because they confuse the template processor. It tries to treat them as template expression boundaries, not characters. In some but not all cases you can replace a `{` with `[[` and a `}` with *]]*. Advice: if your program contains `{` and `}` characters then you should use *General Program Mode*.

10.3.8 Python Template Mode

Python Template Mode (PTM) lets you write templates using native Python and the calibre API[84]. 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:

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

```
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.

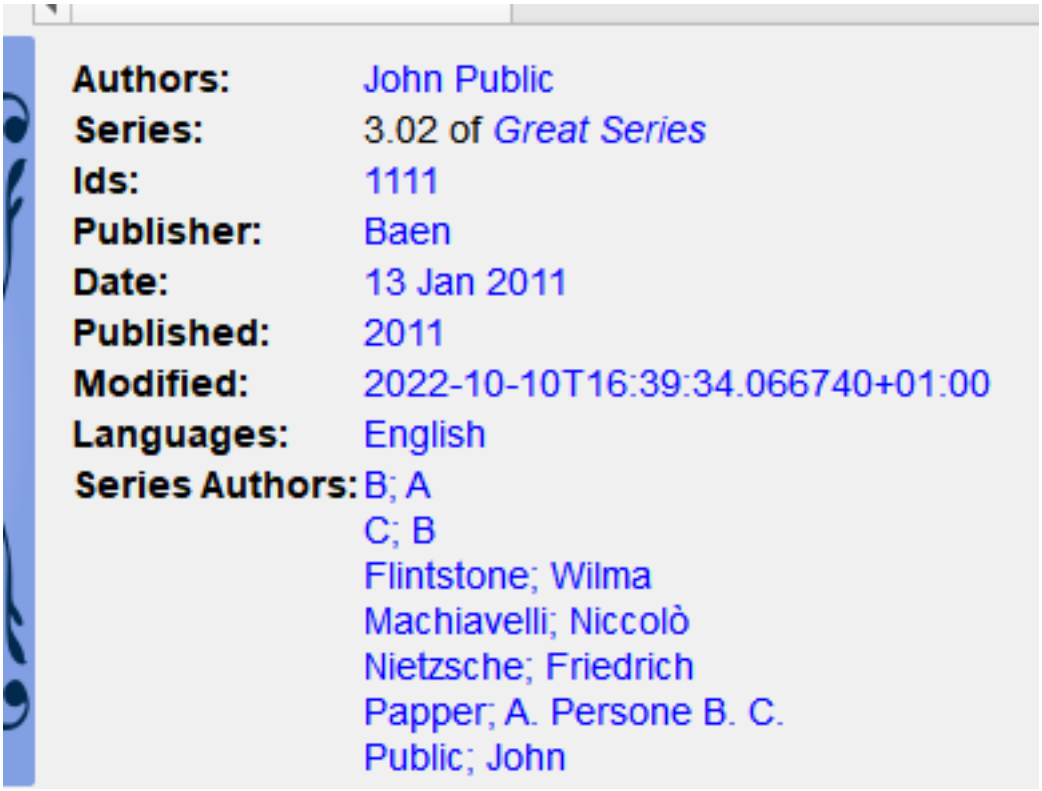
The `context.funcs` attribute allows calling Built-in and User template functions, and Stored GPM/Python templates, so that you can execute them directly in your code. The functions are retrieved using their names. If the name conflicts with a Python keyword, add an underscore to the end of the name. 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('&'))
    # Make a sorted comma-separated string from the result set
    return ', '.join(v.replace(',', ';') for v in sorted(ans))
```

The output in *Book details* looks like this:



10.3.9 Templates and URLs

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

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

Custom column book details search URLs

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

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

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

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

The simplest is a basic template:

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

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

- *make_url* (página 194) `(path, [query_name, query_value]+)` – esta função é a maneira mais fácil de construir uma URL de consulta. Ela usa um `path`, o site e a página que você deseja consultar, e pares `query_name`, `query_value` a partir dos quais a consulta é construída. Em geral, o `query_value` deve ser codificado em URL. Com esta função, ele é sempre codificado e os espaços são sempre substituídos por sinais `'+'`.
- *make_url_extended* (página 195) `(...)` – esta função é similar a *make_url()* (página 194) mas dá a você mais controle sobre os componentes da URL. Os componentes de uma URL são

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

  Veja Uniform Resource Locator[85] na Wikipedia para mais detalhes.

  A função tem duas variantes:

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

  e

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

- *query_string* (página 195) `([query_name, query_value, how_to_encode]+)` – retorna uma string de consulta de URL construída a partir das tríades `query_name, query_value, how_to_encode`. Uma string de consulta é uma série de itens onde cada item se parece com `query_name=query_value` onde `query_value` é codificado por URL, conforme as instruções. Os itens da consulta são separados por `'&'` (e comercial).
- *encode_for_url* (página 194) `(value, use_plus)` – retorna o `value` codificado para uso em uma URL conforme especificado por `use_plus`. O valor é primeiro codificado em URL. Em seguida, se `use_plus` for `0` então os espaços são substituídos por `'+'` (sinais de mais). Se for `1` então os espaços são substituídos por `%20`.

For example, assume you have a custom column *Translators* (`#translators`) where the names are *Last name, First name*. You might need to convert the name to *First name Last name* when creating the URL. You can use the *make_url* (página 194) function to do this:

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

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

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

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

The functions *make_url_extended* (página 195), *query_string* (página 195), and *encode_for_url* (página 194) might be useful depending upon any additional processing complexity.

The calibre URL scheme

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

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

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

  `Library_Name` must be replaced with the name of the calibre library you wish to open. The library name is shown in the title bar of the window. It is a simple name, not the file path to the library. You must spell it as shown in the title bar, including letter case. The character _ (underscore) stands for the current library. If the name contains any

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

spaces or special characters then it must be hex encoded using the *to_hex* (página 196) function, as in the following example:

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

The template generates the URL:

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

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

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

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

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

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

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

It produces the URL:

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

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

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

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

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

The resulting URL is:

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

Here is an example of the same URL built using the :ref:`ff_make_url_extended` function instead of *strcat* (página 202):

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

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

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

An example template is:

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

which produces the URL:

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

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

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

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

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

which produces the URL:

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

10.3.10 Modelos salvos

Both *General Program Mode* (página 167) and *Python Template Mode* (página 172) support saving templates and calling those templates from another template, much like calling stored functions. You save templates using *Preferences → Advanced → Template functions*. More information is provided in that dialog. You call a template the same way you call a function, passing positional arguments if desired. An argument can be any expression. Examples of calling a template, assuming the stored template is named `foo`:

- `foo()` – call the template passing no arguments.
- `foo(a, b)` call the template passing the values of the two variables `a` and `b`.
- `foo(if field('series') then field('series_index') else 0 fi)` – if the book has a `series` then pass the `series_index`, otherwise pass the value `0`.

In GPM you retrieve the arguments passed in the call to the stored template using the `arguments` function. It both declares and initializes local variables, effectively parameters. The variables are positional; they get the value of the parameter given in the call in the same position. If the corresponding parameter is not provided in the call then `arguments` assigns that variable the provided default value. If there is no default value then the variable is set to the empty string. For example, the following `arguments` function declares 2 variables, `key`, `alternate`:

```
arguments(key, alternate='series')
```

Examples, again assuming the stored template is named `foo`:

- `foo('#myseries')` – argument `key` is assigned the value `'myseries'` and the argument `alternate` is assigned the default value `'series'`.
- `foo('series', '#genre')` the variable `key` is assigned the value `'series'` and the variable `alternate` is assigned the value `'#genre'`.
- `foo()` – the variable `key` is assigned the empty string and the variable `alternate` is assigned the value `'series'`.

In PTM the arguments are passed in the `arguments` parameter, which is a list of strings. There isn't any way to specify default values. You must check the length of the `arguments` list to be sure that the number of arguments is what you expect.

An easy way to test stored templates is using the `Template tester` dialog. For ease of access give it a keyboard shortcut in *Preferences → Advanced → Keyboard shortcuts → Template tester*. Giving the `Stored templates` dialog a shortcut will help switching more rapidly between the tester and editing the stored template's source code.

10.3.11 Providing additional information to templates

A developer can choose to pass additional information to the template processor, such as application-specific book metadata or information about what the processor is being asked to do. A template can access this information and use it during the evaluation.

Developer: how to pass additional information

The additional information is a Python dictionary containing pairs `variable_name: variable_value` where the values must be strings. The template can access the dictionary, creating template local variables named `variable_name` containing the value `variable_value`. The user cannot change the name so it is best to use names that won't collide with other template local variables, for example by prefixing the name with an underscore.

This dictionary is passed to the template processor (the `formatter`) using the named parameter `global_vars=your_dict`. The full method signature is:

```
def safe_format(self, fmt, kwargs, error_value, book,
                column_name=None, template_cache=None,
                strip_results=True, template_functions=None,
                global_vars={})
```

Template writer: how to access the additional information

You access the additional information (the `globals` dictionary) in a template using the template function:

```
globals(id[=expression] [, id[=expression]]*)
```

where `id` is any legal variable name. This function checks whether the additional information provided by the developer contains the name. If it does then the function assigns the provided value to a template local variable with that name. If the name is not in the additional information and if an `expression` is provided, the `expression` is evaluated and the result is assigned to the local variable. If neither a value nor an expression is provided, the function assigns the empty string (`''`) to the local variable.

A template can set a value in the `globals` dictionary using the template function:

```
set_globals(id[=expression] [, id[=expression]]*)
```

This function sets the `globals` dictionary key:value pair `id:value` where `value` is the value of the template local variable `id`. If that local variable doesn't exist then `value` is set to the result of evaluating `expression`.

10.3.12 Notes on the difference between modes

The three program modes, *Single Function Mode* (página 163) (SFM), *Template Program Mode* (página 172) (*TPM*), and *General Program Mode* (página 167) (*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 172) section, using `{` and `}` characters in *TPM* string literals can lead to errors or unexpected results because they confuse the template processor. It tries to treat them as template boundaries, not characters. In some but not all cases you can replace a `{` with `[[` and a `}` with *]]*. Generally, if your program contains `{` and `}` characters then you should use *General Program Mode*.

10.3.13 User-defined Python template functions

You can add your own Python functions to the template processor. Such functions can be used in any of the three template programming modes. The functions are added by going to *Preferences* → *Advanced* → *Template functions*. Instructions are shown in that dialog. Note that you can use *Python Templates* for a similar purpose. As calling user-defined functions is faster than calling a Python template, user-defined functions might be more efficient depending on the complexity of what the function or template does.

10.3.14 Special notes for using templates in different contexts

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

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

In icon rules:

- icon rule templates have no book data so field-based functions such as *format_date_field* (página 188), *list_count_field* (página 196), and *check_yes_no* (página 200) won't work.

No servidor de conteúdos:

- Templates have access to the new API but not the old API (LibraryDatabase).
- Because of the above, the following formatter functions are not guaranteed to work in GPM templates (composite columns, icon rules, etc) and should be avoided if you use the content server:
  - *connected_device_name* (página 204)
  - *connected_device_uuid* (página 204)
  - *current_virtual_library_name* (página 204)
  - *is_marked* (página 206)
  - *virtual_libraries* (página 207)

10.3.15 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.

10.3.16 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.

10.3.17 Referência das funções de modelo

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.

- *Alteração de maiúsculas/minúsculas* (página 185)
  - *capitalize* (página 185)
  - *lowercase* (página 185)
  - *titlecase* (página 185)
  - *uppercase* (página 185)
- *Aritmético* (página 185)
  - *add* (página 185)
  - *ceiling* (página 185)
  - *divide* (página 185)
  - *floor* (página 185)
  - *fractional_part* (página 186)
  - *mod* (página 186)
  - *multiply* (página 186)
  - *round* (página 186)
  - *subtract* (página 186)
- *Booleano* (página 186)
  - *and* (página 186)
  - *not* (página 186)
  - *or* (página 187)
- *Formatação de valores* (página 187)
  - *finish_formatting* (página 187)
  - *format_date* (página 187)
  - *format_date_field* (página 188)
  - *format_duration* (página 188)
  - *format_number* (página 190)
  - *human_readable* (página 190)
  - *rating_to_stars* (página 190)
- *Funções de banco de dados* (página 190)
  - *book_count* (página 190)
  - *book_values* (página 191)

  - *extra_file_modtime* (página 191)
  - *extra_file_names* (página 191)
  - *extra_file_size* (página 192)
  - *get_link* (página 192)
  - *get_note* (página 192)
  - *has_extra_files* (página 193)
  - *has_note* (página 193)
- *Funções de data* (página 193)
  - *date_arithmetic* (página 193)
  - *days_between* (página 194)
  - *today* (página 194)
- *Funções do URL* (página 194)
  - *encode_for_url* (página 194)
  - *make_url* (página 194)
  - *make_url_extended* (página 195)
  - *query_string* (página 195)
  - *to_hex* (página 196)
  - *urls_from_identifiers* (página 196)
- *Manipulação de listas* (página 196)
  - *list_count* (página 196)
  - *list_count_field* (página 196)
  - *list_count_matching* (página 197)
  - *list_difference* (página 197)
  - *list_equals* (página 197)
  - *list_intersection* (página 197)
  - *list_join* (página 197)
  - *list_re* (página 198)
  - *list_re_group* (página 198)
  - *list_remove_duplicates* (página 198)
  - *list_sort* (página 198)
  - *list_split* (página 198)
  - *list_union* (página 199)
  - *range* (página 199)
  - *subitems* (página 199)
  - *sublist* (página 200)

- *Manipulação de strings* (página 200)
  - *character* (página 200)
  - *check_yes_no* (página 200)
  - *contains* (página 200)
  - *field_exists* (página 201)
  - *ifempty* (página 201)
  - *re* (página 201)
  - *re_group* (página 201)
  - *shorten* (página 201)
  - *strcat* (página 202)
  - *strcat_max* (página 202)
  - *strlen* (página 202)
  - *substr* (página 202)
  - *swap_around_articles* (página 202)
  - *swap_around_comma* (página 202)
  - *test* (página 202)
  - *transliterate* (página 203)
- *Obter valores dos metadados* (página 203)
  - *annotation_count* (página 203)
  - *approximate_formats* (página 203)
  - *author_links* (página 203)
  - *author_sorts* (página 203)
  - *booksize* (página 204)
  - *connected_device_name* (página 204)
  - *connected_device_uuid* (página 204)
  - *current_library_name* (página 204)
  - *current_library_path* (página 204)
  - *current_virtual_library_name* (página 204)
  - *field* (página 204)
  - *formats_modtimes* (página 205)
  - *formats_path_segments* (página 205)
  - *formats_paths* (página 205)
  - *formats_sizes* (página 206)
  - *has_cover* (página 206)
  - *is_marked* (página 206)

  - *language_codes* (página 206)
  - *language_strings* (página 206)
  - *ondevice* (página 206)
  - *raw_field* (página 206)
  - *raw_list* (página 207)
  - *series_sort* (página 207)
  - *user_categories* (página 207)
  - *virtual_libraries* (página 207)
- *Outro* (página 207)
  - *arguments* (página 207)
  - *assign* (página 207)
  - *globals* (página 208)
  - *is_dark_mode* (página 208)
  - *print* (página 208)
  - *set_globals* (página 208)
- *Pesquisa de listas* (página 208)
  - *identifier_in_list* (página 208)
  - *list_contains* (página 208)
  - *list_item* (página 209)
  - *select* (página 209)
  - *str_in_list* (página 209)
- *Recorrência* (página 209)
  - *eval* (página 209)
  - *template* (página 209)
- *Relacional* (página 210)
  - *cmp* (página 210)
  - *first_matching_cmp* (página 210)
  - *strcmp* (página 210)
  - *strcmpcase* (página 210)
- *Repetir sobre valores* (página 210)
  - *first_non_empty* (página 210)
  - *lookup* (página 211)
  - *switch* (página 211)
  - *switch_if* (página 211)
- *API of the Metadata objects* (página 211)

Alteração de maiúsculas/minúsculas

capitalize

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

capitalize(value) – devolve o valor de value com a primeira letra em maísuculas e as restantes em minúsculas.

lowercase

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

lowercase(value) – devolve o valor de value em minúsculas.

titlecase

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

titlecase(value) – devolve o valor de value no formato caixa de título.

uppercase

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

uppercase(value) – devolve o valor de value em maísculas.

Aritmético

add

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

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

ceiling(value) — devolve o menor número inteiro que seja maior que ou igual a value. Devolve uma exceção se value não for um número.

divide

class calibre.utils.formatter_functions.**BuiltinDivide**

divide(x, y) — devolve x - y. Devolve uma exceção se x ou y não forem números. Esta função pode, muitas vezes, ser substituída pelo operador / .

floor

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

floor(value) — devolve o maior número inteiro que seja menor que ou igual a value. Devolve uma exceção se value não for um número.

fractional_part

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

`fractional_part(value)` — devolve o valor existente depois da casa decimal. Por exemplo, `fractional_part(3.14)` devolve `0.14`. Lança uma exceção caso `value` não seja um número.

mod

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

`mod(value, y)` — devolve o `floor` do remanascente de `value / y`. Devolve uma exceção se `value` ou `y` não forem números.

multiply

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

`multiply(x [,y]*)` – devolve o produto de seus argumentos. Devolve uma exceção se um argumento não for um número. Esta função pode, muitas vezes, ser substituída pelo operador `*`.

round

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

`round(value)` — devolve o número inteiro mais próximo de `value`. Devolve uma exceção se `value` não for um número.

subtract

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

`subtract(x, y)` — devolve `x - y`. Devolve uma exceção se `x` ou `y` não forem números. Esta função pode, muitas vezes, ser substituída pelo operador `-` .

Booleano

and

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

`and(value [, value]*)` – retorna a string `'1'` se todos os valores não estiverem vazios, caso contrário, retorna a string vazia. Você pode ter quantos valores quiser. Na maioria dos casos, você pode usar o operador `&&` em vez desta função. Um motivo para não substituir `and()` por `&&` é quando o curto-circuito pode alterar os resultados por causa de efeitos colaterais. Por exemplo, `and(a='',b=5)` sempre fará ambas as atribuições, onde o operador `&&` não fará a segunda.

not

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

`not(value)` – retorna a string `'1'` se o valor estiver vazio, caso contrário retorna a string vazia. Esta função geralmente pode ser substituída pelo operador unário not (`!`).

or

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

`or(value [, value]*)` – retorna a string `'1'` se algum valor não estiver vazio, caso contrário, retorna a string vazia. Você pode ter quantos valores quiser. Esta função geralmente pode ser substituída pelo operador `||`. Um motivo pelo qual ela não pode ser substituída é se o curto-circuito mudará os resultados por causa de efeitos colaterais.

Formatação de valores

finish_formatting

class calibre.utils.formatter_functions.**BuiltinFinishFormatting**

`finish_formatting(value, format, prefix, suffix)` – aplica `format`, `prefix` e `suffix` ao `value` da mesma forma que é feito em um modelo como `{series_index:05.2f| - |- }`. Esta função é fornecida para facilitar a conversão de modelos complexos de função única ou de modo de programa de modelo para modelos *GPM*. Por exemplo, o programa a seguir produz a mesma saída que o modelo acima :

```
program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")
```

Outro exemplo: para o modelo:

```
{series:re(([^\s])[^\s]+(\s|$),\1)}{series_index:0>2s| - | - }{título}
[/CÓDIGO]
uso:
[CODE]
programa:
strcat(
re(campo('série'), '([^\s])[^\s]+(\s|$)', '\1'),
finish_formatting(campo('índice_de_série'), '0>2s', ' - ', ' - '),
campo('título')
)
```

format_date

class calibre.utils.formatter_functions.**BuiltinFormatDate**

`format_date(value, format_string)` – format the `value`, which must be a date string, using the `format_string`, returning a string. It is best if the date is in ISO format as using other date formats often causes errors because the actual date value cannot be unambiguously determined. Note that the `format_date_field()` function is both faster and more reliable.

The formatting codes are:

- `d` : the day as number without a leading zero (1 to 31)
- `dd` : the day as number with a leading zero (01 to 31)
- `ddd` : the abbreviated localized day name (e.g. "Mon" to "Sun")
- `dddd` : the long localized day name (e.g. "Monday" to "Sunday")
- `M` : the month as number without a leading zero (1 to 12)
- `MM` : the month as number with a leading zero (01 to 12)
- `MMM` : the abbreviated localized month name (e.g. "Jan" to "Dec")

- `MMMM` : the long localized month name (e.g. "January" to "December")
- `yy` : the year as two digit number (00 to 99)
- `yyyy` : the year as four digit number.
- `h` : the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
- `hh` : the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
- `m` : the minutes without a leading 0 (0 to 59)
- `mm` : the minutes with a leading 0 (00 to 59)
- `s` : the seconds without a leading 0 (0 to 59)
- `ss` : the seconds with a leading 0 (00 to 59)
- `ap` : use a 12-hour clock instead of a 24-hour clock, with 'ap' replaced by the lowercase localized string for am or pm
- `AP` : use a 12-hour clock instead of a 24-hour clock, with 'AP' replaced by the uppercase localized string for AM or PM
- `aP` : use a 12-hour clock instead of a 24-hour clock, with 'aP' replaced by the localized string for AM or PM
- `Ap` : use a 12-hour clock instead of a 24-hour clock, with 'Ap' replaced by the localized string for AM or PM
- `iso` : the date with time and timezone. Must be the only format present
- `to_number` : convert the date & time into a floating point number (a *timestamp*)
- `from_number` : convert a floating point number (a *timestamp*) into an ISO-formatted date. If you want a different date format then add the desired formatting string after `from_number` and a colon (`:`). Example:

```
format_date(val, 'from_number:MMM dd yyyy')
```

You might get unexpected results if the date you are formatting contains localized month names, which can happen if you changed the date format to contain `MMMM`. Using `format_date_field()` avoids this problem.

Mostrando a documentação em inglês devido ao erro FFML: Missing [*] in list near text:"[]d : o di" on line 9 in "format_date"

format_date_field

class `calibre.utils.formatter_functions.`**`BuiltinFormatDateField`**

`format_date_field(field_name, format_string)` – formata o valor no campo `field_name`, que deve ser o nome de pesquisa de um campo de data, padrão ou personalizado. Veja *format_date()* (página 187) para os códigos de formatação. Esta função é muito mais rápida que format_date() e deve ser usada quando você estiver formatando o valor em um campo (coluna). Também é mais confiável porque funciona diretamente na data subjacente. Não pode ser usada para datas computadas ou datas em variáveis de string. Exemplos:

```
format_date_field('pubdate', 'yyyy.MM.dd')
format_date_field('#date_read', 'MMM dd, yyyy')
```

format_duration

class `calibre.utils.formatter_functions.`**`BuiltinFormatDuration`**

`format_duration(value, template, [largest_unit])` – format the value, a number of seconds, into a string showing weeks, days, hours, minutes, and seconds. If the value is a float then it is rounded to the nearest integer. You choose how to format the value using a template consisting of value selectors surrounded by `[` and `]` characters. The selectors are:

- `[w]`: weeks
- `[d]`: days
- `[h]`: hours
- `[m]`: minutes
- `[s]`: seconds

You can put arbitrary text between selectors.

The following examples use a duration of 2 days (172,800 seconds) 1 hour (3,600 seconds) and 20 seconds, which totals to 176,420 seconds.

- `format_duration(176420, '[d][h][m][s]')` will return the value `2d 1h 0m 20s`.
- `format_duration(176420, '[h][m][s]')` will return the value `49h 0m 20s`.
- `format_duration(176420, 'Your reading time is [d][h][m][s]')` returns the value `Your reading time is 49h 0m 20s`.
- `format_duration(176420, '[w][d][h][m][s]')` will return the value `2d 1h 0m 20s`. Note that the zero weeks value is not returned.

If you want to see zero values for items such as weeks in the above example, use an uppercase selector. For example, the following uses `'W'` to show zero weeks:

`format_duration(176420, '[W][d][h][m][s]')` returns `0w 2d 1h 0m 20s`.

By default the text following a value is the selector followed by a space. You can change that to whatever text you want. The format for a selector with your text is the selector followed by a colon followed by text segments separated by `'|'` characters. You must include any space characters you want in the output.

You can provide from one to three text segments.

- If you provide one segment, as in `[w: weeks ]` then that segment is used for all values.
- If you provide two segments, as in `[w: weeks | week ]` then the first segment is used for 0 and more than 1. The second segment is used for 1.
- If you provide three segments, as in `[w: weeks | week | weeks ]` then the first segment is used for 0, the second segment is used for 1, and the third segment is used for more than 1.

The second form is equivalent to the third form in many languages.

For example, the selector:

- `[w: weeks | week | weeks ]` produces `'0 weeks '`, `'1 week '`, or `'2 weeks '`.
- `[w: weeks | week ]` produces `'0 weeks '`, `'1 week '`, or `'2 weeks '`.
- `[w: weeks ]` produces `0 weeks '`, `1 weeks '`, or `2 weeks '`.

The optional `largest_unit` parameter specifies the largest of weeks, days, hours, minutes, and seconds that will be produced by the template. It must be one of the value selectors. This can be useful to truncate a value.

`format_duration(176420, '[h][m][s]', 'd')` will return the value `1h 0m 20s` instead of `49h 0m 20s`.

format_number

class calibre.utils.formatter_functions.**BuiltinFormatNumber**

`format_number(value, template)` – interpreta o `value` como um número e formata esse número usando um modelo de formatação Python como `{0:5.2f}` ou `{0:,d}` ou `${0:5,.2f}`. O modelo de formatação deve começar com `{0:` e terminar com `}` como nos exemplos acima. Exceção: você pode deixar de fora o "{0:" inicial e o "}" final se o modelo de formato contiver apenas um formato. Veja a Linguagem de modelo[86] e a documentação do Python[87] para mais exemplos. Retorna a string vazia se a formatação falhar.

human_readable

class calibre.utils.formatter_functions.**BuiltinHumanReadable**

`human_readable(value)` – espera que o `value` seja um número e retorna uma string representando esse número em KB, MB, GB, etc.

rating_to_stars

class calibre.utils.formatter_functions.**BuiltinRatingToStars**

`rating_to_stars(value, use_half_stars)` – Retorna o `value` como uma sequência de caracteres de estrela (★). O valor deve ser um número entre `0` e `5`. Defina `use_half_stars` como `1` se quiser caracteres de meia estrela para números fracionários disponíveis com colunas de classificação personalizadas.

Funções de banco de dados

book_count

class calibre.utils.formatter_functions.**BuiltinBookCount**

`book_count(query, use_vl)` – retorna a contagem de livros encontrados ao pesquisar por `query`. Se `use_vl` for `0` (zero), as bibliotecas virtuais serão ignoradas. Esta função e sua companheira `book_values()` são particularmente úteis em pesquisas de modelo, suportando pesquisas que combinam informações de muitos livros como procurar por séries com apenas um livro. Ela não pode ser usada em colunas compostas, a menos que o ajuste `allow_template_database_functions_in_composites` esteja definido como True. Ela pode ser usada apenas na GUI.

Por exemplo, esta pesquisa de modelo usa esta função e sua companheira para encontrar todas as séries com apenas um livro:

- Defina um modelo armazenado (usando *Preferences → Advanced → Template functions*) chamado `series_only_one_book` (o nome é arbitrário). O modelo é:

```
programa:
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;
```

(continua na próxima página)

[86] https://manual.calibre-ebook.com/template_lang.html
[87] https://docs.python.org/3/library/string.html#formatstrings

(continuação da página anterior)

```
set_globals(vals)
fi;
str_in_list(vals, ',', $series, 1, '')
[/CÓDIGO]
A primeira vez que o modelo é executado (o primeiro livro verificado), ele␣
↪armazena os resultados
das pesquisas no banco de dados em uma variável de modelo ``global`` chamada␣
↪``vals``. Esses
resultados são usados para verificar livros subsequentes sem refazer as pesquisas.
[*] Use o modelo armazenado em uma pesquisa de modelo:
[CODE]
template:"program: series_only_one_book()#@#:n:1"
```

Usar um modelo armazenado em vez de colocar o modelo na pesquisa elimina problemas causados pela exigência de escapar aspas em expressões de pesquisa.

Esta função pode ser usada apenas na GUI e no servidor de conteúdo.

book_values

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

`book_values(column, query, sep, use_vl)` – retorna uma lista dos valores únicos contidos na coluna `column` (um nome de pesquisa), separados por `sep`, nos livros encontrados pela pesquisa por `query`. Se `use_vl` for `0` (zero) então as bibliotecas virtuais serão ignoradas. Esta função e sua companheira `book_count()` são particularmente úteis em pesquisas de modelo, suportando pesquisas que combinam informações de muitos livros, como procurar por séries com apenas um livro. Ela não pode ser usada em colunas compostas, a menos que o ajuste `allow_template_database_functions_in_composites` esteja definido como True. Esta função pode ser usada apenas na GUI e no servidor de conteúdo.

extra_file_modtime

class calibre.utils.formatter_functions.**BuiltinExtraFileModtime**

`extra_file_modtime(file_name, format_string)` – retorna o tempo de modificação do arquivo extra `file_name` na pasta `data/` do livro se ele existir, caso contrário `-1`. O modtime é formatado de acordo com `format_string` (veja *format_date()* (página 187) para detalhes). Se `format_string` for a string vazia, retorna o modtime como o número de segundos em ponto flutuante desde a época. Veja também as funções *has_extra_files()* (página 193), *extra_file_names()* (página 191) e *extra_file_size()* (página 192). A época depende do SO. Esta função pode ser usada apenas na GUI e no servidor de conteúdo.

extra_file_names

class calibre.utils.formatter_functions.**BuiltinExtraFileNames**

`extra_file_names(sep [, pattern])` – retorna uma lista separada por `sep` de arquivos extras na pasta `data/` do livro. Se o parâmetro opcional `pattern`, uma expressão regular, for fornecido, a lista será filtrada para arquivos que correspondem a `pattern`. A correspondência de padrão não diferencia maiúsculas de minúsculas. Veja também as funções *has_extra_files()* (página 193), *extra_file_modtime()* (página 191) e *extra_file_size()* (página 192). Esta função pode ser usada apenas na GUI e no servidor de conteúdo.

extra_file_size

class calibre.utils.formatter_functions.**BuiltinExtraFileSize**

`extra_file_size(file_name)` – retorna o tamanho em bytes do arquivo extra `file_name` na pasta `data/` do livro se existir, caso contrário `-1`. Veja também as funções *has_extra_files()* (página 193), *extra_file_names()* (página 191) e *extra_file_modtime()* (página 191). Esta função pode ser usada apenas na GUI e no servidor de conteúdo.

get_link

class calibre.utils.formatter_functions.**BuiltinGetLink**

`get_link(field_name, field_value)` – busca o link para o campo `field_name` com valor `field_value`. Se não houver um link anexado, retorna a string vazia. Exemplos:

- O seguinte retorna o link anexado à tag `Fiction`:

```
get_link('tags', 'Fiction')
```

- Este modelo faz uma lista dos links para todas as tags associadas a um livro no formato `value:link, ...`:

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

Esta função funciona apenas na GUI e no servidor de conteúdo.

get_note

class calibre.utils.formatter_functions.**BuiltinGetNote**

`get_note(field_name, field_value, plain_text)` – busca a nota para o campo `field_name` com o valor `field_value`. Se `plain_text` estiver vazio, retorna o HTML da nota incluindo imagens. Se `plain_text` for `1` (ou `'1'`), retorna o texto simples da nota. Se a nota não existir, retorna a string vazia em ambos os casos. Exemplo:

- Retorna o HTML da nota anexada à tag *Fiction*:

```
program:
get_note('tags', 'Fiction', '')
```

- Retorna o texto simples da nota anexada ao autor *Isaac Asimov*:

```
program:
get_note('authors', 'Isaac Asimov', 1)
```

Esta função funciona apenas na GUI e no servidor de conteúdo.

has_extra_files

class calibre.utils.formatter_functions.**BuiltinHasExtraFiles**

`has_extra_files([pattern])` – retorna a contagem de arquivos extras, caso contrário '' (a string vazia). Se o parâmetro opcional `pattern` (uma expressão regular) for fornecido, a lista será filtrada para arquivos que correspondem a `pattern` antes que os arquivos sejam contados. A correspondência de padrão não diferencia maiúsculas de minúsculas. Veja também as funções *extra_file_names()* (página 191), *extra_file_size()* (página 192) e *extra_file_modtime()* (página 191). Esta função pode ser usada apenas na GUI e no servidor de conteúdo.

has_note

class calibre.utils.formatter_functions.**BuiltinHasNote**

`has_note(field_name, field_value)`. Verifique se um campo tem uma nota. Esta função tem duas variantes:

- se `field_value` não for `''` (a string vazia) retorne `'1'` se o valor `field_value` no campo `field_name` tiver uma nota, caso contrário `''`.

  Exemplo: `has_note('tags', 'Fiction')` retorna `'1'` se a tag `fiction` tiver uma nota anexada, caso contrário `''`.

- Se `field_value` for `''` então retorne uma lista de valores em `field_name` que tenham uma nota. Se nenhum item no campo tiver uma nota, retorne `''`. Esta variante é útil para mostrar ícones de coluna se qualquer valor no campo tiver uma nota, em vez de um valor específico.

  Exemplo: `has_note('authors', '')` retorna uma lista de autores que têm notas, ou `''` se nenhum autor tiver uma nota.

Você pode testar se todos os valores em `field_name` têm uma nota comparando o comprimento da lista do valor de retorno desta função com o comprimento da lista dos valores em `field_name`. Exemplo:

```
list_count(has_note('authors', ''), '&') ==# list_count_field('authors')
```

Esta função funciona apenas na GUI e no servidor de conteúdo.

Funções de data

date_arithmetic

class calibre.utils.formatter_functions.**BuiltinDateArithmetic**

`date_arithmetic(value, calc_spec, fmt)` – Calcula uma nova data a partir de `value` usando `calc_spec`. Retorna a nova data formatada de acordo com o opcional `fmt`: se não for fornecido, o resultado estará no formato ISO. `calc_spec` é uma string formada pela concatenação de pares de `vW` (`valueWhat`) onde `v` é um número possivelmente negativo e W é uma das seguintes letras:

- `s`: adicione `v` segundos a `date`
- `m`: adicione `v` minutos a `date`
- `h`: adicione `v` horas a `date`
- `d`: adicione `v` dias a `date`
- `w`: adicione `v` semanas a `date`
- `y`: adicione `v` anos a `date`, onde um ano tem 365 dias.

Exemplo: `'1s3d-1m'` adicionará 1 segundo, adicionará 3 dias e subtrairá 1 minuto de `date`.

days_between

class calibre.utils.formatter_functions.**BuiltinDaysBetween**

`days_between(date1, date2)` – retorna o número de dias entre `date1` e `date2`. O número é positivo se `date1` for maior que `date2`, caso contrário, negativo. Se `date1` ou `date2` não forem datas, a função retorna a string vazia.

today

class calibre.utils.formatter_functions.**BuiltinToday**

`today()` – retorna uma string de data+hora para hoje (agora). Este valor é projetado para uso em `format_date` ou `days_between`, mas pode ser manipulado como qualquer outra string. A data está no formato ISO[88] data/hora.

Funções do URL

encode_for_url

class calibre.utils.formatter_functions.**BuiltinEncodeForURL**

`encode_for_url(value, use_plus)` – retorna o `value` codificado para uso em uma URL conforme especificado por `use_plus`. O valor é primeiro codificado em URL. Em seguida, se `use_plus` for `0` então os espaços são substituídos por `'+'` (sinais de mais). Se for `1` então os espaços são substituídos por `%20`.

Se você não quiser que o valor seja codificado, mas que os espaços sejam substituídos, use a função:ref:*re() <ff_re>*, como em `re($series, ' ', '%20')`

Veja também as funções *make_url()* (página 194), *make_url_extended()* (página 195) e *query_string()* (página 195).

make_url

class calibre.utils.formatter_functions.**BuiltinMakeUrl**

`make_url(path, [query_name, query_value]+)` – esta função é a maneira mais fácil de construir uma URL de consulta. Ela usa um `path`, o site e a página que você deseja consultar, e pares `query_name`, `query_value` a partir dos quais a consulta é construída. Em geral, o `query_value` deve ser codificado em URL. Com esta função, ele é sempre codificado e os espaços são sempre substituídos por sinais `'+'`.

Pelo menos um par `query_name, query_value` deve ser fornecido.

Exemplo: construindo uma URL de pesquisa da Wikipedia para o autor *Niccolò Machiavelli*:

```
make_url('https://en.wikipedia.org/w/index.php', 'search', 'Niccolò Machiavelli')
```

retorna

```
https://en.wikipedia.org/w/index.php?search=Niccol%C3%B2+Machiavelli
```

Se você estiver escrevendo um modelo de URL de detalhes do livro de coluna personalizado, use `$item_name` ou `field('item_name')` para obter o valor do campo que foi clicado. Exemplo: se *Niccolò Machiavelli* foi clicado, então você pode construir a URL usando:

```
make_url('https://en.wikipedia.org/w/index.php', 'search', $item_name)
```

Veja também as funções *make_url_extended()* (página 195), *query_string()* (página 195) e *encode_for_url()* (página 194).

[88] https://en.wikipedia.org/wiki/ISO_8601

make_url_extended

class calibre.utils.formatter_functions.**BuiltinMakeUrlExtended**

`make_url_extended(...)` – esta função é similar a *make_url()* (página 194) mas dá a você mais controle sobre os componentes da URL. Os componentes de uma URL são

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

Veja Uniform Resource Locator[89] na Wikipedia para mais detalhes.

A função tem duas variantes:

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

e

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

Esta função retorna uma URL construída a partir de `scheme`, `authority`, `path`, e `query_string` ou uma query string construída a partir dos pares de argumentos de consulta. A `authority` pode estar vazia, o que é o caso de URLs de esquema `calibre`. Você deve fornecer uma `query_string` ou pelo menos um par `query_name, query_value`. Se você fornecer `query_string` e ela estiver vazia, a URL resultante não terá uma seção de query string.

Exemplo 1: construindo uma URL de pesquisa da Wikipedia para o autor *Niccolò Machiavelli*:

```
make_url_extended('https', 'en.wikipedia.org', '/w/index.php', 'search', 'Niccolò␣
↪Machiavelli')
```

retorna

```
https://en.wikipedia.org/w/index.php?search=Niccol%C3%B2+Machiavelli
```

Veja a função *query_string()* (página 195) para um exemplo usando `make_url_extended()` com uma `query_string`.

Se você estiver escrevendo um modelo de URL de detalhes do livro de coluna personalizado, use `$item_name` ou `field('item_name')` para obter o valor do campo que foi clicado. Exemplo: se *Niccolò Machiavelli* foi clicado, então você pode construir a URL usando:

```
make_url_extended('https', 'en.wikipedia.org', '/w/index.php', 'search', $item_name')
```

Veja também as funções *make_url()* (página 194), *query_string()* (página 195) e *encode_for_url()* (página 194).

query_string

class calibre.utils.formatter_functions.**BuiltinQueryString**

`query_string([query_name, query_value, how_to_encode]+)`– retorna uma string de consulta de URL construída a partir das tríades `query_name, query_value, how_to_encode`. Uma string de consulta é uma série de itens onde cada item se parece com `query_name=query_value` onde `query_value` é codificado por URL, conforme as instruções. Os itens da consulta são separados por `'&'` (e comercial).

Se `how_to_encode` for `0`, então `query_value` será codificado e os espaços serão substituídos por sinais `'+'` (mais). Se `how_to_encode` for `1`, então `query_value` será codificado com espaços substituídos por `%20`. Se `how_to_encode` for `2`, `query_value` será retornado inalterado; nenhuma codificação será feita e os espaços não serão substituídos. Se você quiser que `query_value` não seja codificado, mas que os espaços sejam substituídos, use a função *re()* (página 201) , como em `re($series, ' ', '%20')`

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

Use esta função se precisar de controle específico sobre como as partes da string de consulta são construídas. Você poderia então usar a string de consulta resultante em *make_url_extended()* (página 195), como em [CÓDIGO] make_url_extended('https', 'your_host', 'your_path', query_string('encoded', 'Hendrik Bäßler', 0, 'unencoded', 'Hendrik Bäßler', 2)) [/CÓDIGO] fornecendo a você [CÓDIGO] https://your_host/your_path?encoded=Hendrik+B%C3%A4%C3%9Fler&unencoded=Hendrik Bäßler [/CÓDIGO]

Você deve ter pelo menos uma tríade `query_name, query_value, how_to_encode`, mas pode ter quantas desejar.

O valor retornado é uma string de consulta de URL com todos os itens especificados, por exemplo: `name1=val1[&nameN=valN]*`. Observe que o separador `'?'` *path / query string* não está incluído no resultado retornado.

Se você estiver escrevendo um modelo de URL de detalhes do livro de colunas personalizado, use `$item_name` ou `field('item_name')` para obter o valor não codificado do campo que foi clicado. Você também tem `item_value_quoted`, onde o valor já está codificado com sinais de mais substituindo espaços, e `item_value_no_plus`, onde o valor já está codificado com `%20` substituindo espaços.

Veja também as funções *make_url()* (página 194), *make_url_extended()* (página 195) e *encode_for_url()* (página 194).

to_hex

class `calibre.utils.formatter_functions.`**`BuiltinToHex`**

`to_hex(val)` – retorna a string `val` codificada em hexadecimal. Isso é útil ao construir URLs de calibre.

urls_from_identifiers

class `calibre.utils.formatter_functions.`**`BuiltinUrlsFromIdentifiers`**

`urls_from_identifiers(identifiers, sort_results)` – dada uma lista separada por vírgulas de `identificadores`, onde um `identificador` é um par de valores separados por dois pontos (`id_name:id_value`), retorna uma lista separada por vírgulas de URLs HTML geradas a partir dos identificadores. A lista não é classificada se `sort_results` for `0` (caractere ou número), caso contrário, é classificada alfabeticamente pelo nome do identificador. Os URLs são gerados da mesma forma que a coluna de identificadores incorporada quando mostrada em Detalhes do livro.

Manipulação de listas

list_count

class `calibre.utils.formatter_functions.`**`BuiltinCount`**

`list_count(value, separator)` – interpreta o valor como uma lista de itens separados por `separator` e retorna o número de itens na lista. A maioria das listas usa uma vírgula como separador, mas `authors` usa um e comercial (&).

Exemplos: `{tags:list_count(,)}`, `{authors:list_count(&)}`.

Aliases: `count()`, `list_count()`

list_count_field

class `calibre.utils.formatter_functions.`**`BuiltinFieldListCount`**

`list_count_field(lookup_name)` – retorna a contagem de itens no campo com o nome de lookup `lookup_name`. O campo deve ser multivalorado, como `authors` ou `tags`, caso contrário, a função gera um erro. Esta função é muito mais rápida que `list_count()` porque opera diretamente em dados do calibre sem convertê-los em uma string primeiro. Exemplo: `list_count_field('tags')`.

list_count_matching

class calibre.utils.formatter_functions.**BuiltinListCountMatching**

`list_count_matching(value, pattern, separator)` – interpreta `value` como uma lista de itens separados por `separator`, retornando o número de itens na lista que correspondem à expressão regular `pattern`.

Aliases: `list_count_matching()`, `count_matching()`

list_difference

class calibre.utils.formatter_functions.**BuiltinListDifference**

`list_difference(list1, list2, separator)` – retorna uma lista feita removendo de `list1` qualquer item encontrado em `list2` usando uma comparação sem distinção entre maiúsculas e minúsculas. Os itens em `list1` e `list2` são separados por `separator`, assim como os itens na lista retornada.

list_equals

class calibre.utils.formatter_functions.**BuiltinListEquals**

`list_equals(list1, sep1, list2, sep2, yes_val, no_val)` – retorna `yes_val` se `list1` e `list2` contiverem os mesmos itens, caso contrário, retorna `no_val`. Os itens são determinados pela divisão de cada lista usando o caractere separador apropriado (`sep1` ou `sep2`). A ordem dos itens nas listas não é relevante. A comparação não diferencia maiúsculas de minúsculas.

list_intersection

class calibre.utils.formatter_functions.**BuiltinListIntersection**

`list_intersection(list1, list2, separator)` – retorna uma lista feita removendo de `list1` qualquer item não encontrado em `list2`, usando uma comparação sem distinção entre maiúsculas e minúsculas. Os itens em `list1` e `list2` são separados por `separator`, assim como os itens na lista retornada.

list_join

class calibre.utils.formatter_functions.**BuiltinListJoin**

`list_join(with_separator, list1, separator1 [, list2, separator2]*)` – retorna uma lista feita juntando os itens nas listas de origem (`list1` etc) usando `with_separator` entre os itens na lista de resultados. Itens em cada origem `list[123...]` são separados pelo `separator[123...]` associado. Uma lista pode conter zero valores. Pode ser um campo como `publisher` que é de valor único, efetivamente uma lista de um item. Duplicatas são removidas usando uma comparação que não diferencia maiúsculas de minúsculas. Itens são retornados na ordem em que aparecem nas listas de origem. Se itens em listas diferem apenas em letras maiúsculas, então a última é usada. Todos os separadores podem ter mais de um caractere.

Exemplo:

```
programa:
list_join('#@#', $authors, '&', $tags, ',')
```

Você pode usar `list_join` nos resultados de chamadas anteriores para `list_join` da seguinte forma:

```
programa:
a = list_join('#@#', $authors, '&', $tags, ',');
b = list_join('#@#', a, '#@#', $#genre, ',', $#people, '&', 'some value', ',')
```

Você pode usar expressões para gerar uma lista. Por exemplo, suponha que você queira itens para `authors` e `#genre`, mas com o gênero alterado para a palavra "Genre: " seguido pela primeira letra do gênero, ou seja, o gênero "Fiction" se torna "Genre: F". O seguinte fará isso:

```
program:
    list_join('#@#', $authors, '&', list_re($#genre, ',', '^(.).*$', 'Genre: \1'),  ',
↪')
```

list_re

class `calibre.utils.formatter_functions.`**`BuiltinListRe`**

`list_re(src_list, separator, include_re, opt_replace)` – Construa uma lista primeiro separando `src_list` em itens usando o caractere `separator`. Para cada item na lista, verifique se ele corresponde a `include_re`. Se corresponder, adicione-o à lista a ser retornada. Se `opt_replace` não for a string vazia, então aplique a substituição antes de adicionar o item à lista retornada.

list_re_group

class `calibre.utils.formatter_functions.`**`BuiltinListReGroup`**

`list_re_group(src_list, separator, include_re, search_re [,template_for_group]*)` – Como list_re, exceto que as substituições não são opcionais. Ele usa `re_group(item, search_re, template ...)` ao fazer as substituições.

list_remove_duplicates

class `calibre.utils.formatter_functions.`**`BuiltinListRemoveDuplicates`**

`list_remove_duplicates(list, separator)` – retorna uma lista feita removendo itens duplicados em `list`. Se os itens diferem apenas em caso, o último é retornado. Os itens em `list` são separados por `separator`, assim como os itens na lista retornada.

list_sort

class `calibre.utils.formatter_functions.`**`BuiltinListSort`**

`list_sort(value, direction, separator)` – retorna `value` classificado usando uma classificação lexical sem distinção entre maiúsculas e minúsculas. Se `direction` for zero (número ou caractere), `value` é classificado em ordem crescente, caso contrário, em ordem decrescente. Os itens da lista são separados por `separator`, assim como os itens na lista retornada.

list_split

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:

```
list_split('one:two:foo', ':', 'var')
```

is equivalent to:

```
var_0 = 'one'
var_1 = 'two'
var_2 = 'foo'
```

list_union

class calibre.utils.formatter_functions.**BuiltinListUnion**

`list_union(list1, list2, separator)` – retorna uma lista feita pela fusão dos itens em `list1` e `list2`, removendo itens duplicados usando uma comparação que não diferencia maiúsculas de minúsculas. Se os itens diferem em maiúsculas e minúsculas, o de `list1` é usado. Os itens em `list1` e `list2` são separados por `separator`, assim como os itens na lista retornada. Aliases: `merge_lists()`, `list_union()`

range

class calibre.utils.formatter_functions.**BuiltinRange**

`range(start, stop, step, limit)` – retorna uma lista de números gerados por looping sobre o intervalo especificado pelos parâmetros `start`, `stop` e `step`, com um comprimento máximo de `limit`. O primeiro valor produzido é 'start'. Valores subsequentes `next_v = current_v + step`. O loop continua enquanto `next_v < stop` assumindo que `step` é positivo, caso contrário, enquanto `next_v > stop`. Uma lista vazia é produzida se `start` falhar no teste: `start >= stop` se `step` for positivo. O `limit` define o comprimento máximo da lista e tem um padrão de 1000. Os parâmetros `start`, `step` e `limit` são opcionais. Chamar `range()` com um argumento especifica `stop`. Dois argumentos especificam `start` e `stop`. Três argumentos especificam `start`, `stop` e `step`. Quatro argumentos especificam `start`, `stop`, `step` e `limit`.

Exemplos:

```
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) -> erro(limite excedido)
```

subitems

class calibre.utils.formatter_functions.**BuiltinSubitems**

`subitems(value, start_index, end_index)` – Esta função separa listas de itens hierárquicos semelhantes a tags, como gêneros. Ela interpreta o `value` como uma lista separada por vírgulas de itens semelhantes a tags, onde cada item é uma lista separada por ponto. Ela retorna uma nova lista feita extraindo de cada item os componentes de `start_index` a `end_index`, e então mesclando os resultados novamente. Duplicatas são removidas. O primeiro subitem em uma lista separada por ponto tem um índice de zero. Se um índice for negativo, ele conta a partir do fim da lista. Como um caso especial, um `end_index` de zero é assumido como o comprimento da lista.

Exemplos:

- Supondo uma coluna #genre contendo "A.B.C":
  - `{#genre:subitems(0,1)}` retorna "A"
  - `{#genre:subitems(0,2)}` retorna "A.B"
  - `{#genre:subitems(1,0)}` retorna "B.C"

- Supondo uma coluna #genre contendo "A.B.C, D.E":
  - `{#genre:subitems(0,1)}` retorna "A, D"
  - `{#genre:subitems(0,2)}` retorna "A.B, D.E"

sublist

class `calibre.utils.formatter_functions.`**BuiltinSublist**

`sublist(value, start_index, end_index, separator)` – interpreta o `value` como uma lista de itens separados por `separator`, retornando uma nova lista feita a partir dos itens de `start_index` a `end_index`. O primeiro item é o número zero. Se um índice for negativo, então ele conta a partir do fim da lista. Como um caso especial, um end_index de zero é assumido como o comprimento da lista.

Exemplos assumindo que a coluna tags (que é separada por vírgulas) contém "A, B, C":

- `{tags:sublist(0,1,\,)}` retorna "A"
- `{tags:sublist(-1,0,\,)}` retorna "C"
- `{tags:sublist(0,-1,\,)}` retorna "A, B"

Manipulação de strings

character

class `calibre.utils.formatter_functions.`**BuiltinCharacter**

`character(character_name)` – retorna o caractere nomeado por character_name. Por exemplo, `character('newline')` retorna um caractere de nova linha (`'\n'`). Os nomes de caracteres suportados são `newline`, `return`, `tab` e `backslash`. Esta função é usada para colocar esses caracteres na saída de modelos.

check_yes_no

class `calibre.utils.formatter_functions.`**BuiltinCheckYesNo**

`check_yes_no(field_name, is_undefined, is_false, is_true)` – verifica se o valor do campo sim/não nomeado pelo nome de pesquisa `field_name` é um dos valores especificados pelos parâmetros, retornando `'Sim'` se uma correspondência for encontrada caso contrário, retornando a string vazia. Defina o parâmetro `is_undefined`, `is_false` ou `is_true` como 1 (o número) para verificar essa condição, caso contrário, defina-o como 0.

Exemplo: `check_yes_no("#bool", 1, 0, 1)` retorna `'Sim'` se o campo sim/não `#bool` for Verdadeiro ou indefinido (nem Verdadeiro nem Falso).

Mais de um dos seguintes: `is_undefined`, `is_false` ou `is_true` pode ser definido como 1.

contains

class `calibre.utils.formatter_functions.`**BuiltinContains**

`contains(value, pattern, text_if_match, text_if_not_match)` – verifica se o valor é correspondido pela expressão regular `pattern`. Retorna `text_if_match` se o padrão corresponde ao valor, caso contrário retorna `text_if_not_match`.

field_exists

class calibre.utils.formatter_functions.**BuiltinFieldExists**

`field_exists(lookup_name)` – verifica se um campo (coluna) com o nome de pesquisa `lookup_name` existe, retornando `'1'` se sim e a string vazia se não.

ifempty

class calibre.utils.formatter_functions.**BuiltinIfempty**

`ifempty(value, text_if_empty)` – se o `value` não estiver vazio, então retorne esse `value`, caso contrário, retorne `text_if_empty`.

re

class calibre.utils.formatter_functions.**BuiltinRe**

`re(value, pattern, replacement)` – retorna o `value` após aplicar a expressão regular. Todas as instâncias de `pattern` no valor são substituídas por `replacement`. A linguagem de modelo usa expressões Python regulares[90] que não diferenciam maiúsculas de minúsculas.

re_group

class calibre.utils.formatter_functions.**BuiltinReGroup**

`re_group(value, pattern [, template_for_group]*)` – retorna uma string feita aplicando a expressão regular `pattern` a `value` e substituindo cada instância correspondente pelo valor retornado pelo template correspondente. No Template Program Mode[91], como para as funções `template` e `eval`, você usa `[[` para `{` e `]]` para `}`.

O exemplo a seguir procura uma série com mais de uma palavra e coloca a primeira palavra em maiúscula:

```
program: re_group(field('series'), "(\S* )(.*)", "{$:uppercase()}", "{$}")'}
```

shorten

class calibre.utils.formatter_functions.**BuiltinShorten**

`shorten(value, left_chars, middle_text, right_chars)` – Return a shortened version of the `value`, consisting of `left_chars` characters from the beginning of the `value`, followed by `middle_text`, followed by `right_chars` characters from the end of the `value`. `left_chars` and `right_chars` must be non-negative integers.

Example: assume you want to display the title with a length of at most 15 characters in length. One template that does this is `{title:shorten(9,-,5)}`. For a book with the title *Ancient English Laws inthe Times of Ivanhoe* the result will be *Ancient E-anhoe*: the first 9 characters of the title, a `-`, then the last 5 characters. If the value's length is less than `left chars` + `right chars` + the length of `middle text` then the value will be returned unchanged. For example, the title *TheDome* would not be changed.

Mostrando a documentação em inglês devido ao erro FFML: Missing closing "`" for CODE_TEXT on line 12 in "shorten"

[90] https://docs.python.org/3/library/re.html
[91] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

strcat

class calibre.utils.formatter_functions.**BuiltinStrcat**

`strcat(a [, b]*)` — devolve uma sequência formada pela concatenação dos seus argumentos. Pode receber qualquer número de argumentos. Esta função pode, muitas vezes, ser substituída pelo operador `&`.

strcat_max

class calibre.utils.formatter_functions.**BuiltinStrcatMax**

`strcat_max(max, string1 [, prefix2, string2]*)` – Retorna uma string formada pela concatenação dos argumentos. O valor retornado é inicializado como `string1`. Strings feitas de pares `prefix, string` são adicionadas ao final do valor, desde que o comprimento da string resultante seja menor que `max`. Prefixos podem estar vazios. Retorna `string1` mesmo se `string1` for maior que `max`. Você pode passar quantos pares `prefix, string` desejar.

strlen

class calibre.utils.formatter_functions.**BuiltinStrlen**

`strlen(value)` – Devolve o comprimento da sequência `value`.

substr

class calibre.utils.formatter_functions.**BuiltinSubstr**

`substr(value, start, end)` – retorna o `start`'ésimo até o `end`'ésimo caracteres de `value`. O primeiro caractere em `value` é o zero'ésimo caractere. Se `end` for negativo, então indica que muitos caracteres estão sendo contados da direita. Se `end` for zero, então indica o último caractere. Por exemplo, `substr('12345', 1, 0)` retorna `'2345'`, e `substr('12345', 1, -1)` retorna `'234'`.

swap_around_articles

class calibre.utils.formatter_functions.**BuiltinSwapAroundArticles**

`swap_around_articles(value, separator)` – retorna o `value` com os artigos movidos para o final, separados por ponto e vírgula. O `value` pode ser uma lista, nesse caso, cada item da lista é processado. Se o `value` for uma lista, você deve fornecer o `separator`. Se nenhum `separator` for fornecido ou o separador for uma string vazia, o `value` será tratado como um valor único, não uma lista. Os `articles` são aqueles usados pelo calibre para gerar o `title_sort`.

swap_around_comma

class calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

`swap_around_comma(value)` – dado um `value` do formato `B, A`, retorna `A B`. Isso é mais útil para converter nomes no formato LN, FN para FN LN. Se não houver uma vírgula no `value`, a função retorna o valor inalterado.

test

class calibre.utils.formatter_functions.**BuiltinTest**

`test(value, text_if_not_empty, text_if_empty)` — devolve *text_if_not_empty*` se o valor não estiver vazio, caso contrário devolve ``text_if_empty`.

transliterate

class calibre.utils.formatter_functions.**BuiltinTransliterate**

`transliterate(value)` – Retorna uma string em um alfabeto latino formada pela aproximação do som das palavras em `value`. Por exemplo, se `value` for `Фёдор Михайлович Достоевский` esta função retorna `Fiodor Mikhailovich Dostoievskii`.

Obter valores dos metadados

annotation_count

class calibre.utils.formatter_functions.**BuiltinAnnotationCount**

`annotation_count()` – retorna o número total de anotações de todos os tipos anexadas ao livro atual. Esta função funciona apenas na GUI e no servidor de conteúdo.

approximate_formats

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

`approximate_formats()` – retorna uma lista separada por vírgulas de formatos associados ao livro. Como a lista vem do banco de dados do calibre em vez do sistema de arquivos, não há garantia de que a lista esteja correta, embora provavelmente esteja. Observe que os nomes dos formatos resultantes são sempre maiúsculos, como no EPUB. A função `approximate_formats()` é muito mais rápida do que as funções `formats_...`.

Esta função funciona apenas na GUI. Se você quiser usar esses valores em modelos de salvar em disco ou enviar para o dispositivo, você deve criar uma "Coluna criada a partir de outras colunas" personalizada, usar a função no modelo dessa coluna e usar o valor dessa coluna em seus modelos de salvar/enviar.

author_links

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

`author_links(val_separator, pair_separator)` – retorna uma string contendo uma lista de autores e valores de link desses autores no formato: `author1 val_separator author1_link pair_separator author2 val_separator author2_link` etc.

Um autor é separado do valor do link pela string `val_separator` sem espaços adicionados. Supondo que o `val_separator` seja dois pontos, os pares `author:link value` são separados pelo argumento da string `pair_separator` sem espaços adicionados. Cabe a você escolher separadores que não ocorram em nomes de autores ou links. Um autor é incluído mesmo se o link do autor estiver vazio.

author_sorts

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

`author_sorts(val_separator)` – retorna uma string contendo uma lista de valores de classificação do autor para os autores do livro. A classificação é aquela nas informações de metadados do autor, que pode ser diferente do author_sort em livros. A lista retornada tem o formato `author sort 1 val_separator author sort 2` etc. sem espaços adicionados. Os valores de classificação do autor nesta lista estão na mesma ordem que os autores do livro. Se você quiser espaços ao redor de `val_separator` então inclua-os na string `val_separator`.

booksize

class calibre.utils.formatter_functions.**BuiltinBooksize**

`booksize()` – retorna o valor do campo `size` do calibre. Retorna " se o livro não tiver formatos.

Esta função funciona apenas na GUI. Se você quiser usar este valor em modelos de salvar em disco ou enviar para o dispositivo, então você deve criar uma "Coluna criada a partir de outras colunas" personalizada, usar a função no modelo dessa coluna e usar o valor dessa coluna em seus modelos de salvar/enviar

connected_device_name

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceName**

`connected_device_name(storage_location_key)` – se um dispositivo estiver conectado, então retorna o nome do dispositivo, caso contrário, retorna a string vazia. Cada local de armazenamento em um dispositivo tem seu próprio nome de dispositivo. Os nomes `storage_location_key` são `'main'`, `'carda'` e `'cardb'`. Esta função funciona apenas na GUI.

connected_device_uuid

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceUUID**

`connected_device_uuid(storage_location_key)` – se um dispositivo estiver conectado, então retorna o uuid do dispositivo (id exclusivo), caso contrário, retorna a string vazia. Cada local de armazenamento em um dispositivo tem um uuid diferente. Os nomes de localização `storage_location_key` são `'main'`, `'carda'` e `'cardb'`. Esta função funciona somente na GUI.

current_library_name

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**

`current_library_name()` – devolve último nome no caminho para a biblioteca atual do Calibre.

current_library_path

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryPath**

`current_library_path()` – devolve caminho completo para a biblioteca atual do Calibre.

current_virtual_library_name

class calibre.utils.formatter_functions.**BuiltinCurrentVirtualLibraryName**

`current_virtual_library_name()` – retorna o nome da biblioteca virtual atual se houver uma, caso contrário, a string vazia. O caso do nome da biblioteca é preservado. Exemplo:

```
program: current_virtual_library_name()
```

Esta função funciona apenas na GUI.

field

class calibre.utils.formatter_functions.**BuiltinField**

`field(lookup_name)` – retorna o valor do campo de metadados com o nome de pesquisa `lookup_name`. O prefixo `$` pode ser usado em vez da função, como em `$tags`.

formats_modtimes

class calibre.utils.formatter_functions.**BuiltinFormatsModtimes**

formats_modtimes(date_format_string) – retorna uma lista separada por vírgulas de itens separados por dois pontos FMT:DATE representando horários de modificação para os formatos de um livro. O parâmetro date_format_string especifica como a data deve ser formatada. Veja a função *format_date()* (página 187) para detalhes. Você pode usar a função *select()* (página 209) para obter o horário de modificação para um formato específico. Observe que os nomes de formato são sempre maiúsculos, como no EPUB.

formats_path_segments

class calibre.utils.formatter_functions.**BuiltinFormatsPathSegments**

formats_path_segments(with_author, with_title, with_format, with_ext, sep) – return parts of the path to a book format in the calibre library separated by sep. The parameter sep should usually be a slash ('/'). One use is to be sure that paths generated in Save to disk and Send to device templates are shortened consistently. Another is to be sure the paths on the device match the paths in the calibre library.

A book path consists of 3 segments: the author, the title including the calibre database id in parentheses, and the format (author - title). Calibre can shorten any of the three because of file name length limitations. You choose which segments to include by passing 1 for that segment. If you don't want a segment then pass 0 or the empty string for that segment. For example, the following returns just the format name without the extension:

```
formats_path_segments(0, 0, 1, 0, '/')
```

Because there is only one segment the separator is ignored.

If there are multiple formats (multiple extensions) then one of the extensions will be picked at random. If you care about which extension is used then get the path without the extension then add the desired extension to it.

Examples: Assume there is a book in the calibre library with an epub format by Joe Blogs with title 'Help'. It would have the path

```
Joe Blogs/Help - (calibre_id)/Help - Joe Blogs.epub
```

The following shows what is returned for various parameters:

- formats_path_segments(0, 0, 1, 0, '/') returns *Help - Joe Blogs*
- formats_path_segments(0, 0, 1, 1, '/') returns *Help - Joe Blogs.epub*
- formats_path_segments(1, 0, 1, 1, '/') returns *Joe Blogs/Help - Joe Blogs.epub*
- formats_path_segments(1, 0, 1, 0, '/') returns *Joe Blogs/Help - Joe Blogs*
- formats_path_segments(0, 1, 0, 0, '/') returns *Help - (calibre_id)*

formats_paths

class calibre.utils.formatter_functions.**BuiltinFormatsPaths**

formats_paths([separator]) – return a separator-separated list of colon-separated items FMT:PATH giving the full path to the formats of a book. The separator argument is optional. If not supplied then the separator is ', ' (comma space). If the separator is a comma then you can use the select() function to get the path for a specific format. Note that format names are always uppercase, as in EPUB.

formats_sizes

class calibre.utils.formatter_functions.**BuiltinFormatsSizes**

`formats_sizes()` – retorna uma lista separada por vírgulas de itens `FMT:SIZE` separados por dois pontos, fornecendo os tamanhos dos formatos de um livro em bytes. Você pode usar a função `select()` para obter o tamanho de um formato específico. Observe que os nomes dos formatos são sempre maiúsculos, como em EPUB.

has_cover

class calibre.utils.formatter_functions.**BuiltinHasCover**

`has_cover()` – devolve `'Yes'` se o livro possuir capa, caso contrário devolve uma expressão vazia.

is_marked

class calibre.utils.formatter_functions.**BuiltinIsMarked**

`is_marked()` – verifica se o livro está *marcado* no calibre. Se estiver, então retorna o valor da marca, seja `'true'` (letras minúsculas) ou uma lista separada por vírgulas de marcas nomeadas. Retorna `''` (a string vazia) se o livro não estiver marcado. Esta função funciona apenas na GUI.

language_codes

class calibre.utils.formatter_functions.**BuiltinLanguageCodes**

`language_codes(lang_strings)` – retorna os códigos de idioma[92] para os nomes de idioma passados em `lang_strings`. As strings devem estar no idioma do local atual. `lang_strings` é uma lista separada por vírgulas.

language_strings

class calibre.utils.formatter_functions.**BuiltinLanguageStrings**

`language_strings(value, localize)` – retorna os nomes de idioma para os códigos de idioma (veja aqui para nomes e códigos[93]) passados em `value`. Exemplo: `{languages:language_strings()}`. Se `localize` for zero, retorna as strings em inglês. Se `localize` não for zero, retorna as strings no idioma do local atual. `lang_codes` é uma lista separada por vírgulas.

ondevice

class calibre.utils.formatter_functions.**BuiltinOndevice**

`ondevice()` – retorna a string `'Sim'` se `ondevice` estiver definido, caso contrário retorna a string vazia. Esta função funciona apenas na GUI. Se você quiser usar este valor em modelos de salvar em disco ou enviar para o dispositivo, então você deve criar uma "Coluna personalizada construída a partir de outras colunas", usar a função no modelo dessa coluna e usar o valor dessa coluna em seus modelos de salvar/enviar.

raw_field

class calibre.utils.formatter_functions.**BuiltinRawField**

`raw_field(lookup_name [, optional_default])` – retorna o campo de metadados nomeado por `lookup_name` sem aplicar nenhuma formatação. Ele avalia e retorna o segundo argumento opcional `optional_default` se o valor do campo for indefinido (`None`). O prefixo `$$` pode ser usado em vez da função, como em `$$pubdate`.

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

raw_list

class calibre.utils.formatter_functions.**BuiltinRawList**

`raw_list(lookup_name, separator)` — devolve a lista de metadados nomeada por `lookup_name` 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()` – retorna uma lista separada por vírgulas das categorias de usuário que contêm este livro. Esta função funciona apenas na GUI. Se você quiser usar esses valores em modelos de salvar em disco ou enviar para o dispositivo, então você deve criar uma *Coluna personalizada construída a partir de outras colunas*, usar a função no modelo dessa coluna, e usar o valor dessa coluna em seus modelos de salvar/enviar

virtual_libraries

class calibre.utils.formatter_functions.**BuiltinVirtualLibraries**

`virtual_libraries()` – retorna uma lista separada por vírgulas de bibliotecas virtuais que contêm este livro. Esta função funciona apenas na GUI. Se você quiser usar esses valores em modelos de salvar em disco ou enviar para dispositivo, então você deve criar uma "Coluna construída a partir de outras colunas" personalizada, usar a função no modelo dessa coluna, e usar o valor dessa coluna em seus modelos de salvar/enviar.

Outro

arguments

class calibre.utils.formatter_functions.**BuiltinArguments**

`arguments(id[=expression] [, id[=expression]]*)` – Usado em um template armazenado para recuperar os argumentos passados na chamada. Ele declara e inicializa variáveis locais com os nomes fornecidos, os `id``s, tornando-os efetivamente parâmetros. As variáveis são posicionais; elas obtêm o valor do argumento fornecido na chamada na mesma posição. Se o argumento correspondente não for fornecido na chamada, então ``arguments()` atribui a essa variável o valor padrão fornecido. Se não houver valor padrão, então a variável é definida para a string vazia.

assign

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

`assign(id, val)` — atribui `value` a `id` e depois devolve `value`. O `id` deve ser um identificador e não uma expressão. Esta função pode, muitas vezes, ser substituída pelo operador =

globals

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

`globals(id[=expression] [, id[=expression]]*)` – Recupera "variáveis globais" que podem ser passadas para o formatador. O nome `id` é o nome da variável global. Ele declara e inicializa variáveis locais com os nomes das variáveis globais passadas (os parâmetros `id`. Se a variável correspondente não for fornecida nos globais, então ele atribui a essa variável o valor padrão fornecido. Se não houver valor padrão, então a variável é definida para a string vazia.)

is_dark_mode

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

`is_dark_mode()` – retorna `'1'` se o calibre estiver rodando no modo escuro, `''` (a string vazia) caso contrário. Esta função pode ser usada em regras avançadas de cores e ícones para escolher cores/ícones diferentes de acordo com o modo. Exemplo:

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

print

class calibre.utils.formatter_functions.**BuiltinPrint**

`print(a [, b]*)` — imprime os argumentos para a saída predefinida. A menos que vá iniciar o Calibre a partir da linha de comandos (`calibre-debug -g`), a saída irá parar a um buraco negro. A função `print` devlve sempre o primeiro argumento.

set_globals

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

`set_globals(id[=expression] [, id[=expression]]*)` – Define *variáveisglobais* que podem ser passadas para o formatador. Os globais recebem o nome do `id` passado. O valor do `id` é usado a menos que uma expressão seja fornecida.

Pesquisa de listas

identifier_in_list

class calibre.utils.formatter_functions.**BuiltinIdentifierInList**

`identifier_in_list(val, id_name [, found_val, not_found_val])` – trata `val` como uma lista de identificadores separados por vírgulas. Um identificador tem o formato `id_name:value`. O parâmetro `id_name` é o texto id_name a pesquisar, seja `id_name` ou `id_name:regexp`. O primeiro caso corresponde se houver algum identificador que corresponda a esse id_name. O segundo caso corresponde se id_name corresponder a um identificador e o regexp corresponder ao valor do identificador. Se `found_val` e `not_found_val` forem fornecidos, então se houver uma correspondência, então retorne `found_val`, caso contrário, retorne `not_found_val`. Se `found_val` e `not_found_val` não forem fornecidos, então, se houver uma correspondência, retorne o par `identificador:valor`, caso contrário, a string vazia (`''`).

list_contains

class calibre.utils.formatter_functions.**BuiltinInList**

`list_contains(value, separator, [ pattern, found_val, ]* not_found_val)` – interpreta o `value` como uma lista de itens separados por `separator`, verificando o `pattern` contra cada item na lista. Se o `pattern` corresponder a um item, então retorne `found_val`, caso contrário, retorne `not_found_val`. O par `pattern` e `found_value` pode ser repetido quantas vezes desejar, permitindo retornar valores diferentes dependendo do valor do item. Os padrões são verificados em ordem, e a primeira correspondência é retornada.

Aliases: in_list(), list_contains()

list_item

class calibre.utils.formatter_functions.**BuiltinListitem**

`list_item(value, index, separator)` – interpreta o `value` como uma lista de itens separados por `separator`, retornando o 'index'ésimo item. O primeiro item é o número zero. O último item tem o índice `-1` como em `list_item(-1,separator)`. Se o item não estiver na lista, então a string vazia é retornada. O separador tem o mesmo significado que na função count, geralmente vírgula, mas é e comercial para listas do tipo autor.

select

class calibre.utils.formatter_functions.**BuiltinSelect**

`select(value, key)` – interpreta o `value` como uma lista de itens separados por vírgulas com cada item tendo o formato `id:id_value` (o formato `identifier` do calibre). A função encontra o primeiro par com o id igual a `key` e retorna o `id_value` correspondente. Se nenhum id corresponder, a função retorna a string vazia.

str_in_list

class calibre.utils.formatter_functions.**BuiltinStrInList**

`str_in_list(value, separator, [ string, found_val, ]+ not_found_val)` – interpreta o `value` como uma lista de itens separados por `separator` e então compara `string` com cada valor na lista. A `string` não é uma expressão regular. Se `string` for igual a qualquer item (ignorando maiúsculas e minúsculas), então retorna o correspondente `found_val`. Se `string` contiver `separators`, então também é tratado como uma lista e cada subvalor é verificado. Os pares `string` e `found_value` podem ser repetidos quantas vezes forem desejadas, permitindo retornar valores diferentes dependendo do valor da string. Se nenhuma das strings corresponder, então `not_found_value` é retornado. As strings são verificadas em ordem. A primeira correspondência é retornada.

Recorrência

eval

class calibre.utils.formatter_functions.**BuiltinEval**

`eval(string)` – avalia a string como um programa, passando as variáveis locais. Isso permite usar o processador de modelo para construir resultados complexos a partir de variáveis locais. No Modo de Programa de Modelo[94], porque os caracteres `{` e `}` são interpretados antes que o modelo seja avaliado, você deve usar `[[` para o caractere `{` e `]]` para o caractere `}` . Eles são convertidos automaticamente. Observe também que prefixos e sufixos (a sintaxe `|prefix|suffix`) não podem ser usados no argumento para esta função ao usar o Modo de Programa de Modelo.

template

[94] https://manual.calibre-ebook.com/template_lang.html#more-complex-programs-in-template-expressions-template-program-mode

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 partilhadas entre a chamador e a análise do modelo. Se não usar o modo de Programação geral, e porque os caracteres `{` e `}` são especiais, deve usar `[[` para `{` e `]]` para o carácter `}`. Serão convertidos automaticamente. Por exemplo, `template(\'[[title_sort]]\')` irá analisar o modelo `{title_sort}` e devolver o seu valor. Note também que prefixos e sufixos (a sintaxwe `|prefix|suffix`) não podem ser usados no argumento para esta função ao usar o modo de Programação por modelos.

Relacional

cmp

class calibre.utils.formatter_functions.**BuiltinCmp**

`cmp(value, y, lt, eq, gt)` – compares `value` and `y` after converting both to numbers. Returns `lt` if `value <# y`, `eq` if `value ==# y`, otherwise `gt`. This function can usually be replaced with one of the numeric compare operators (`==#`, `<#`, `>#`, etc).

Mostrando a documentação em inglês devido ao erro FFML: Missing closing "`" for italics on line 2 in "cmp"

first_matching_cmp

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

`first_matching_cmp(val, [ cmp, result, ]* else_result)` — compara `val < cmp` em sequência, devolvendo o `result` associado para a primeira comparação com sucesso. Devolve `else_result` se não houver comparações com sucesso. Exemplo:

```
i = 10;
first_matching_cmp(i,5,"small",10,"middle",15,"large","giant")
```

devolve `"large"`. O mesmo exemplo, mas em que o primeiro valor seja 16, devolve `"giant"`.

strcmp

class calibre.utils.formatter_functions.**BuiltinStrcmp**

`strcmp(x, y, lt, eq, gt)` — faz uma comparação lexical, não diferenciadora de maiúsculas, de `x` e `y`. Devolve `lt` se `x < y`. Devolve `eq` se `x == y`. Caso contrário devolve `gt`. Esta função pode ser substituída, em muitos casos, por um dos operadores de comparação lexical (`==`, `>`, `<`, etc.)

strcmpcase

class calibre.utils.formatter_functions.**BuiltinStrcmpcase**

`strcmpcase(x, y, lt, eq, gt)` — faz uma comparação lexical entre `x` e `y` com diferenciação entre maiúsculas e minúsculas. Devolve `lt` se `x < y`. Devolve `eq` se `x == y`. De contrário devolve `gt` . Nota: Este NÃO é o comportamento predefinido usado pelo Calibre, por exemplo, nos operadores de comparação léxical (`==`, `>`, `<`, etc.). Esta função pode causar resultados inesperados. Utilize `strcmp()` sempre que possível.

Repetir sobre valores

first_non_empty

class calibre.utils.formatter_functions.**BuiltinFirstNonEmpty**

first_non_empty(value [, value]*) – retorna o primeiro value que não está vazio. Se todos os valores estiverem vazios, a string vazia será retornada. Você pode ter quantos valores quiser.

lookup

class calibre.utils.formatter_functions.**BuiltinLookup**

lookup(value, [pattern, key,]* else_key) – Os padrões serão verificados em relação ao value em ordem. Se um pattern corresponder, o valor do campo nomeado por key será retornado. Se nenhum padrão corresponder, o valor do campo nomeado por else_key será retornado. Veja também a função *switch()* (página 211).

switch

class calibre.utils.formatter_functions.**BuiltinSwitch**

switch(value, [patternN, valueN,]+ else_value) – para cada par patternN, valueN, verifica se o value corresponde à expressão regular patternN e, se for o caso, retorna o valueN associado. Se nenhum patternN corresponder, então else_value é retornado. Você pode ter quantos pares patternN, valueN desejar. A primeira correspondência é retornada.

switch_if

class calibre.utils.formatter_functions.**BuiltinSwitchIf**

switch_if([test_expression, value_expression,]+ else_expression) – para cada par test_expression, value_expression, verifica se test_expression é True (não vazio) e, se for, retorna o resultado de value_expression. Se nenhuma test_expression for True, o resultado de else_expression é retornado. Você pode ter quantos pares test_expression, value_expression quiser.

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 211) 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 213).

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
```

(continua na próxima página)

(continuação da página anterior)

```
    # If None, means book
    'publication_type',
    'uuid',             # A UUID usually of type 4
    'languages',        # ordered list of languages in this publication
    'publisher',        # Simple string, no special semantics
    # Absolute path to image file encoded in filesystem_encoding
    'cover',
    # Of the form (format, data) where format is, e.g. 'jpeg', 'png', 'gif'...
    'cover_data',
    # Either thumbnail data, or an object with the attribute
    # image_path which is the path to an image file, encoded
    # in filesystem_encoding
    'thumbnail',
))

BOOK_STRUCTURE_FIELDS = frozenset((
    # These are used by code, Null values are None.
    'toc', 'spine', 'guide', 'manifest',
))

USER_METADATA_FIELDS = frozenset((
    # A dict of dicts similar to field_metadata. Each field description dict
    # also contains a value field with the key #value#.
    'user_metadata',
))

DEVICE_METADATA_FIELDS = frozenset((
    'device_collections',   # Ordered list of strings
    'lpath',                # Unicode, / separated
    'size',                 # In bytes
    'mime',                 # Mimetype of the book file being represented
))

CALIBRE_METADATA_FIELDS = frozenset((
    'application_id',   # An application id, currently set to the db_id.
    'db_id',            # the calibre primary key of the item.
    'formats',          # list of formats (extensions) for this book
    # a dict of user category names, where the value is a list of item names
    # from the book that are in that category
    'user_categories',
    # a dict of items to associated hyperlink
    'link_maps',
))

ALL_METADATA_FIELDS =      SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           USER_METADATA_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)

# All fields except custom fields
```

(continua na próxima página)

(continuação da página anterior)

```
STANDARD_METADATA_FIELDS = SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)

# Metadata fields that smart update must do special processing to copy.
SC_FIELDS_NOT_COPIED =     frozenset(('title', 'title_sort', 'authors',
                                      'author_sort', 'author_sort_map',
                                      'cover_data', 'tags', 'languages',
                                      'identifiers'))

# Metadata fields that smart update should copy only if the source is not None
SC_FIELDS_COPY_NOT_NULL =  frozenset(('device_collections', 'lpath', 'size', 'comments
↪', 'thumbnail'))

# Metadata fields that smart update should copy without special handling
SC_COPYABLE_FIELDS =       SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS) - \
                           SC_FIELDS_NOT_COPIED.union(
                           SC_FIELDS_COPY_NOT_NULL)

SERIALIZABLE_FIELDS =      SOCIAL_METADATA_FIELDS.union(
                           USER_METADATA_FIELDS).union(
                           PUBLICATION_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS).union(
                           DEVICE_METADATA_FIELDS) - \
                           frozenset(('device_collections', 'formats',
                               'cover_data'))
# these are rebuilt when needed
```

10.4 Tudo sobre como usar 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.

Conteúdo

- *Primeiro, uma palavra de advertência e encorajamento* (página 216)
- *Onde você pode usar expressões regulares no calibre?* (página 216)
- *O que na terra é uma expressão regular?* (página 216)
- *Importa-se em explicar?* (página 217)
- *Isso não soa tão ruim. Qual é o próximo?* (página 217)
- *Ei, claro! Isso está começando a fazer sentido!* (página 217)

- *Well, these special characters are very neat and all, but what if I wanted to match a dot or a question mark?* (página 218)
- *Então, quais são os conjuntos mais úteis?* (página 218)
- *But if I had a few varying strings I wanted to match, things get complicated?* (página 218)
- *Você Perdeu…* (página 219)
- *No início, você disse que havia uma maneira de fazer expressões regulares insensíveis?* (página 219)
- *Acho que estou começando a entender essas expressões regulares agora… como eu faço para usá-las no calibre?* (página 219)
  - *Conversões* (página 219)
  - *Adicionando livros* (página 220)
  - *Edição em massa de metadados* (página 220)
- *Quick reference* (página 221)
- *Créditos* (página 225)

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 você 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 na terra *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 em 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 soa tão ruim. Qual é o próximo?

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 Ei, claro! Isso está começando 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[95]). 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

[95] https://www.mobileread.com/forums/showthread.php?t=75594"

ask in the forum if you are unclear on this point). The opening tag can be described using `<b.*?>`, the closing tag using `</b>`, thus we could remove everything between those tags using `<b.*?>.*?</b>`. But using this expression would be a bad idea, because it removes everything enclosed by <b>- tags (which, by the way, render the enclosed text in bold print), and it's a fair bet that we'll remove portions of the book in this way. Instead, include the beginning of the enclosed string as well, making the regular expression `<b.*?>\s*Generated\s+by\s+ABC\s+Amber\s+LIT.*?</b>` The `\s` with quantifiers are included here instead of explicitly using the spaces as seen in the string to catch any variations of the string that might occur. Remember to check what calibre will remove to make sure you don't remove any portions you want to keep if you test a new expression. If you only check one occurrence, you might miss a mismatch somewhere else in the text. Also note that should you accidentally remove more or fewer tags than you actually wanted to, calibre tries to repair the damaged code after doing the removal.

Adicionando 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.

Edição em massa de metadados

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[96].

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.

[96] 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.

Conteúdo

- *Classes de caracteres* (página 221)
- *Shorthand character classes* (página 222)
- *Os quantificadores* (página 222)
- *Greed* (página 222)
- *Alternation* (página 222)
- *Exclusão* (página 222)
- *Âncoras* (página 223)
- *Grupos* (página 223)
- *Lookarounds* (página 223)
- *Recursion* (página 224)
- *Caracteres especiais* (página 224)
- *Meta-caracteres* (página 224)
- *Modos* (página 225)

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

Representa-ção	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[97]. The actual regular expression library used by calibre is: regex[98] which supports several useful enhancements over the Python standard library one.

10.5 Crie seus próprios plugins para aumentar as funcionalidades do Calibre

O calibre possui um design muito modular. Quase todas as funcionalidades do calibre vêm na forma de plugins. Os plug-ins são usados para conversão, para baixar notícias (embora sejam chamadas de receitas), para vários componentes da interface do usuário, para conectar-se a dispositivos diferentes, para processar arquivos ao adicioná-los ao calibre e assim por diante. Você pode obter uma lista completa de todos os plugins integrados no calibre, visitando: guilabel: *Preferências-> Avançado-> Plugins*.

Aqui vamos mostrar a você como criar seus próprios plugins e adicionar novos atributos para o calibre

Conteúdo

- *Anatomia de um plugin do calibre* (página 226)

[97] https://docs.python.org/library/re.html
[98] https://bitbucket.org/mrabarnett/mrab-regex/src/hg/

- *Um plugin de Interface de Usuário* (página 227)
  - *__init__.py* (página 228)
  - *ui.py* (página 230)
  - *main.py* (página 231)
  - *Obtendo recursos do arquivo* ZIP *do plugin* (página 234)
  - *Habilitando configuração de usuário do seu plugin* (página 234)
- *Editar plug-ins de livros* (página 236)
  - *main.py* (página 236)
- *Adicionando traduções para o seu plugin* (página 239)
- *A API do plugin* (página 240)
- *Depurando plugins* (página 240)
- *Mais exemplos de plugins* (página 240)
- *Compartilhando seus plugins com os outros* (página 241)

Nota

Somente se aplica as versões do calibre >= 0.8.60

10.5.1 Anatomia de um plugin do 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)
```

(continua na próxima página)

(continuação da página anterior)

```
    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
```

Isso é tudo. Para adicionar este código ao Calibre como um plugin, apenas execute o seguinte na pasta na qual foi criado `__init__.py`:

```
calibre-customize -b .
```

Nota

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

Baixe o plugin Hello World em helloworld_plugin.zip[99].

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.

10.5.2 Um plugin de Interface de Usuário

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[100]

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.tx

Um arquivo de texto vazio usado para ativar a mágica do plug-in com vários arquivos. Este arquivo deve estar presente em todos os plugins que usam mais de um arquivo .py. Ele deve estar vazio e seu nome de arquivo deve estar no formato: plugin-import-name-**some_name**.txt``. A presença desse arquivo permite importar código dos arquivos .py presentes dentro do arquivo ZIP, usando uma instrução como:

```
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".

[99] https://calibre-ebook.com/downloads/helloworld_plugin.zip
[100] https://calibre-ebook.com/downloads/interface_demo_plugin.zip

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
Como antes, o arquivo que define a classe de plug-in

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 deste plugin

about.txt
Um arquivo de texto com informações sobre o plugin

translations
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

__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
```

(continua na próxima página)

(continuação da página anterior)

```
        Preferences->Plugins
        '''
        return True

    def config_widget(self):
        '''
        Implement this method and :meth:`save_settings` in your plugin to
        use a custom configuration dialog.

        This method, if implemented, must return a QWidget. The widget can have
        an optional method validate() that takes no arguments and is called
        immediately after the user clicks OK. Changes are applied if and only
        if the method returns True.

        If for some reason you cannot perform the configuration at this time,
        return a tuple of two strings (message, details), these will be
        displayed as a warning dialog to the user and the process will be
        aborted.

        The base class implementation of this method raises NotImplementedError
        so by default no user configuration is possible.
        '''
        # It is important to put this import statement here rather than at the
        # top of the module as importing the config class will also cause the
        # GUI libraries to be loaded, which we do not want when using calibre
        # from the command line
        from calibre_plugins.interface_demo.config import ConfigWidget
        return ConfigWidget()

    def save_settings(self, config_widget):
        '''
        Save the settings specified by the user with config_widget.

        :param config_widget: The widget returned by :meth:`config_widget`.
        '''
        config_widget.save_settings()

        # Apply the changes
        ac = self.actual_plugin_
        if ac is not None:
            ac.apply_settings()
```

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 don't want to use a keyboard
    # shortcut. Remember that currently calibre has no central management for
    # keyboard shortcuts, so try to use an unusual/unused shortcut.
    action_spec = ('Interface Plugin Demo', None,
            'Run the Interface Plugin Demo', 'Ctrl+Shift+F1')

    def genesis(self):
        # This method is called once per plugin, do initial setup here

        # Set the icon for this interface action
        # The get_icons function is a builtin function defined for all your
        # plugin code. It loads icons from the plugin zip file. It returns
        # QIcon objects, if you want the actual data, use the analogous
        # get_resources builtin function.
        #
        # Note that if you are loading more than one icon, for performance, you
        # should pass a list of names to get_icons. In this case, get_icons
        # will return a dictionary mapping names to QIcons. Names that
        # are not found in the zip file will result in null QIcons.
        icon = get_icons('images/icon.png', 'Interface Demo Plugin')

        # The qaction is automatically created from the action_spec defined
        # above
        self.qaction.setIcon(icon)
        self.qaction.triggered.connect(self.show_dialog)

    def show_dialog(self):
        # The base plugin object defined in __init__.py
        base_plugin_object = self.interface_action_base_plugin
        # Show the config dialog
        # The config dialog can also be shown from within
        # Preferences->Plugins, which is why the do_user_config
        # method is defined on the base plugin class
        do_user_config = base_plugin_object.do_user_config

        # self.gui is the main calibre GUI. It acts as the gateway to access
        # all the elements of the calibre user interface, it should also be the
        # parent of the dialog
        d = DemoDialog(self.gui, self.qaction.icon(), do_user_config)
        d.show()
```

(continua na próxima página)

(continuação da página anterior)

```
    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

        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)
```

(continua na próxima página)

(continuação da página anterior)

```
        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
        # 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
```

(continua na próxima página)

(continuação da página anterior)

```
            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 don't use add_format_with_hooks as the hooks were
                # already run when the file was first added to calibre.
                db.add_format(book_id, fmt, ffile, run_hooks=False)

        info_dialog(self, 'Updated files',
                f'Updated the metadata in the files of {len(ids)} book(s)',
                show=True)

    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])
```

Obtendo recursos do arquivo *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.

Habilitando configuração de usuário do 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.

        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 don't accidentally clobber a calibre config file
prefs = JSONConfig('plugins/interface_demo')

# Set defaults
prefs.defaults['hello_world_msg'] = 'Hello, World!'


class ConfigWidget(QWidget):

    def __init__(self):
        QWidget.__init__(self)
        self.l = QHBoxLayout()
        self.setLayout(self.l)

        self.label = QLabel('Hello world &message:')
        self.l.addWidget(self.label)

        self.msg = QLineEdit(self)
        self.msg.setText(prefs['hello_world_msg'])
        self.l.addWidget(self.msg)
        self.label.setBuddy(self.msg)

    def save_settings(self):
        prefs['hello_world_msg'] = self.msg.text()
```

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 plug-ins de livros

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[101].

The first step, as for all plugins is to create the import name empty txt file, as described *above* (página 227). We shall name the file `plugin-import-name-editor_plugin_demo.txt`.

Now we create the mandatory `__init__.py` file that contains metadata about the plugin – its name, author, version, etc.

```


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 369) 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 369) class for more details.

main.py

Here we will see the definition of a single tool that will multiply all font sizes in the book by a number provided by the user. This tool demonstrates various important concepts that you will need in developing your own plugins, so you should read the (heavily commented) source code carefully.

```


from css_parser.css import CSSRule
from qt.core import QAction, QInputDialog

from calibre import force_unicode
from calibre.ebooks.oeb.polish.container import OEB_DOCS, OEB_STYLES, serialize
from calibre.gui2 import error_dialog

# The base class that all tools must inherit from
from calibre.gui2.tweak_book.plugin import Tool


```

(continua na próxima página)

[101] https://calibre-ebook.com/downloads/editor_demo_plugin.zip

(continuação da página anterior)

```
class DemoTool(Tool):

    #: Set this to a unique name it will be used as a key
    name = 'demo-tool'

    #: If True the user can choose to place this tool in the plugins toolbar
    allowed_in_toolbar = True

    #: If True the user can choose to place this tool in the plugins menu
    allowed_in_menu = True

    def create_action(self, for_toolbar=True):
        # Create an action, this will be added to the plugins toolbar and
        # the plugins menu
        ac = QAction(get_icons('images/icon.png'), 'Magnify fonts', self.gui)  #␣
↪noqa: F821
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
            # register it for the action created for the menu, not the toolbar,
            # to avoid a double trigger
            self.register_shortcut(ac, 'magnify-fonts-tool', default_keys=(
↪'Ctrl+Shift+Alt+D',))
        ac.triggered.connect(self.ask_user)
        return ac

    def ask_user(self):
        # Ask the user for a factor by which to multiply all font sizes
        factor, ok = QInputDialog.getDouble(
            self.gui, 'Enter a magnification factor', 'Allow font sizes in the book␣
↪will be multiplied by the specified factor',
            value=2, min=0.1, max=4
        )
        if ok:
            # Ensure any in progress editing the user is doing is present in the␣
↪container
            self.boss.commit_all_editors_to_container()
            try:
                self.magnify_fonts(factor)
            except Exception:
                # Something bad happened report the error to the user
                import traceback
                error_dialog(self.gui, _('Failed to magnify fonts'), _(
                    'Failed to magnify fonts, click "Show details" for more info'),
                    det_msg=traceback.format_exc(), show=True)
                # Revert to the saved restore point
                self.boss.revert_requested(self.boss.global_undo.previous_container)
            else:
                # Show the user what changes we have made, allowing her to
                # revert them if necessary
                self.boss.show_current_diff()
                # Update the editor UI to take into account all the changes we
                # have made
```

(continua na próxima página)

(continuação da página anterior)

```
                self.boss.apply_container_update_to_gui()

    def magnify_fonts(self, factor):
        # Magnify all font sizes defined in the book by the specified factor
        # First we create a restore point so that the user can undo all changes
        # we make.
        self.boss.add_savepoint('Before: Magnify fonts')

        container = self.current_container  # The book being edited as a container␣
↪object

        # Iterate over all style declarations in the book, this means css
        # stylesheets, <style> tags and style="" attributes
        for name, media_type in container.mime_map.items():
            if media_type in OEB_STYLES:
                # A stylesheet. Parsed stylesheets are css_parser CSSStylesheet
                # objects.
                self.magnify_stylesheet(container.parsed(name), factor)
                container.dirty(name)  # Tell the container that we have changed the␣
↪stylesheet
            elif media_type in OEB_DOCS:
                # A HTML file. Parsed HTML files are lxml elements

                for style_tag in container.parsed(name).xpath('//*[local-name="style"]
↪'):
                    if style_tag.text and style_tag.get('type', None) in {None, 'text/
↪css'}:
                        # We have an inline CSS <style> tag, parse it into a
                        # stylesheet object
                        sheet = container.parse_css(style_tag.text)
                        self.magnify_stylesheet(sheet, factor)
                        style_tag.text = serialize(sheet, 'text/css', pretty_
↪print=True)
                        container.dirty(name)  # Tell the container that we have␣
↪changed the stylesheet
                for elem in container.parsed(name).xpath('//*[@style]'):
                    # Process inline style attributes
                    block = container.parse_css(elem.get('style'), is_
↪declaration=True)
                    self.magnify_declaration(block, factor)
                    elem.set('style', force_unicode(block.getCssText(separator=' '),
↪'utf-8'))

    def magnify_stylesheet(self, sheet, factor):
        # Magnify all fonts in the specified stylesheet by the specified
        # factor.
        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
```

(continua na próxima página)

(continuação da página anterior)

```
        val = style.getPropertyValue('font-size')
        if not val:
            return
        # see if the font-size contains a number
        num = re.search(r'[0-9.]+', val)
        if num is not None:
            num = num.group()
            val = val.replace(num, f'{float(num)*factor:f}')
            style.setProperty('font-size', val)
        # We should also be dealing with the font shorthand property and
        # font sizes specified as non numbers, but those are left as exercises
        # for the reader
```

Let's break down `main.py`. We see that it defines a single tool, named *Magnify fonts*. This tool will ask the user for a number and multiply all font sizes in the book by that number.

The first important thing is the tool name which you must set to some relatively unique string as it will be used as the key for this tool.

The next important entry point is the `calibre.gui2.tweak_book.plugin.Tool.create_action()` (página 370). 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 370) 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 362) 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 ferramentas de edição de e-book* (página 361).

10.5.4 Adicionando traduções para o 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[102] program for this.

Send these .po files to your translators. Once you get them back, compile them into .mo files. You can again use Poedit for that, or just do:

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

[102] https://poedit.net/

Put the .mo files into the `translations` folder in your plugin.

The last step is to simply call the function *load_translations()* at the top of your plugin's .py files. For performance reasons you should only call this function in those .py files that actually have translatable strings. So in a typical User Interface plugin you would call it at the top of `ui.py` but not `__init__.py`.

You can test the translations of your plugins by changing the user interface language in calibre under *Preferences → Interface → Look & feel* or by running calibre with the `CALIBRE_OVERRIDE_LANG` environment variable set. For example:

```
CALIBRE_OVERRIDE_LANG=de
```

Replace `de` with the language code of the language you want to test.

For translations with plurals, use the `ngettext()` function instead of `_()`. For example:

```
ngettext('Delete a book', 'Delete {} books', num_books).format(num_books)
```

10.5.5 A API do 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 256).

Your plugin is almost certainly going to use code from calibre. To learn how to find various bits of functionality in the calibre code base, read the section on the calibre *Layout do código* (página 344).

10.5.6 Depurando plugins

A primeira e mais importante etapa é executar o calibre no modo de depuração. Você 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 no console; ao executar a partir do *Calibre*, a saída irá para um arquivo `.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[103].

[103] https://www.mobileread.com/forums/showthread.php?t=118764

10.5.8 Compartilhando seus plugins com os 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[104].

10.6 Tipografia matemática em e-books

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[105] 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 arquivo HTML simples com matemáticas

É possível escrever matemática embutida dentro de um arquivo *HTML* simples e o visualizador de E-book do *Calibre* irá renderizá-lo em uma matemática adequadamente tipada. No exemplo abaixo, usamos a notação *TeX* para matemática. Você verá que pode usar comandos *TeX* normais, com a pequena ressalva de que os sinais & comercial e menor e maior devem ser escritos como `&`, `<` e `>` respectivamente.

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>
```

É isso, agora é possível digitar matemática da mesma forma que faria em um arquivo `.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 se parece com a seguinte captura de tela no visualizador de E-book do *Calibre*.

$$\begin{aligned}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= \rho x - y - xz \\
\dot{z} &= -\beta z + xy
\end{aligned}$$

Figura1: *The Lorenz Equations*

[104] https://www.mobileread.com/forums/forumdisplay.php?f=237
[105] https://www.mathjax.org

O arquivo *HTML* completo, com mais equações e matemática embutida, é reproduzido abaixo. É possível converter este arquivo *HTML* em *EPUB* no *Calibre* para acabar com um e-book que você pode distribuir facilmente para outras pessoas.

```
<!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>\[
\left( \sum_{k=1}^n a_k b_k \right)^{\!\!2} \leq
 \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
\]</p>

<h2>A Cross Product Formula</h2>
```

(continua na próxima página)

(continuação da página anterior)

```
<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>Inline Mathematics</h2>

<p>While display equations look good for a page of samples, the
ability to mix math and text in a paragraph is also important.  This
expression \(\sqrt{3x-1}+(1+x)^2\) is an example of an inline equation.  As
you see, equations can be used this way as well, without unduly
disturbing the spacing between lines.</p>

<h2>References to equations</h2>
```

(continua na próxima página)

(continuação da página anterior)

```
<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>
```

Nota

O visualizador de E-book do calibre suporta tanto MathML quanto TeX, mas é necessário incluir a linha `<script type="text/x-mathjax-config"></script>` no seu arquivo HTML, senão o MathML não irá renderizar.

10.6.2 Mais informações

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[106].

10.7 Criar catálogos AZW3 • EPUB • MOBI

A função Criar catálogo do calibre permite que você crie um catálogo da sua biblioteca em uma variedade de formatos. Esta página da ajuda descreve as opções de catalogação ao gerar um catálogo nos formatos AZW3, EPUB e MOBI.

- *Selecionar livros para o catálogo* (página 244)
- *Sessões incluidas* (página 245)
- *Prefixos* (página 246)
- *Livros apagados* (página 246)
- *Apagar gêneros* (página 247)
- *Outras opções* (página 247)
- *Capas personalizadas dos catálogos* (página 248)
- *Recursos adicionais de ajuda* (página 248)

10.7.1 Selecionar livros para o catálogo

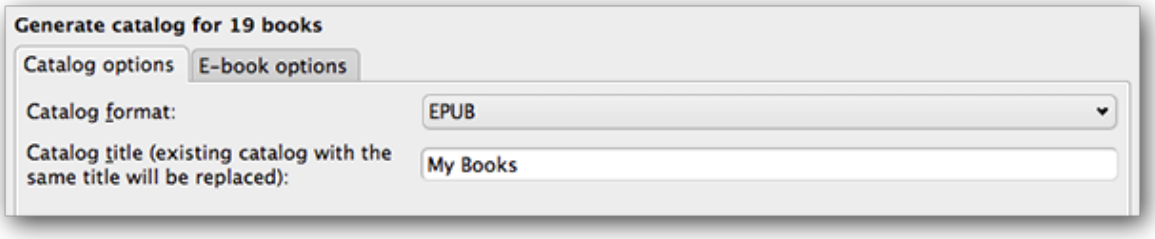
Se você quiser *toda* a sua biblioteca catalogada, remova qualquer critério de busca ou filtro na janela principal. Com um único livro selecionado, todos os livros na sua biblioteca serão candidatos à inclusão no catálogo gerado. Livros individuais podem ser excluídos por vários critérios; veja abaixo a seção *Apagar gêneros* (página 247) para mais informações.

Se você quiser apenas *algumas* de suas bibliotecas catalogadas, você tem duas opções:

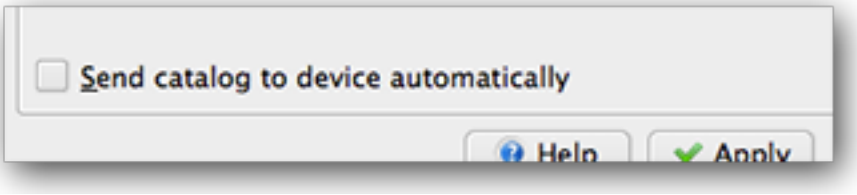
- Crie uma seleção múltipla de livros que você quer catalogados. Com mais de um livro selecionado na janela principal do calibre, apenas os livros selecionados serão catalogados.
- Utilize a caixa de busca ou a navegação por etiqueta para filtrar os livros. Apenas os livros mostrados serão catalogados.

[106] https://www.mathjax.org

Para iniciar geração de catálogo, selecione o item do menu :guilabel: Converter livros > Criar catálogo dos livros na sua biblioteca do calibre. Você pode acrescentar também :guilabel: um botão Criar catálogo na barra de ferramenta em *Preferências* > *Interface* > *Barras de Ferramentas* para acesso fácil ao Gerar diálogo do catálogo.

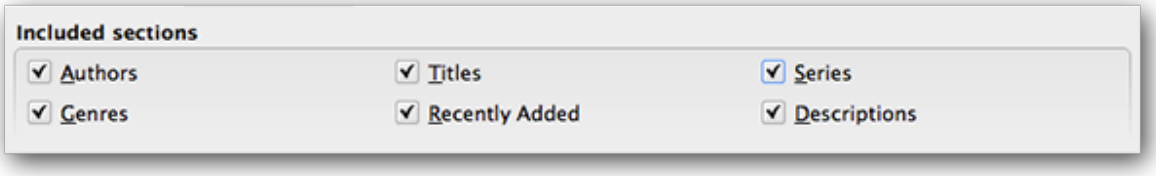


Em: guilabel : ` Opções de Catálogo` , selecione ** AZW3 , EPUB ou MOBI ** como o formato de catálogo. No campo : guilabel : ` Título do Catálogo ` , forneça um nome que vai ser usado para o catálogo gerado . Se um catálogo com o mesmo nome e formato já existir, ele será substituído com o catálogo recém-criado.



Autorizando: guilabel : ` Enviar catálogo para dispositivos automaticamente` irá baixar o catálogo gerado para um dispositivo conectado após a conclusão.

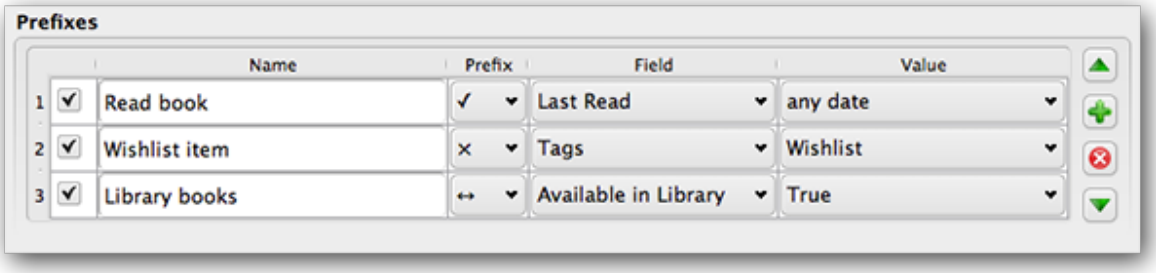
10.7.2 Sessões incluidas



Seções ativadas por uma marca de verificação, serão incluídas no catálogo gerado:

- *Autores* - todos os livros, ordenados por autor, apresentados em um formato de lista. Os livros que não fazem parte de uma série são listados antes dos livros que fazem parte de uma.
- *Títulos* - todos os livros, ordenados por título, apresentados em formato de lista.
- *Séries* - todos os livros que fazem parte de uma série, ordenados por série, apresentados em um formato de lista.
- *Gêneros* - lista de gêneros individuais, ordenados por Autor e Séries.
- *Recently Added* - all books, sorted in reverse chronological order. List includes books added in the last 30 days, then a month-by-month listing of added books.
- : guilabel: *Descrição* - página de descrição detalhada para cada livro, incluindo a contracapa e comentários. Ordenados por autor, com livros de não listados em série antes dos livros seriados.

10.7.3 Prefixos



As regras de prefixo permitem que você adicione um prefixo para reservar listas quando determinados critérios são cumpridos. Por exemplo, você pode marcar os livros que já leu com um visto e livros de uma lista de desejos com um X.

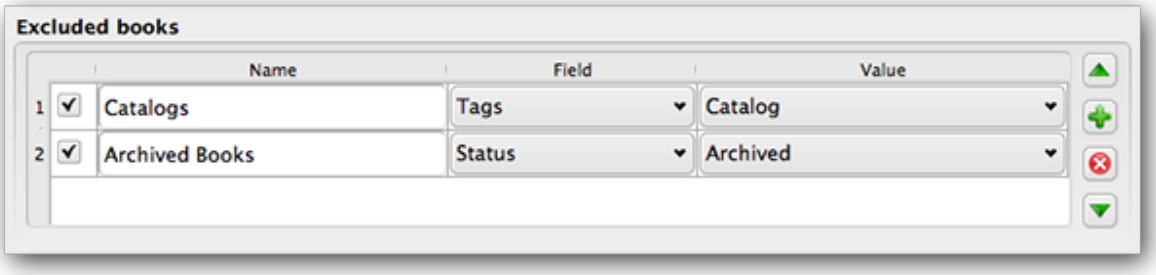
A caixa de seleção na primeira coluna ativa a regra. *Nome* é um nome da regra que você fornece. *Campo* pode ser *Etiquetas* ou uma coluna personalizada da sua biblioteca. *Valor* é o conteúdo que corresponde ao *Campo*. Quando uma regra de prefixo for satisfeita, o livro será marcado com o *Prefixo* selecionado.

Três regras de prefixo foram especificadas no exemplo acima:

1. *Lidos* especifica que um livro com uma data qualquer na coluna personalizada *Último lido* vai recebe um visto.
2. *Item da lista de desejos* especifica que qualquer livro com a etiqueta *Lista de desejos* vai receber um X.
3. *Livros da biblioteca* especifica que qualquer livro com valor Verdade (ou Sim) na coluna personalizada *Disponível* vai receber um seta dupla.

A primeira regra de correspondência fornece o prefixo. Regras desativadas ou incompletas serão ignoradas.

10.7.4 Livros apagados



As regras de exclusão permitem que você especifique os livros que não serão catalogados.

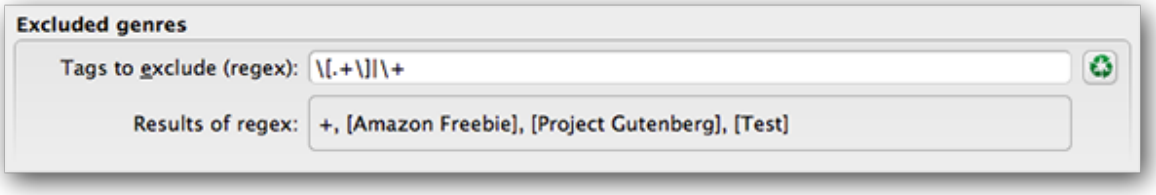
A caixa de seleção na primeira coluna ativa a regra. Dê um nome à regra na coluna *Nome*. *Campo* pode ser *Etiquetas* ou uma coluna personalizada de sua biblioteca. *Valor* é o conteúdo que corresponde ao *Campo*. Quando uma regra de exclusão for satisfeita, o livro será apagado do catálogo.

Duas regras de exclusão foram especificadas no exemplo acima:

1. A regra *Catálogos* especifica que quaisquer livros como a etiqueta *Catálogo* serão apagados do catálogo.
2. A regra *Livros arquivados* especifica que quaisquer livros com o valor *Arquivado* na coluna personalizada *Status* serão apagados do catálogo.

Todas as regras são avaliadas para cada livro. Regras desativadas ou incompletas são ignoradas.

10.7.5 Apagar gêneros



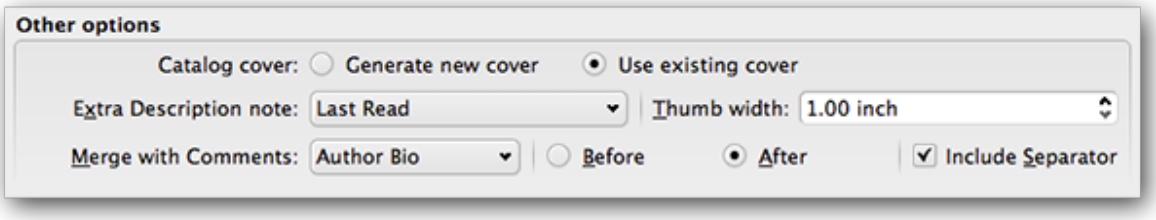
Ao se gerar um catálogo, as etiquetas da base de dados são usadas como gêneros. Por exemplo, se você usa as etiquetas `Ficção` e `Não ficção`, elas vão se tornar gêneros no catálogo que serão atribuídos aos respectivos livros. Os livros serão listados em cada gênero para o qual houver uma etiqueta correspondente.

Você pode usar algumas etiquetas para outros fins, talvez um + para indicar um livro de leitura, ou uma etiqueta entre colchetes como `[Amazon Freebie]` para indicar a fonte de um livro. A *Excluir gênero* permite especificar as etiquetas que você não quer que sejam utilizados como gêneros no catálogo. O padrão regex de exclusão `\[.+\]\+` evita que as etiquetas na forma `[etiqueta]` e o sinal `+`, que marca os livros lidos, sejam usados como gênero no catálogo.

Você também pode usar um nome exato de etiqueta em uma regex. Por exemplo, `[Amazon Freebie]` ou `[Project Gutenberg]`. Se você quer listar múltiplas etiquetas exatas para exclusão, ponha um caractere de barra vertical (|) entre eles: `[Amazon Freebie]|[Project Gutenberg]`.

Resultados da regex mostra as etiquetas que serão excluídas quando o catálogo for elaborado, de acordo com as etiquetas de sua base de dados e com o padrão regex utilizado. Os resultados são atualizados, conforme o padrão regex é modificado.

10.7.6 Outras opções



Capa do catálogo especifica se deve gerar uma nova capa ou usar uma capa já existente. É possível criar uma capa personalizada para os seus catálogos - ver *Capas personalizadas dos catálogos* (página 248) para mais informações. Se você criou uma capa personalizada que pretende reutilizar, selecione *Usar capa existente*; Caso contrário, selecione *Gerar nova capa*.

Nota de Descrição adicional especifica os conteúdos de uma coluna personalizada que serão inseridos na página Descrição, próximo da miniatura da capa. Por exemplo: você pode exibir a última data em que leu um livro usando a coluna personalizada *Última leitura*. Para usos avançados da função nota de Descrição, leia esta postagem no fórum do calibre[107].

Largura da miniatura especifica a preferência de largura das miniaturas de capa incluídas com a página Descrição. As miniaturas são armazenadas em cache para melhorar o desempenho. Para provar larguras diferentes, gere um catálogo com poucos livros até que consiga identificar a largura preferencial. Então, gere o catálogo completo. Ao gerar pela primeira vez um catálogo com nova largura de miniatura, o desempenho pode ser mais lento; no entanto, nas próximas vezes, o chace de miniaturas é aproveitado.

[107] https://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395

Merge with comments especifica a coluna personalizada cujo conteúdo não dever ser eliminado ao ser mesclado com os metadados dos comentários durante a criação do catálogo. Por exemplo, você pode anexar o conteúdo de uma coluna personalizada *Author bio* ao metadados de comentários. Você ainda pode inserir os conteúdos *antes ou depois* dos comentários e, opcionalmente, separá-los com um filete horizontal. Entre os tipos possíveis de coluna personalizada estão `texto, comentários e composição`.

10.7.7 Capas personalizadas dos catálogos

Ao instalar Generate Cover plugin[108] , você pode criar capas personalizadas para seus catálogos. Instale o plugin em: *Preferências> Avançado> Plugins> Obter novos plugins*.

10.7.8 Recursos adicionais de ajuda

Para mais informações sobre a função Catálogo do calibre, leia a nota no fórum de MobileRead Creating Catalogs - Start here[109], que indica como personalizar os templates de catálogo e enviar relatório de erro.

Para fazer perguntas e discutir com outros usuários a função Catálogo do calibre, visite o fórum do MobileRead Library Management[110].

10.8 Bibliotecas virtuais

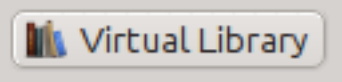
No calibre, uma biblioteca Virtual é uma forma de fazer com que o calibre abra apenas um pequeno conjunto de uma biblioteca normal. Por exemplo, pode-se querer trabalhar apenas com livros de um certo autor, ou livros que tenham uma etiqueta. Usar bibliotecas Virtuais é a forma ideal de dividir sua coleção de livros em sub-coleções menores. É melhor do que dividir sua biblioteca em várias bibliotecas menores porque, quando se quiser pesquisar em toda a biblioteca, pode-se simplesmente voltar à visualização da biblioteca toda. Não há formas de pesquisar em várias bibliotecas separadas simultaneamente no calibre.

Uma biblioteca Virtual é diferente de uma busca simples. Uma busca só irá restringir a lista de livros exibidos. Uma biblioteca Virtual faz isso e também ela restringe as entradas exibidas na etiqueta *Navegador de etiquetas* para a esquerda. O Navegador de etiquetas irá mostras apenas etiquetas, autores, séries, editoras, etc. que venham em livros na biblioteca

[108] https://www.mobileread.com/forums/showthread.php?t=124219
[109] https://www.mobileread.com/forums/showthread.php?t=118556
[110] https://www.mobileread.com/forums/forumdisplay.php?f=236

Virtual. Uma biblioteca Virtual, assim, se comporta como se a biblioteca atual contivesse apenas um conjunto restrito de livros.

10.8.1 Criação de bibliotecas virtuais



Para usar uma biblioteca Virtual clique no botão *Biblioteca Virtual* localizado à esquerda da Barra de Pesquisa e selecione a opção *Criar biblioteca Virtual*. Como primeiro exemplo, crie uma biblioteca Virtual que mostre apenas os livros de um autor específico. Clique no atalho *Autores* como mostrado na imagem abaixo e escolha o autor que deseja consultar e clique em OK.

A janela Criar biblioteca Virtual foi preenchida para você. Clique em OK e você verá que uma nova biblioteca Virtual foi criada, e automaticamente tornada ativa, que exibe apenas os livros do autor selecionado. Para o calibre, é como se sua biblioteca contivesse apenas os livros do autor selecionado.

Pode-se alternar para a biblioteca inteira a qualquer momento, novamente clicando em *Biblioteca Virtual* e selecionando a opção chamada <*None*>.

Bibliotecas Virtuais são baseadas em *buscas*. Você pode usar qualquer busca como base para uma biblioteca Virtual. A biblioteca Virtual irá conter apenas os livros selecionados pela busca. Primeiro, digite o que você quer buscar na barra de Pesquisa ou crie uma busca usando a etiqueta *Navegador de Etiquetas*. Quando você estiver feliz com os resultados da busca, clique no botão *Biblioteca Virtual*, escolha *Criar biblioteca Virtual* e digite um nome para a nova biblioteca Virtual. A biblioteca Virtual será então criada, baseada na busca que foi digitada. Buscas são muito poderosas, para exemplos dos tipos de coisas que você pode fazer com elas, consulte *A interface de pesquisa* (página 12).

Exemplos de bibliotecas virtuais úteis

- **Livros adicionados no calibre no último dia:**
  data:>1diaatrás
- **Livros adicionados ao calibre no último mês:**
  data:>30diasatrás
- **Livros com uma classificação de 5 estrelas:**
  classificação:5
- **Livros com uma classificação de pelo menos 4 estrelas:**
  classificação:>=4

- **Livros sem classificação:**
  classificação:falso
- **Periódicos baixados via função Obter Notícias no calibre:**
  etiquetas:=Novidades and autor:=calibre
- **Livros sem etiquetas:**
  etiquetas:falso
- **Livros sem capas:**
  capa:falso

10.8.2 Trabalhando com bibliotecas Virtuais

Pode-se editar uma biblioteca Virtual previamente criada ou apagá-la, clicando em *Biblioteca Virtual* e escolhendo a ação apropriada.

Você pode dizer ao calibre que você sempre quer ativar uma biblioteca Virtual específica quando a biblioteca atual seja aberta, indo em *Preferências → Interface → Comportamento*.

Você pode rapidamente usar a busca atual como biblioteca Virtual temporária clicando no botão:guilabel:*Biblioteca Virtual* e escolhendo a opção **busca atual* .

Você pode exibir todas as bibliotecas Virtuais disponíveis como abas acima da lista de livros. Isto é particularmente útil se você goste de frequentemente alternar entre bibliotecas Virtuais. Clique no botão *Biblioteca Virtual* e selecione *Mostrar bibliotecas Virtuais como abas*. Pode-se rearranjar as abas arrastando-as e fechando as abas que você não deseja ver. Abas fechadas podem ser restauradas ao clicar com o botão direito sobre a barra de abas.

10.8.3 Usar bibliotecas virtuais nas pesquisas

Pode-se pesquisar em livros que estão em uma biblioteca Virtual usando o prefixo `vl:`. Por exemplo, `vl:Lidos` irá buscar em todos os livros na biblioteca virtual *Lidos*. A busca `vl:Lidos and vl:"Ficção Científica"` vai buscar em todos os livros que estejam tanto na biblioteca Virtual *Lidos* quanto na *Ficção Científica*.

O valor após `vl:` deve ser o nome de uma biblioteca Virtual. Se o nome da biblioteca Virtual contiver espaços, coloque-o entre aspas.

Um uso para uma busca em biblioteca Virtual está no servidor de Conteúdo. Em *Preferências → Compartilhar pela rede → Exigir usuário e senha*, pode-se limitar as bibliotecas Virtuais visíveis e um usuário. Para cada biblioteca visível, pode-se especificar uma expressão de busca para limitar ainda mais quais livros são vistos. Use `vl:"Nome da biblioteca Virtual"` para limitar os livros para aqueles presentes nesta biblioteca Virtual.

10.8.4 Usar restrições adicionais

Pode-se restringir ainda mais os livros exibidos em uma biblioteca Virtual usando *Restrição adicional*. Uma restrição adicional é uma busca salva previamente criada, que pode ser aplicada à biblioteca Virtual atual para restringir ainda mais os livros mostrados na biblioteca Virtual. Por exemplo, digamos que você tenha uma biblioteca Virtual para livros marcados como *Ficção Histórica* e uma busca salva que mostre os livros não lidos, você pode clicar no botão *Biblioteca Virtual* e escolher a opção *Restrição adicional* para mostrar apenas os livros de Ficção Histórica que não foram lidos. Para aprender a utilizar buscas salvas, veja *Salvando pesquisas* (página 18).

CAPÍTULO 11

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.

- *Mudar para uma biblioteca específica* (página 251)
- *Mostrar um livro específico no calibre* (página 252)
- *Abra um livro específico no visualizador de e-books em uma posição específica* (página 252)
- *Procurando por livros* (página 252)
- *Abra uma janela de detalhes do livro em um livro em alguma biblioteca* (página 253)
- *Abra as notas associadas a um autor/seri/etc.* (página 253)
- *Codificação Hex dos parâmetros da URL* (página 254)

11.1 Mudar 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
```

Isso mostrará o livro com `book_id` (um número) no calibre. Os ids para os livros podem ser vistos na interface do calibre ao passar o mouse sobre o link *Clique para abrir* no painel *Detalhes do livro*, é o número entre colchetes no final do caminho para a pasta do livro.

Você pode copiar um link para o livro atual exibido no calibre clicando com o botão direito no painel "Detalhes do livro" e escolhendo "Copiar link para o livro".

Se uma pesquisa estiver ativa e o livro não for correspondido pela pesquisa, então a pesquisa é limpa.

Se uma biblioteca virtual for selecionada, o calibre a usará ao mostrar o livro. Se o livro não for encontrado nessa biblioteca virtual, a biblioteca virtual será limpa.

Se você quiser alternar para uma biblioteca virtual específica ao mostrar o livro, 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
```

substituindo espaços no nome da biblioteca virtual por "%20". Se o livro não for encontrado nessa biblioteca virtual, a biblioteca virtual será ignorada.

11.3 Abra um livro específico no visualizador de e-books em uma posição específica

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 Procurando por 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.

Por padrão, se uma biblioteca virtual for selecionada, o calibre irá limpá-la antes de fazer a pesquisa para garantir que todos os livros sejam encontrados. Se você deseja preservar a biblioteca virtual, use:

```
calibre://search/Library_Name?q=query&virtual_library=_
```

Se você quiser alternar para uma biblioteca virtual específica, use:

```
calibre://search/Library_Name?virtual_library=Library%20Name
or
calibre://search/Library_Name?encoded_virtual_library=hex_encoded_virtual_library_name
```

substituindo espaços no nome da biblioteca virtual por `%20`.

Se você realizar uma busca no calibre e quiser gerar um link para ela, você pode fazer isso clicando com o botão direito na barra de pesquisa e escolhendo "Copiar busca como URL".

11.5 Abra uma janela de detalhes do livro em um livro em alguma biblioteca

A sintaxe da URL é:

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

Isso abre uma janela de detalhes do livro no livro especificado da biblioteca especificada sem alterar a biblioteca atual ou o livro selecionado.

11.6 Abra as notas associadas a um autor/seri/etc.

A sintaxe da URL é:

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

Isso abre uma janela mostrando as notas do item especificado. A maneira mais fácil de criar URLs desse tipo é mostrar as notas desejadas no calibre e clicar no botão *Copy URL* para copiar a URL para a área de transferência e colá-la onde você precisar.

Aqui `Field_Name` é o nome das colunas, como `authors` ou `tags`. Para colunas criadas pelo usuário, substitua o `#` inicial no nome do campo por um sublinhado, então `#mytags` se torna `_mytags`.

Além de especificar itens por id usando `Item_Id`, você também pode especificá-los por nome usando `val_Item_Name` ou `hex_Hex_Encoded_Item_Name`. Por exemplo:

```
calibre://book-details/Library_Name/authors/val_John%20Doe
```

11.7 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 o calibre

calibre tem um design altamente modular. Várias partes dele podem ser personalizadas. Aqui, você aprenderá:

- como usar variáveis de ambiente e *tweaks* para personalizar o comportamento do calibre,
- como especificar seus próprios recursos estáticos como ícones e modelos para se sobrepor ao padrão
- como usar *plugins* para adicionar funcionalidades ao calibre.
- como compartilhar temas de ícones e plugins com outros usuários calibre.
- ver como criar *receitas* para adicionar novas fontes de conteúdo online ao calibre, visite a Seção *Adicionar seus sites de notícia favoritos ao calibre* (página 33).

Nota

Repare que, embora os temas de ícones e plugins sejam indexados e disponíveis para download via atualizador interno do calibre, eles não fazem parte do calibre, e seus locais canônicos para suporte e código fonte são nos fórums Mobileread[111] em seus tópicos de suporte.

- *Variáveis de ambiente* (página 287)
- *Ajustes* (página 288)
- *Sobrescrevendo ícones, templates, etc* (página 300)
- *Criando seus próprios temas de ícones para o calibre* (página 301)
- *Personalizando o calibre com plugins* (página 301)

[111] 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 256)
- *FileTypePlugin* (página 258)
- *Plugins de Metadados* (página 260)
- *Plugins de Catálogo* (página 261)
- *Plug-ins de download de metadados* (página 262)
- *Plugins de Conversão* (página 265)
- *Drivers de Dispositivo* (página 268)
- *User interface actions* (página 281)
- *Plugins de Preferências* (página 285)

12.1.1 Plugin

class `calibre.customize.`**`Plugin`**(*plugin_path*)

Um plugin calibre. Membros úteis incluem:

- `self.installation_type`: Registra como o plugin foi instalado.
- **`self.plugin_path`: Armazena caminho do arquivo ZIP que contém** este plugin ou nenhum se for um plugin integrado
- **`self.site_customization`: Guarda um texto de personalização inserido** pelo usuário.

Métodos que devem ser sobrescritos nas sub classes:

- *`initialize()`* (página 257)
- *`customization_help()`* (página 258)

Métodos úteis:

- *`temporary_file()`* (página 258)
- `__enter__()`
- *`load_resources()`* (página 257)

`supported_platforms = []`

Lista de plataformas em que este plugin funciona. Por exemplo : " [' windows ', ' osx' , 'linux']"

`name = 'Trivial Plugin'`

O nome deste plugin. É necessário determinar algo além de Plugin Trivial para que ele funcione.

`version = (1, 0, 0)`

A versão deste plugin em 3 partes (versão principal, sub-versão, revisão)

`description = 'Não faz absolutamente nada'`

Uma curta descrição do que este plugin faz

author = 'Desconhecido'

O autor deste plugin

priority = 1

Quando existe mais de um plugin para um tipo de arquivo, os plugins são executados em ordem decrescente de prioridade. Os plugins com prioridade maior serão executados primeiro. A maior prioridade possível é `sys.maxsize`. A prioridade padrão é 1.

minimum_calibre_version = (0, 4, 118)

A menor versão do calibre que este plugin necessita

installation_type = None

A forma como este plugin está instalado

can_be_disabled = True

Se falso, o usuário não será capaz de desativar este plugin. Use com cuidado.

type = 'Base'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

initialize()

Chamado uma vez quando os plugins do calibre são inicializados. Os plugins são reinicializados sempre que um novo plugin é acrescentado. Além disso, repare que se o plugin é executado em um processo de operação, como ao adicionar livros, então o plugin será inicializado para cada novo processo de operação.

Execute qualquer inicialização específica de plugin aqui, como extração de recursos do arquivo ZIP do plugin. O caminho do arquivo ZIP está disponível como `self.plugin_path`.

Repare que `self.site_customization` **não** está disponível neste momento.

config_widget()

Implemente este método e *save_settings()* (página 257) no seu plugin para usar uma janela de configuração personalizada, ao invés de confiar na configuração padrão baseada em strings simples.

Este método, se implementado, deve retornar um QWidget. O widget pode ter um método opcional validate() que não aceita argumentos e é chamado imediatamente depois de o usuário clicar em OK. As alterações são aplicadas se, e somente se, o método retornar Verdadeiro.

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*)

Salvas as configurações especificadas pelo usuário com config_widget.

Parâmetros

config_widget – O widget retornado por *config_widget()* (página 257).

do_user_config(*parent=None*)

This method shows a configuration dialog for this plugin. It returns True if the user clicks OK, False otherwise. The changes are automatically applied.

load_resources(*names*)

If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

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

Retorna
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 retorna ajuda em HTML, caso contrário retorna ajuda em texto simples.

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*)

Base: *Plugin* (página 256)

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 arquivo'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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.

Retorna

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 260) which is called when the book record is created for the first time. This method is called whenever a new file is added to a book record. It is useful for modifying the book record based on the contents of the newly added file.

Parâmetros

- **book_id** – ID no banco 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 no banco 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 no banco 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 259), which is called after a single book file has been added to a book. postadd() is called only when an entire book record with possibly more than one book file has been created for the first time. This is useful if you wish to modify the book record in the database when the book is first added to calibre.

Parâmetros

- **book_id** – ID no banco 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*)

Base: `Plugin` (página 256)

A plugin that implements reading metadata from a set of file types.

file_types = {}

Set of file types for which this plugin should be run. For example: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : " [' windows ', ' osx' , 'linux']"

version = (8, 7, 0)

A versão deste plugin em 3 partes (versão principal, sub-versão, revisão)

author = 'Kovid Goyal'

O autor deste plugin

type = 'Leitor de metadados'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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 260).

Retorna

Um objeto `calibre.ebooks.metadata.book.Metadata`

class calibre.customize.**MetadataWriterPlugin**(**args*, ***kwargs*)

Base: `Plugin` (página 256)

A plugin that implements reading metadata from a set of file types.

file_types = {}

Set of file types for which this plugin should be run. For example: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Lista de plataformas em que este plugin funciona. Por exemplo : " [' windows ', ' osx' , 'linux']"

version = (8, 7, 0)

A versão deste plugin em 3 partes (versão principal, sub-versão, revisão)

author = 'Kovid Goyal'

O autor deste plugin

type = 'Escritor de metadados'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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 260).
- **mi** – Um objeto `calibre.ebooks.metadata.book.Metadata`

12.1.4 Plugins de Catálogo

class calibre.customize.**CatalogPlugin**(*plugin_path*)

Base: `Plugin` (página 256)

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 catálogo'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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 Plug-ins de download de metadados

class calibre.ebooks.metadata.sources.base.**Source**(**args*, ***kwargs*)

Base: *Plugin* (página 256)

type = 'Fonte de metadados'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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 = ()

Uma lista de objetos `Option`. Eles serão utilizados para construir automaticamente o widget de configuração para este plugin

config_help_message = None

Uma string que é exibida no topo do widget de configuração para este 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 retorna ajuda em HTML, caso contrário retorna ajuda em texto simples.

config_widget()

Implemente este método e *save_settings()* (página 263) no seu plugin para usar uma janela de configuração personalizada, ao invés de confiar na configuração padrão baseada em strings simples.

Este método, se implementado, deve retornar um QWidget. O widget pode ter um método opcional validate() que não aceita argumentos e é chamado imediatamente depois de o usuário clicar em OK. As alterações são aplicadas se, e somente se, o método retornar Verdadeiro.

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*)

Salvas as configurações especificadas pelo usuário com config_widget.

Parâmetros

config_widget – O widget retornado por *config_widget()* (página 263).

get_author_tokens(*authors*, *only_first_author=True*)

Take a list of authors and return a list of tokens useful for an AND search query. This function tries to return tokens in first name middle names last name order, by assuming that if a comma is in the author name, the name is in lastname, other names form.

get_title_tokens(*title*, *strip_joiners=True*, *strip_subtitle=False*)

Take a title and return a list of tokens useful for an AND search query. Excludes connectives(optionally) and punctuation.

split_jobs(*jobs*, *num*)

Split a list of jobs into at most num groups, as evenly as possible

test_fields(*mi*)

Return the first field from self.touched_fields that is null on the mi object

clean_downloaded_metadata(*mi*)

Call this method in your plugin's identify method to normalize metadata before putting the Metadata object into result_queue. You can of course, use a custom algorithm suited to your metadata source.

get_book_url(*identifiers*)

Return a 3-tuple or None. The 3-tuple is of the form: (identifier_type, identifier_value, URL). The URL is the URL for the book identified by identifiers at this source. identifier_type, identifier_value specify the identifier corresponding to the URL. This URL must be browsable to by a human using a browser. It is meant to provide a clickable link for the user to easily visit the books page at this source. If no URL is found, return None. This method must be quick, and consistent, so only implement it if it is possible to construct the URL from a known scheme given identifiers.

get_book_url_name(*idtype*, *idval*, *url*)

Return a human readable name from the return value of get_book_url().

get_book_urls(*identifiers*)

Override this method if you would like to return multiple URLs for this book. Return a list of 3-tuples. By default this method simply calls `get_book_url()` (página 263).

get_cached_cover_url(*identifiers*)

Return cached cover URL for the book identified by the identifiers dictionary or None if no such URL exists.

Note that this method must only return validated URLs, i.e. not URLS that could result in a generic cover image or a not found error.

id_from_url(*url*)

Parse a URL and return a tuple of the form: (identifier_type, identifier_value). If the URL does not match the pattern for the metadata source, return None.

identify_results_keygen(*title=None*, *authors=None*, *identifiers={}*)

Return a function that is used to generate a key that can sort Metadata objects by their relevance given a search query (title, authors, identifiers).

These keys are used to sort the results of a call to `identify()` (página 264).

For details on the default algorithm see `InternalMetadataCompareKeyGen` (página 265). 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.

Retorna
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 264). 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 Conversão

class calibre.customize.conversion.**InputFormatPlugin**(**args*)

Base: `Plugin` (página 256)

InputFormatPlugins are responsible for converting a document into HTML+OPF+CSS+etc. The results of the conversion *must* be encoded in UTF-8. The main action happens in `convert()` (página 266).

type = 'Entrada de conversão'
O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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 266) instead. Every option must be an instance of `OptionRecommendation`.

options = {}

Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = {}

A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

get_images()

Return a list of absolute paths to the images, if this input plugin represents an image collection. The list of images is in the same order as the spine and the TOC.

convert(*stream*, *options*, *file_ext*, *log*, *accelerators*)

This method must be implemented in sub-classes. It must return the path to the created OPF file or an `OEBBook` instance. All output should be contained in the current folder. If this plugin creates files outside the current folder they must be deleted/marked for deletion before this method returns.

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** – Um objeto `calibre.utils.logging.Log`. Todas as saídas devem usar este objeto.
- **accelerators** – A dictionary of various information that the input plugin can get easily that would speed up the subsequent stages of the conversion.

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

Called to allow the input plugin to perform postprocessing after the book has been parsed.

specialize(*oeb*, *opts*, *log*, *output_fmt*)

Called to allow the input plugin to specialize the parsed book for a particular output format. Called after postprocess_book and before any transforms are performed on the parsed book.

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)

Chamado para criar o widget usado para configurar este plugin na interface gráfica do calibre. O widget precisa ser uma instância da classe PluginWidget. Veja os plugins de entrada internos para exemplos.

class calibre.customize.conversion.**OutputFormatPlugin**(**args*)

Base: *Plugin* (página 256)

OutputFormatPlugins are responsible for converting an OEB document (OPF+HTML) into an output e-book.

The OEB document can be assumed to be encoded in UTF-8. The main action happens in *convert()* (página 267).

type = 'Saída de conversão'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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 267) instead. Every option must be an instance of `OptionRecommendation`.

options = {}

Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = {}

A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

property description

str(object='') -> str str(bytes_or_buffer[, encoding[, errors]]) -> str

Create a new string object from the given object. If encoding or errors is specified, then the object must expose a data buffer that will be decoded using the given encoding and error handler. Otherwise, returns the result of object.__str__() (if defined) or repr(object). encoding defaults to 'utf-8'. errors defaults to 'strict'.

convert(*oeb_book*, *output*, *input_plugin*, *opts*, *log*)

Render the contents of *oeb_book* (which is an instance of `calibre.ebooks.oeb.OEBBook`) to the file specified by output.

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*)

Chamado para criar o widget usado para configurar este plugin na interface gráfica do calibre. O widget precisa ser uma instância da classe PluginWidget. Veja os plugins de entrada internos para exemplos.

12.1.7 Drivers de Dispositivo

A classe base para todos os drivers de dispositivo é `DevicePlugin` (página 268). 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*)

Base: `Plugin` (página 256)

Defines the interface that should be implemented by backends that communicate with an e-book reader.

type = 'Interface do dispositivo'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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

apelido de `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 271) method after the books lists have been loaded to get the driveinfo.

ASK_TO_ALLOW_CONNECT = False

If set to True, calibre will ask the user if they want to manage the device with calibre, the first time it is detected. If you set this to True you must implement *get_device_uid()* (página 274) and *ignore_connected_device()* (página 274) and *get_user_blacklisted_devices()* (página 274) and *set_user_blacklisted_devices()* (página 274)

user_feedback_after_callback = None

Set this to a dictionary of the form {'title':title, 'msg':msg, 'det_msg':detailed_msg} to have calibre popup a message to the user after some callbacks are run (currently only upload_books). Be careful to not spam the user with too many messages. This variable is checked after *every* callback, so only set it when you really need to.

classmethod get_open_popup_message()

GUI displays this as a non-modal popup. Should be an instance of OpenPopupMessage

classmethod model_metadata() → tuple[ModelMetadata, ...]

Metadata about all the actual device models this driver supports

is_usb_connected(*devices_on_system*, *debug=False*, *only_presence=False*)

Return True, device_info if a device handled by this plugin is currently connected.

Parâmetros

devices_on_system – Lista dos dispositivos conectados atualmente

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 que foram 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 270).

Note that for devices based on USBMS this method by default delegates to *can_handle()* (página 270). So you only need to override *can_handle()* (página 270) 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 270).

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 278) has an implementation of this function that should serve as a good example for USB Mass storage devices.

This method can raise an OpenFeedback exception to display a message to the user.

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}.

Retorna

(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

Retorna
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

Retorna
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.

Retorna
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).

Retorna
A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (página 272).

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 272).
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

delete_books(*paths*, *end_session=True*)

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 271) call.

classmethod config_widget()

Deve retornar um QWidget. O QWidget contém a configuração para a interface do dispositivo

classmethod save_settings(*settings_widget*)

Deveria salvar a configuração em disco. Pega o widget criado em *config_widget()* (página 273) e salva todas as configurações em disco.

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).

Retorna

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()

Retorna verdadeiro se o plug-in foi iniciado, caso contrário 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*)

Base: `list`

Uma lista de livros. Cada objeto de livro deve ter os campos

1. título
2. autores
3. tamanho (tamanho do arquivo 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 baseados em armazenamento em massa USB

A classe base para tais dispositivos é `calibre.devices.usbms.driver.USBMS` (página 279). 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]
    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*)

Base: DeviceConfig, *DevicePlugin* (página 268)

This class provides logic common to all drivers for devices that export themselves as USB Mass Storage devices. Provides implementations for mounting/ejecting of USBMS devices on all platforms.

VENDOR_ID = 0

VENDOR_ID can be either an integer, a list of integers or a dictionary If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0

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

MAX_PATH_LEN = 250

O comprimento máximo do caminhos criado no dispositivo

NEWS_IN_FOLDER = True

Coloque notícias em sua própria pasta

classmethod model_metadata() → tuple[ModelMetadata, ...]

Metadata about all the actual device models this driver supports

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

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

Retorna

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

Retorna

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 278) has an implementation of this function that should serve as a good example for USB Mass storage devices.

This method can raise an OpenFeedback exception to display a message to the user.

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*)

Base: *CLI* (página 279), *Device* (página 276)

The base class for all USBMS devices. Implements the logic for sending/getting/updating metadata/caching metadata/etc.

description = 'Comunicar-se com um leitor de e-books.'

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

apelido de `BookList`

book_class

apelido de `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}.

Retorna

(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.

Retorna

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).

Retorna

A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (página 281).

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 280).
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth`books(oncard='cardb')`).

delete_books(*paths*, *end_session=True*)

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 `InterfaceActionBase` 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*)

Base: `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 281) objects have the same name, the one with higher priority takes precedence.

Sub-classes should implement the *genesis()* (página 284), *library_changed()* (página 284), *location_selected()* (página 284), *shutting_down()* (página 284), *initialization_complete()* (página 284) and *tag_browser_context_action()* (página 284) methods.

Once initialized, this plugin has access to the main calibre GUI via the `gui` member. You can access other plugins by name, for example:

```
self.gui.iactions['Save To Disk']
```

To access the actual plugin, use the `interface_action_base_plugin` attribute, this attribute only becomes available after the plugin has been initialized. Useful if you want to use methods from the plugin class like do_user_config().

The QAction specified by *action_spec* (página 282) is automatically create and made available as `self.qaction`.

name = 'Implement me'

The plugin name. If two plugins with the same name are present, the one with higher priority takes precedence.

priority = 1

The plugin priority. If two plugins with the same name are present, the one with higher priority takes precedence.

popup_type = 1

The menu popup type for when this plugin is added to a toolbar

auto_repeat = False

Whether this action should be auto repeated when its shortcut key is held down.

action_spec = ('text', 'icon', None, None)

Of the form: (text, icon_path, tooltip, keyboard shortcut). icon, tooltip and keyboard shortcut can be None. keyboard shortcut must be either a string, None or tuple of shortcuts. If None, a keyboard shortcut corresponding to the action is not registered. If you pass an empty tuple, then the shortcut is registered with no default key binding.

action_shortcut_name = None

If not None, used for the name displayed to the user when customizing the keyboard shortcuts for the above action spec instead of action_spec[0]

action_add_menu = False

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 283), :meth`:accept_drag_move_event`, `drop_event()` (página 283) for details.

accept_enter_event(*event*, *mime_data*)

This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

accept_drag_move_event(*event*, *mime_data*)

This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

drop_event(*event*, *mime_data*)

This method should perform some useful action and return True iff this interface action is capable of handling the drop event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI. You should not perform blocking/long operations in this function. Instead emit a signal or use QTimer.singleShot and return quickly. See the builtin actions for examples.

create_menu_action(*menu*, *unique_name*, *text*, *icon=None*, *shortcut=None*, *description=None*, *triggered=None*, *shortcut_name=None*, *persist_shortcut=False*)

Convenience method to easily add actions to a QMenu. Returns the created QAction. This action has one extra attribute calibre_shortcut_unique_name which if not None refers to the unique name under which this action is registered with the keyboard manager.

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 da 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

Retorna
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 282) is available as `self.qaction`.

location_selected(*loc*)

Called whenever the book list being displayed in calibre changes. Currently values for loc are: `library, main, card and cardb`.

This method should enable/disable this action and its sub actions as appropriate for the location.

library_about_to_change(*olddb*, *db*)

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 284).

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*)

Base: *Plugin* (página 256)

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 = 'Ação da interface do usuário'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

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 Plugins de Preferências

class calibre.customize.**PreferencesPlugin**(*plugin_path*)

Base: *Plugin* (página 256)

Um plugin representando um widget exibido na caixa de diálogo Preferências.

Este plugin tem apenas um método importante, *create_widget()* (página 286). Os vários campos do plugin controlam como ele é categorizado na interface gráfica.

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'

O tipo deste plugin. Usado para definir categorias de plugins na interface gráfica

can_be_disabled = False

Se falso, o usuário não será capaz de desativar este plugin. Use com cuidado.

config_widget = None

Caminho de importação para módulo que contém uma classe chamada ConfigWidget, que implementa ConfigWidgetInterface. Usado por *create_widget()* (página 286).

category_order = 100

Where in the list of categories the *category* (página 285) of this plugin should be.

name_order = 100

Where in the list of names in a category, the *gui_name* (página 285) 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*)

Cria e retorna o widget Qt usado efetivamente para configurar este grupo de preferências. o widget precisa implementar a *calibre.gui2.preferences.ConfigWidgetInterface* (página 286).

A implementação padrão usa *config_widget* (página 285) para instanciar o widget.

class calibre.gui2.preferences.**ConfigWidgetInterface**

Esta classe define a interface que todos os widgets exibidos na caixa de diálogo Preferências precisam implementar. Veja *ConfigWidgetBase* (página 286) para uma classe básica que implementa esta interface e define vários métodos de conveniência também.

changed_signal = None

Este sinal deve ser emitido sempre que o usuário alterar um valor neste widget

supports_restoring_to_defaults = True

Set to True iff the `restore_to_defaults()` method is implemented.

restore_defaults_desc = 'Restaurar configurações para valores padrão. Você tem que clicar em Aplicar para realmente salvar os configurações padrão.'

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 286) returns True.

genesis(*gui*)

Chamado uma vez antes de o widget ser exibido, deveria executar qualquer ajuste necessário.

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*)

Chamado uma vez após o widget ser finalizado. Responsável por fazer com que a interface gráfica leia novamente qualquer configuração alterada. Repare que, por padrão, a interface gráfica reinicializa vários elementos, então a maioria dos widgets não precisarão usar este método.

initial_tab_changed()

Chamado se a aba exibida inicialmente for alterada antes de o widget ser exibido, mas depois de ser inicializado.

class calibre.gui2.preferences.**ConfigWidgetBase**(*parent=None*)

Classe base que contém código para adicionar com rapidez widgets de configuração padrão, como caixas de seleção, caixas de combo, campos de texto e afins. Veja o método `register()` (página 287).

This class automatically handles change notification, resetting to default, translation between gui objects and config objects, etc. for registered settings.

Se o seu widget de configuração herdar desta classe, mas incluir configuração que não está registrada, você precisa ignorar os métodos `ConfigWidgetInterface` (página 286) e chamar os métodos da classe base dentro dos substitutos.

changed_signal

Este sinal deve ser emitido sempre que o usuário alterar um valor neste 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 287) returns True.

register(*name*, *config_obj*, *gui_name=None*, *choices=None*, *restart_required=False*, *empty_string_is_None=True*, *setting=<class 'calibre.gui2.preferences.Setting'>*)

Register a setting.

Parâmetros

- **name** – The setting name
- **config_obj** – The config object that reads/writes the setting
- **gui_name** – The name of the GUI object that presents an interface to change the setting. By default it is assumed to be `'opt_' + name`.
- **choices** – If this setting is a multiple choice (combobox) based setting, the list of choices. The list is a list of two element tuples of the form: `[(gui name, value), ...]`
- **setting** – The class responsible for managing this setting. The default class handles almost all cases, so this param is rarely used.

initialize()

Should set all config values to their initial values (the values stored in the config files). A "return" statement is optional. Return False if the dialog is not to be shown.

commit(**args*)

Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an `AbortCommit` exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

restore_defaults(**args*)

Should set all config values to their defaults.

12.2 Variáveis de ambiente

- `CALIBRE_CONFIG_DIRECTORY` - define a pasta onde os arquivos de configuração são armazenados/lidos.
- `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 arquivo metadata.db. Usar esta variável permite-lhe ter o arquivo metadata.db numa localização diferente da pasta onde se encontra a biblioteca. É útil se a sua pasta da biblioteca se encontra em um drive compartilhado em rede que não tenha suporte para acesso exclusivo a arquivos.
- `CALIBRE_DEVELOP_FROM` - usado para executar o calibre a partir de um ambiente de desenvolvimento . Veja *Configurando um ambiente de desenvolvimento do calibre* (página 343).
- `CALIBRE_OVERRIDE_LANG` - usado para forçar o idioma usado pela interface (ISO 639 códigos de idiomas)
- `CALIBRE_TEST_TRANSLATION` - usado para testar um arquivo de tradução .po (deve conter o caminho para o arquivo .po)
- `CALIBRE_NO_NATIVE_FILEDIALOGS` - faz com que o calibre não utilize o arquivo de diálogos nativo para a seleção de arquivos/pastas.
- `CALIBRE_NO_NATIVE_MENUBAR` - faz com que o calibre não crie um menu nativo (global) no Ubuntu Unity e outros ambiente Linux similares. Ao invés disso, o menu é colocado dentro da janela, como é feito tradicionalmente.
- `CALIBRE_USE_SYSTEM_THEME` - por default, no Linux, calibre usa seu próprio Qt style interno. Isto serve para evitar problemas ou panes causados por incompatibilidades entre a versão do Qt do calibre e a versão do sistema. A desvantagem é que o calibre pode não seguir a configuração do sistema. e você configurar esta variável de ambiente no Linux, ela vai fazer com que o calibre use o tema do sistema – tenha cuidado com os problemas e panes.
- `CALIBRE_SHOW_DEPRECATION_WARNINGS` - faz com que o calibre exiba avisos de descontinuação para stdout. Útil para desenvolvedores do calibre.
- `CALIBRE_NO_DEFAULT_PROGRAMS` - impede que o calibre registre automaticamente os tipos de arquivos que ele é capaz de interpretar no Windows.
- `CALIBRE_USE_SYSTEM_CERTIFICATES` - faz com que o calibre use o repositório de certificados do sistema para validação de certificados SSL ao invés de seu próprio repositório de certificados para Windows e macOS.
- `CALIBRE_NO_ICONS_IN_MENUS` - Desabilitar ícones em menus
- `QT_QPA_PLATFORM` - No Linux, configure para `wayland` para forçar o calibre a usar o protocolo Wayland e `xcb` para forçar o uso do protocolo X11.
- `SYSFS_PATH` - Utilizar se o sysfs está montado em outro local da árvore de arquivos que não em /sys
- `http_proxy`, `https_proxy` - usado no Linux para especificar um proxy HTTP(S)

Veja Como definir variáveis de ambiente no Windows[112]. 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
```

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>
```

(continua na próxima página)

[112] https://www.computerhope.com/issues/ch000549.htm

(continuação da página anterior)

```
# Contains various tweaks that affect calibre behavior. Only edit this file if
# you know what you are doing. If you delete this file, it will be recreated from
# defaults.

#: Auto increment series index
# The algorithm used to assign a book added to an existing series a series number.
# New series numbers assigned using this tweak are always integer values, except
# if a constant non-integer is specified.
# Possible values are:
#   next - First available integer larger than the largest existing number
#   first_free - First available integer larger than 0
#   next_free - First available integer larger than the smallest existing number
#   last_free - First available integer smaller than the largest existing number.␣
↪Return largest existing + 1 if no free number is found
#   const - Assign the number 1 always
#   no_change - Do not change the series index
#   a number - Assign that number always. The number is not in quotes. Note that 0.0␣
↪can be used here.
# Examples:
#   series_index_auto_increment = 'next'
#   series_index_auto_increment = 'next_free'
#   series_index_auto_increment = 16.5
#
# Set the use_series_auto_increment_tweak_when_importing tweak to True to
# use the above values when importing/adding books. If this tweak is set to
# False (the default) then the series number will be set to 1 if it is not
# explicitly set during the import. If set to True, then the
# series index will be set according to the series_index_auto_increment setting.
# Note that the use_series_auto_increment_tweak_when_importing tweak is used
# only when a value is not provided during import. If the importing regular
# expression produces a value for series_index, or if you are reading metadata
# from books and the import plugin produces a value, then that value will
# be used irrespective of the setting of the tweak.
series_index_auto_increment = 'next'
use_series_auto_increment_tweak_when_importing = False

#: Add separator after completing an author name
# Set this if the completion separator should be appended to the end of the
# completed text to automatically begin a new completion operation for authors.
# It can be either True or False
authors_completer_append_separator = False

#: Author sort name algorithm
# The algorithm used to copy author to author_sort.
# Possible values are:
#  invert: use "fn ln" -> "ln, fn"
#  copy  : copy author to author_sort without modification
#  comma : use 'copy' if there is a ',' in the name, otherwise use 'invert'
#  nocomma : "fn ln" -> "ln fn" (without the comma)
# When this tweak is changed, the author_sort values stored with each author
```

(continua na próxima página)

(continuação da página anterior)

```
# must be recomputed by right-clicking on an author in the left-hand tags
# panel, selecting 'Manage authors', and pressing
# 'Recalculate all author sort values'.
#
# The author_name_suffixes are words that are ignored when they occur at the
# end of an author name. The case of the suffix is ignored and trailing
# periods are automatically handled.
#
# The same is true for author_name_prefixes.
#
# The author_name_copywords are a set of words which, if they occur in an
# author name, cause the automatically generated author sort string to be
# identical to the author's name. This means that the sort for a string like
# "Acme Inc." will be "Acme Inc." instead of "Inc., Acme".
#
# If author_use_surname_prefixes is enabled, any of the words in
# author_surname_prefixes will be treated as a prefix to the surname, if they
# occur before the surname. So for example, "John von Neumann" would be sorted
# as "von Neumann, John" and not "Neumann, John von".
author_sort_copy_method = 'comma'
author_name_suffixes = ('Jr', 'Sr', 'Inc', 'Ph.D', 'Phd',
                        'MD', 'M.D', 'I', 'II', 'III', 'IV',
                        'Junior', 'Senior')
author_name_prefixes = ('Mr', 'Mrs', 'Ms', 'Dr', 'Prof')
author_name_copywords = (
    'Agency', 'Corporation', 'Company', 'Co.', 'Council',
    'Committee', 'Inc.', 'Institute', 'National', 'Society', 'Club', 'Team',
    'Software', 'Games', 'Entertainment', 'Media', 'Studios',
)
author_use_surname_prefixes = False
author_surname_prefixes = ('da', 'de', 'di', 'la', 'le', 'van', 'von')

#: Splitting multiple author names
# By default, calibre splits a string containing multiple author names on
# ampersands and the words "and" and "with". You can customize the splitting
# by changing the regular expression below. Strings are split on whatever the
# specified regular expression matches, in addition to ampersands.
# Default: r'(?i),?\s+(and|with)\s+'
authors_split_regex = r'(?i),?\s+(and|with)\s+'

#: Use author_sort and/or series_sort for names in Tag browser
# Set which author or series field is used to display as the item name in the
# Tag browser (the list of authors, series, publishers etc on the left hand
# side). The choices for author are 'author' and 'author_sort'. The choices for
# series are 'series' and 'series_sort'. This tweak affects only what is
# displayed under the category in the Tag browser and Content server. Please
# note that if you set this to …_sort, it is possible to see duplicate names in
# the list because although it is guaranteed that author and series names are
# unique, there is no such guarantee for their sort values. Showing duplicates
# won't break anything but it could lead to some confusion. The tooltip for an
# item will show the item's name no matter how you set this tweak.
# Examples:
```

(continua na próxima página)

(continuação da página anterior)

```
#   categories_use_field_for_author_name = 'author'
#   categories_use_field_for_author_name = 'author_sort'
#   categories_use_field_for_series_name = 'series'
#   categories_use_field_for_series_name = 'series_sort'
categories_use_field_for_author_name = 'author'
categories_use_field_for_series_name = 'series'

#: Control partitioning of Tag browser
# When partitioning the Tag browser, the format of the subcategory label is
# controlled by a template: categories_collapsed_name_template if sorting by
# name, categories_collapsed_rating_template if sorting by average rating, and
# categories_collapsed_popularity_template if sorting by popularity. There are
# two variables available to the template: first and last. The variable 'first'
# is the initial item in the subcategory, and the variable 'last' is the final
# item in the subcategory. Both variables are 'objects'; they each have multiple
# values that are obtained by using a suffix. For example, first.name for an
# author category will be the name of the author. The sub-values available are:
#  name: the printable name of the item
#  count: the number of books that references this item
#  avg_rating: the average rating of all the books referencing this item
#  sort: the sort value. For authors, this is the author_sort for that author
#  category: the category (e.g., authors, series) that the item is in.
# Note that the "r'" in front of the { is necessary if there are backslashes
# (\ characters) in the template. It doesn't hurt anything to leave it there
# even if there aren't any backslashes.
categories_collapsed_name_template = r'{first.sort:shorten(4,,0)} - {last.
↪sort:shorten(4,,0)}'
categories_collapsed_rating_template = r'{first.avg_rating:4.2f:ifempty(0)} - {last.
↪avg_rating:4.2f:ifempty(0)}'
categories_collapsed_popularity_template = r'{first.count:d} - {last.count:d}'

#: Specify columns to sort the booklist by on startup
# Provide a set of columns to be sorted on when calibre starts.
# The argument is None if saved sort history is to be used
# otherwise it is a list of column,order pairs. Column is the
# lookup/search name, found using the tooltip for the column
# Order is 0 for ascending, 1 for descending.
# For example, set it to [('authors',0),('title',0)] to sort by
# title within authors.
sort_columns_at_startup = None

#: Control how dates are displayed
# Format to be used for publication date and the timestamp (date).
#  A string controlling how the publication date is displayed in the GUI
#  d     the day as number without a leading zero (1 to 31)
#  dd    the day as number with a leading zero (01 to 31)
#  ddd   the abbreviated localized day name (e.g. 'Mon' to 'Sun')
#  dddd  the long localized day name (e.g. 'Monday' to 'Sunday')
#  M     the month as number without a leading zero (1-12)
#  MM    the month as number with a leading zero (01-12)
#  MMM   the abbreviated localized month name (e.g. 'Jan' to 'Dec')
#  MMMM  the long localized month name (e.g. 'January' to 'December')
```

(continua na próxima página)

(continuação da página anterior)

```
#  yy    the year as two digit number (00-99)
#  yyyy  the year as four digit number
#  h     the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
#  hh    the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
#  m     the minutes without a leading 0 (0 to 59)
#  mm    the minutes with a leading 0 (00 to 59)
#  s     the seconds without a leading 0 (0 to 59)
#  ss    the seconds with a leading 0 (00 to 59)
#  ap    use a 12-hour clock instead of a 24-hour clock, with "ap" replaced by the␣
↪lowercase localized string for am or pm
#  AP    use a 12-hour clock instead of a 24-hour clock, with "AP" replaced by the␣
↪uppercase localized string for AM or PM
#  aP    use a 12-hour clock instead of a 24-hour clock, with "aP" replaced by the␣
↪localized string for am or pm
#  Ap    use a 12-hour clock instead of a 24-hour clock, with "Ap" replaced by the␣
↪localized string for AM or PM
#  iso   the date with time and timezone. Must be the only format present
#  For example, given the date of 9 Jan 2010, the following formats show
#  MMM yyyy ==> Jan 2010    yyyy ==> 2010       dd MMM yyyy ==> 09 Jan 2010
#  MM/yyyy ==> 01/2010      d/M/yy ==> 9/1/10   yy ==> 10
#
# publication default if not set: MMM yyyy
# timestamp default if not set: dd MMM yyyy
# last_modified_display_format if not set: dd MMM yyyy
gui_pubdate_display_format = 'MMM yyyy'
gui_timestamp_display_format = 'dd MMM yyyy'
gui_last_modified_display_format = 'dd MMM yyyy'

#: Control sorting of titles and series in the library display
# Control title and series sorting in the library view. If set to
# 'library_order', the title sort field will be used instead of the title.
# Unless you have manually edited the title sort field, leading articles such as
# The and A will be ignored. If set to 'strictly_alphabetic', the titles will be
# sorted as-is (sort by title instead of title sort). For example, with
# library_order, The Client will sort under 'C'. With strictly_alphabetic, the
# book will sort under 'T'.
# This flag affects calibre's library display. It has no effect on devices. In
# addition, titles for books added before changing the flag will retain their
# order until the title is edited. Editing a title and hitting Enter
# without changing anything is sufficient to change the sort. Or you can use
# the 'Update title sort' action in the Bulk metadata edit dialog to update
# it for many books at once.
title_series_sorting = 'library_order'

#: Control formatting of title and series when used in templates
# Control how title and series names are formatted when saving to disk/sending
# to device. The behavior depends on the field being processed. If processing
# title, then if this tweak is set to 'library_order', the title will be
# replaced with title_sort. If it is set to 'strictly_alphabetic', then the
# title will not be changed. If processing series, then if set to
# '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.
```

(continua na próxima página)

(continuação da página anterior)

```
# For example, if the tweak is set to library_order, "The Lord of the Rings"
# will become "Lord of the Rings, The". If the tweak is set to
# strictly_alphabetic, it would remain "The Lord of the Rings". Note that the
# formatter function raw_field will return the base value for title and
# series regardless of the setting of this tweak.
save_template_title_series_sorting = 'library_order'

#: Set the list of words considered to be "articles" for sort strings
# Set the list of words that are to be considered 'articles' when computing the
# title sort strings. The articles differ by language. By default, calibre uses
# a combination of articles from English and whatever language the calibre user
# interface is set to. In addition, in some contexts where the book language is
# available, the language of the book is used. You can change the list of
# articles for a given language or add a new language by editing
# per_language_title_sort_articles. To tell calibre to use a language other
# than the user interface language, set, default_language_for_title_sort. For
# example, to use German, set it to 'deu'. A value of None means the user
# interface language is used. The setting title_sort_articles is ignored
# (present only for legacy reasons).
per_language_title_sort_articles = {
        # English
        'eng': (r'A\s+', r'The\s+', r'An\s+'),
        # Esperanto
        'epo': (r'La\s+', r"L'", 'L´'),
        # Spanish
        'spa': (r'El\s+', r'La\s+', r'Lo\s+', r'Los\s+', r'Las\s+', r'Un\s+',
                r'Una\s+', r'Unos\s+', r'Unas\s+'),
        # French
        'fra': (r'Le\s+', r'La\s+', r"L'", r'L´', r'L’', r'Les\s+', r'Un\s+', r'Une\s+
↪',
                r'Des\s+', r'De\s+La\s+', r'De\s+', r"D'", r'D´', r'D’'),
        # Polish
        'pol': (),
        # Italian
        'ita': (r'Lo\s+', r'Il\s+', r"L'", r'L´', r'La\s+', r'Gli\s+',
                r'I\s+', r'Le\s+', r'Uno\s+', r'Un\s+', r'Una\s+', "rUn'",
                r'Un´', r'Dei\s+', r'Degli\s+', r'Delle\s+', r'Del\s+',
                r'Della\s+', r'Dello\s+', r"Dell'", r'Dell´'),
        # Portuguese
        'por': (r'A\s+', r'O\s+', r'Os\s+', r'As\s+', r'Um\s+', r'Uns\s+',
                r'Uma\s+', r'Umas\s+'),
        # Romanian
        'ron': (r'Un\s+', r'O\s+', r'Nişte\s+'),
        # German
        'deu': (r'Der\s+', r'Die\s+', r'Das\s+', r'Den\s+', r'Ein\s+',
                r'Eine\s+', r'Einen\s+', r'Dem\s+', r'Des\s+', r'Einem\s+',
                r'Eines\s+'),
        # Dutch
        'nld': (r'De\s+', r'Het\s+', r'Een\s+', r"'n\s+", r"'s\s+", r'Ene\s+',
                r'Ener\s+', r'Enes\s+', r'Den\s+', r'Der\s+', r'Des\s+',
                r"'t\s+"),
        # Swedish
```

(continua na próxima página)

(continuação da página anterior)

```
        'swe': (r'En\s+', r'Ett\s+', r'Det\s+', r'Den\s+', r'De\s+'),
        # Turkish
        'tur': (r'Bir\s+',),
        # Afrikaans
        'afr': (r"'n\s+", r'Die\s+'),
        # Greek
        'ell': (r'O\s+', r'I\s+', r'To\s+', r'Ta\s+', r'Tus\s+', r'Tis\s+',
                r"'Enas\s+", r"'Mia\s+", r"'Ena\s+", r"'Enan\s+"),
        # Hungarian
        'hun': (r'A\s+', r'Az\s+', r'Egy\s+'),
}
default_language_for_title_sort = None
title_sort_articles = r'^(A|The|An)\s+'

#: Specify a folder calibre should connect to at startup
# Specify a folder that calibre should connect to at startup using
# connect_to_folder. This must be a full path to the folder. If the folder does
# not exist when calibre starts, it is ignored.
# Example for Windows:
#     auto_connect_to_folder = 'C:/Users/someone/Desktop/testlib'
# Example for other operating systems:
#     auto_connect_to_folder = '/home/dropbox/My Dropbox/someone/library'
auto_connect_to_folder = ''

#: Specify renaming rules for SONY collections
# Specify renaming rules for SONY collections. This tweak is only applicable if
# metadata management is set to automatic. Collections on SONYs are named
# depending upon whether the field is standard or custom. A collection derived
# from a standard field is named for the value in that field.
#
# For example, if the standard 'series' column contains the value 'Darkover', then the
# collection name is 'Darkover'. A collection derived from a custom field will
# have the name of the field added to the value. For example, if a custom series
# column named 'My Series' contains the name 'Darkover', then the collection
# will by default be named 'Darkover (My Series)'. For purposes of this
# documentation, 'Darkover' is called the value and 'My Series' is called the
# category. If two books have fields that generate the same collection name,
# then both books will be in that collection.
#
# This set of tweaks lets you specify for a standard or custom field how
# the collections are to be named. You can use it to add a description to a
# standard field, for example 'Foo (Tag)' instead of the 'Foo'. You can also use
# it to force multiple fields to end up in the same collection.
#
# For example, you could force the values in 'series', '#my_series_1', and
# '#my_series_2' to appear in collections named 'some_value (Series)', thereby
# merging all of the fields into one set of collections.
#
# There are two related tweaks. The first determines the category name to use
# for a metadata field.  The second is a template, used to determines how the
# value and category are combined to create the collection name.
# The syntax of the first tweak, sony_collection_renaming_rules, is:
```

(continua na próxima página)

(continuação da página anterior)

```
# {'field_lookup_name':'category_name_to_use', 'lookup_name':'name', ...}
#
# The second tweak, sony_collection_name_template, is a template. It uses the
# same template language as plugboards and save templates. This tweak controls
# how the value and category are combined together to make the collection name.
# The only two fields available are {category} and {value}. The {value} field is
# never empty. The {category} field can be empty. The default is to put the
# value first, then the category enclosed in parentheses, it isn't empty:
# '{value} {category:|(|)}'
#
# Examples: The first three examples assume that the second tweak
# has not been changed.
#
# 1) I want three series columns to be merged into one set of collections. The
# column lookup names are 'series', '#series_1' and '#series_2'. I want nothing
# in the parenthesis. The value to use in the tweak value would be:
#    sony_collection_renaming_rules={'series':'', '#series_1':'', '#series_2':''}
#
# 2) I want the word '(Series)' to appear on collections made from series, and
# the word '(Tag)' to appear on collections made from tags. Use:
#   sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#
# 3) I want 'series' and '#myseries' to be merged, and for the collection name
# to have '(Series)' appended. The renaming rule is:
#   sony_collection_renaming_rules={'series':'Series', '#myseries':'Series'}
#
# 4) Same as example 2, but instead of having the category name in parentheses
# and appended to the value, I want it prepended and separated by a colon, such
# as in Series: Darkover. I must change the template used to format the category name
#
# The resulting two tweaks are:
#    sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#    sony_collection_name_template='{category:||: }{value}'
sony_collection_renaming_rules = {}
sony_collection_name_template = '{value}{category:| (|)}'

#: Specify how SONY collections are sorted
# Specify how SONY collections are sorted. This tweak is only applicable if
# metadata management is set to automatic. You can indicate which metadata is to
# be used to sort on a collection-by-collection basis. The format of the tweak
# is a list of metadata fields from which collections are made, followed by the
# name of the metadata field containing the sort value.
# Example: The following indicates that collections built from pubdate and tags
# are to be sorted by the value in the custom column '#mydate', that collections
# built from 'series' are to be sorted by 'series_index', and that all other
# collections are to be sorted by title. If a collection metadata field is not
# named, then if it is a series- based collection it is sorted by series order,
# otherwise it is sorted by title order.
# [(['pubdate', 'tags'],'#mydate'), (['series'],'series_index'), (['*'], 'title')]
# Note that the bracketing and parentheses are required. The syntax is
# [ ( [list of fields], sort field ) , ( [ list of fields ] , sort field ) ]
# Default: empty (no rules), so no collection attributes are named.
```

(continua na próxima página)

(continuação da página anterior)

```
sony_collection_sorting_rules = []

#: Control how tags are applied when copying books to another library
# Set this to True to ensure that tags in 'Tags to add when adding
# a book' are added when copying books to another library
add_new_book_tags_when_importing_books = False

#: Set the maximum number of sort 'levels'
# Set the maximum number of sort 'levels' that calibre will use to resort the
# library after certain operations such as searches or device insertion. Each
# sort level adds a performance penalty. If the database is large (thousands of
# books) the penalty might be noticeable. If you are not concerned about multi-
# level sorts, and if you are seeing a slowdown, reduce the value of this tweak.
maximum_resort_levels = 5

#: Choose whether dates are sorted using visible fields
# Date values contain both a date and a time. When sorted, all the fields are
# used, regardless of what is displayed. Set this tweak to True to use only
# the fields that are being displayed.
sort_dates_using_visible_fields = False

#: Fuzz value for trimming covers
# The value used for the fuzz distance when trimming a cover.
# Colors within this distance are considered equal.
# The distance is in absolute intensity units.
cover_trim_fuzz_value = 10

#: Control behavior of the book list
# You can control the behavior of double clicks and pressing Enter on the books
# list. Choices: open_viewer, do_nothing, show_book_details,
# show_locked_book_details, edit_cell, edit_metadata. Selecting anything other
# than open_viewer, show_book_details, or show_locked_book_details has the side
# effect of disabling editing a field using a single click.
# Default: open_viewer.
# Example: doubleclick_on_library_view = 'do_nothing'
# You can also control whether the book list scrolls per item or
# per pixel. Default is per item.
doubleclick_on_library_view = 'open_viewer'
enter_key_behavior = 'do_nothing'
horizontal_scrolling_per_column = False
vertical_scrolling_per_row = False

#: Language to use when sorting
# Setting this tweak will force sorting to use the
# collating order for the specified language. This might be useful if you run
# calibre in English but want sorting to work in the language where you live.
# Set the tweak to the desired ISO 639-1 language code, in lower case.
# You can find the list of supported locales at
# https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
# Default: locale_for_sorting = '' -- use the language calibre displays in
# Example: locale_for_sorting = 'fr' -- sort using French rules.
# Example: locale_for_sorting = 'nb' -- sort using Norwegian rules.
```

(continua na próxima página)

(continuação da página anterior)

```
locale_for_sorting = ''

#: The number of seconds to wait before sending emails
# The number of seconds to wait before sending emails when using a
# public email server like GMX/Hotmail/Gmail. Default is: 5 minutes
# Setting it to lower may cause the server's SPAM controls to kick in,
# making email sending fail. Changes will take effect only after a restart of
# calibre. You can also change the list of hosts that calibre considers
# to be public relays here. Any relay host ending with one of the suffixes
# in the list below will be considered a public email server.
public_smtp_relay_delay = 301
public_smtp_relay_host_suffixes = ['gmail.com', 'live.com', 'gmx.com', 'outlook.com']

#: The maximum width and height for covers saved in the calibre library
# All covers in the calibre library will be resized, preserving aspect ratio,
# to fit within this size. This is to prevent slowdowns caused by extremely
# large covers
maximum_cover_size = (1650, 2200)

#: Where to send downloaded news
# When automatically sending downloaded news to a connected device, calibre
# will by default send it to the main memory. By changing this tweak, you can
# control where it is sent. Valid values are "main", "carda", "cardb". Note
# that if there isn't enough free space available on the location you choose,
# the files will be sent to the location with the most free space.
send_news_to_device_location = 'main'

#: Unified toolbar on macOS
# If you enable this option and restart calibre, the toolbar will be 'unified'
# with the titlebar as is normal for macOS applications. However, doing this has
# various bugs, for instance the minimum width of the toolbar becomes twice
# what it should be and it causes other random bugs on some systems, so turn it
# on at your own risk!
unified_title_toolbar_on_osx = False

#: Save original file when converting/polishing from same format to same format
# When calibre does a conversion from the same format to the same format, for
# example, from EPUB to EPUB, the original file is saved, so that in case the
# conversion is poor, you can tweak the settings and run it again. By setting
# this to False you can prevent calibre from saving the original file.
# Similarly, by setting save_original_format_when_polishing to False you can
# prevent calibre from saving the original file when polishing.
save_original_format = True
save_original_format_when_polishing = True

#: Number of recently viewed books to show
# Right-clicking the "View" button shows a list of recently viewed books. Control
# how many should be shown, here.
gui_view_history_size = 15

#: 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
```

(continua na próxima página)

(continuação da página anterior)

```
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0

#: What format to default to when using the "Unpack book" feature
# The "Unpack book" feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
# to that format rather than your output format preference.
# Set to a value of 'remember' to use whichever format you chose last time you
# used the "Unpack book" feature.
# Examples:
#   default_tweak_format = None       (Use output format)
#   default_tweak_format = 'EPUB'
#   default_tweak_format = 'remember'
default_tweak_format = None

#: Do not preselect a completion when editing authors/tags/series/etc.
# This means that you can make changes and press Enter and your changes will
# not be overwritten by a matching completion. However, if you wish to use the
# completions you will now have to press Tab to select one before pressing
# Enter. Which technique you prefer will depend on the state of metadata in
# your library and your personal editing style.
#
# If preselect_first_completion is False and you want Tab to accept what you
# typed instead of the first completion then set tab_accepts_uncompleted_text
# to True. If you do this then to select from the completions you must press
# the Down or Up arrow keys. The tweak tab_accepts_uncompleted_text is ignored
# if preselect_first_completion is True
preselect_first_completion = False
tab_accepts_uncompleted_text = False

#: Completion mode when editing authors/tags/series/etc.
# By default, when completing items, calibre will show you all the candidates
# that start with the text you have already typed. You can instead have it show
# all candidates that contain the text you have already typed. To do this, set
# completion_mode to 'contains'. For example, if you type asi it will match both
# Asimov and Quasimodo, whereas the default behavior would match only Asimov.
completion_mode = 'prefix'

#: Sort the list of libraries alphabetically
# The list of libraries in the Copy to library and Quick switch menus are
# normally sorted by most used. However, if there are more than a certain
# number of such libraries, the sorting becomes alphabetic. You can set that
# number here. The default is ten libraries.
many_libraries = 10

#: Choose available output formats for conversion
# Restrict the list of available output formats in the conversion dialogs.
# 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
```

(continua na próxima página)

(continuação da página anterior)

```
# all available output formats to be present.
restrict_output_formats = None

#: Set the thumbnail image quality used by the Content server
# The quality of a thumbnail is largely controlled by the compression quality
# used when creating it. Set this to a larger number to improve the quality.
# Note that the thumbnails get much larger with larger compression quality
# numbers.
# The value can be between 50 and 99
content_server_thumbnail_compression_quality = 75

#: Image file types to treat as e-books when dropping onto the "Book details" panel
# Normally, if you drop any image file in a format known to calibre onto the
# "Book details" panel, it will be used to set the cover. If you want to store
# some image types as e-books instead, you can set this tweak.
# Examples:
#    cover_drop_exclude = {'tiff', 'webp'}
cover_drop_exclude = ()

#: Exclude fields when copy/pasting metadata
# You can ask calibre to not paste some metadata fields when using the
# Edit metadata->Copy metadata/Paste metadata actions. For example,
# exclude_fields_on_paste = ['cover', 'timestamp', '#mycolumn']
# to prevent pasting of the cover, Date and custom column, mycolumn.
# You can also add a shortcut in Preferences->Shortcuts->Edit metadata
# to paste metadata ignoring this tweak.
exclude_fields_on_paste = []

#: Skip internet connected check
# Skip checking whether the internet is available before downloading news.
# Useful if for some reason your operating systems network checking
# facilities are not reliable (for example NetworkManager on Linux).
skip_network_check = False

#: Tab stop width in the template editor
# Sets the width of the tab stop in the template editor in "average characters".
# For example, a value of 1 results in a space with the width of one average␣
↪character.
template_editor_tab_stop_width = 4

#: Value for undefined numbers when sorting
# Sets the value to use for undefined numbers when sorting.
# For example, the value -10 sorts undefined numbers as if they were set to -10.
# Use 'maximum' for the largest possible number. Use 'minimum' for the smallest
# possible number. Quotes are optional if entering a number.
# Examples:
#   value_for_undefined_numbers_when_sorting = -100
#   value_for_undefined_numbers_when_sorting = '2'
#   value_for_undefined_numbers_when_sorting = -0.01
#   value_for_undefined_numbers_when_sorting = 'minimum'
#   value_for_undefined_numbers_when_sorting = 'maximum'
value_for_undefined_numbers_when_sorting = 0
```

(continua na próxima página)

(continuação da página anterior)

```
#: Allow template database functions in composite columns
# If True then the template database functions book_values() and book_count()
# can be used in composite custom columns. Note: setting this tweak to True and
# using these functions in composites can be very slow.
# Default: False
allow_template_database_functions_in_composites = False


#: Change the programs that are run when opening files/URLs
# By default, calibre passes URLs to the operating system to open using
# whatever default programs are configured there. Here you can override
# that by specifying the program to use, per URL type. For local files,
# the type is "file" and for web links it is "http*". For example:
# openers_by_scheme = { "http*": "firefox %u" } will make calibre run Firefox
# for https://whatever URLs. %u is replaced by the URL to be opened. The scheme
# takes a glob pattern allowing a single entry to match multiple URL types.
openers_by_scheme = {}

#: Set the first day of the week for calendar popups
# It must be one of the values Default, Sunday, Monday, Tuesday, Wednesday,
# Thursday, Friday, or Saturday, all in English, spelled exactly as shown.
calendar_start_day_of_week = 'Default'

#: East Asian language to use for transliteration
# Setting this tweak will make calibre use the specified language as the "base"
# language when transliterating East Asian languages to English. This might be
# useful if you run calibre in English but want text transliterated to
# Japanese instead of Chinese. The valid values are:
#   'ja' for Japanese
#   'kr' for Korean
#   'vn' for Vietnamese
#   'zh' for Chinese
# Any other value will use the language set in calibre preferences as the base
# language. A base language other than those in the above list causes transliteration
# with a base language of Chinese.
# Example: east_asian_base_language = 'ja'
east_asian_base_language = ''
```

12.4 Sobrescrevendo ícones, templates, etc

Nota

O calibre tem suporte direto para vários temas de ícones, existem diversos temas de ícones disponíveis para o calibre, que você pode usar ao acessar *Preferences → Interface → Look & Feel → Change icon theme*. Dê preferência a usar temas de ícones ao invés de configurar ícones individuais.

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, se encontra 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 você decidiu instalar o Calibre.

Não se deve alterar os arquivos desta pasta de recursos, já que suas alterações serão sobrescritas na próxima atualização do calibre. Ao invés disso, acesse *Preferences → Advanced → Miscellaneous* e clique em *Open calibre configuration folder*. Nesta pasta de configuração, crie uma subpasta chamada resources e coloque nela os arquivos que você deseja sobrepor. Coloque os arquivos nas subpastas apropriadas, por exemplo coloque as imagens na pasta `resources/images`, etc. O calibre irá utilizar automaticamente seu arquivo personalizado ao invés do arquivo padrão da instalação na próxima vez que ele for iniciado.

Por exemplo, se você desejar trocar o ícone para a ação *Remove books* , olhe primeiro na pasta nativa de recursos e veja que o arquivo relevante é `resources/images/remove_books.png`. Presumindo que você tenha um ícone alternativo em formato PNG, chamado `my_remove_books.png` salve-o na pasta de configuração como `resources/images/remove_books.png`. Todos os ícones utilizados pela interface de usuários do calibre estão em `resources/images` e suas subpastas. A colocação de um arquivo substituto aqui terá prioridade até maior do que um tema de ícones personalizado.

12.5 Criando seus próprios temas de ícones para o calibre

Se você criou um belo conjunto de ícones e deseja compartilhá-lo com outros usuários do calibre via suporte integrado para tema de ícones, é fácil empacotar seus ícones em um tema. Para fazer isso, acesse *Preferences → Miscellaneous → Create icon theme*, selecione a pasta onde você colocou seus ícones. Depois preencha os metadados do tema e clique em OK. Isto vai gerar um arquivo ZIP contendo os temas de ícones. Você pode fazer upload do arquivo para o fórum do calibre em Mobileread[113] e então seu tema será disponibilizado via suporte integrado para tema de ícones. Por padrão, o tema de ícones que você criou será instalado como seu tema atual no calibre. Se você estiver testando seu tema, lembre-se de remover as imagens da pasta `resources/images` , para que os ícones do tema sejam usados.

Desde a versão 6 do calibre, é possível ter ícones personalizados para modos de tema claro e escuro. É só criar duas versões do ícone e colocar os sufixos `-for-dark-theme` e `-for-light-theme`. Por exemplo, `modified-for-dark-theme.png` e `modified-for-light-theme.png`. Aí o calibre irá usar automaticamente o ícone apropriado, baseado no tema corrente.

12.6 Personalizando o calibre com plugins

O calibre possui um design muito modular. Quase todas as funcionalidades do calibre vêm na forma de plugins. Os plugins são usados para conversão, para baixar notícias (embora sejam chamadas de receitas), para vários componentes da interface do usuário, para conectar-se a dispositivos diferentes, para processar arquivos ao adicioná-los ao calibre e assim por diante. Você pode obter uma lista completa de todos os plugins integrados no calibre, visitando: guilabel: *Preferências-> Avançado-> Plugins*.

Você 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 ser verificado no tutorial *Crie seus próprios plugins para aumentar as funcionalidades do Calibre* (página 225).

Após você escrever um plugin, você pode fazer upload no fórum de plugins do calibre em Mobileread[114] e ele será disponibilizado via atualizador interno de plugins do calibre.

[113] https://www.mobileread.com/forums/forumdisplay.php?f=166
[114] https://www.mobileread.com/forums/forumdisplay.php?f=237

CAPÍTULO 13

Interface da Linha de Comando

> **Nota**
>
> No macOS, as ferramentas de linha de comando estão dentro do pacote do calibre. Por exemplo, se você instalou o calibre em `/Applications`, as ferramentas da linha de comando estão em `/Applications/calibre.app/Contents/MacOS/`. Então, por exemplo, para executar o `ebook-convert`, você precisaria usar: `/Applications/calibre.app/Contents/MacOS/ebook-convert`.

13.1 Comandos Documentados

13.1.1 `calibre`

```
calibre [opções] [caminho_para_o_livro ou url do calibre ...]
```

Inicia a interface gráfica principal do **calibre** e, opcionalmente, adiciona o livro em caminho_para_o_livro à base de dados. Você também pode especificar URLs do **calibre** para realizar várias ações diferentes, e não apenas adicionar livros. Por exemplo:

calibre://view-book/biblioteca_teste/1842/epub

Abrirá o livro com o id 1842 no formato EPUB na biblioteca "biblioteca_teste" no Visualizador de e-books do **calibre**. Nomes de biblioteca são os nomes das pastas das bibliotecas com os espaços substituídos por underline. Uma descrição completa das várias ações baseadas em URL está no Manual do Usuário.

Sempre que você for passar argumentos para **calibre** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--detach

Desanexar do terminal de controle, se houver algum (somente Linux)

--help, -h

mostrar esta mensagem de ajuda e sair

--ignore-plugins

Ignorar plugins personalizados, útil se você instalou um plugin que está impedindo o calibre de começar

--no-update-check

Não verificar atualizações

--shutdown-running-calibre, -s

Faz com que uma instância ativa do calibre, se houver uma, seja desligada. Note que se houver trabalhos ativos, estes serão abortados silenciosamente, então, use com cuidado.

--start-in-tray

Iniciar minimizado na bandeja do sistema.

--verbose, -v

Ignorado, não use. Presente apenas por motivos de legalidade

--version

mostrar número da versão do programa e sair

--with-library

Usar a biblioteca localizada no caminho específico.

13.1.2 calibre-customize

```
opções calibre-customize
```

personalizar o calibre usando plugins externos

Sempre que você for passar argumentos para **calibre-customize** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--add-plugin, -a

Adiciona um plugin pela especificação do caminho de um arquivo ZIP que o contenha.

--build-plugin, -b

Para desenvolvedores de plugins: o caminho para a pasta em que você está desenvolvendo o plugin. Esse comando vai automaticamente comprimi-lo e atualizá-lo no calibre.

--customize-plugin

Configurar plugin. Especifique o nome do plugin e a string de configuração separados por uma vírgula. A string de configuração é a mesma que você forneceria ao configurar o plugin na janela principal do calibre.

--disable-plugin

Desativar o plugin com nome

--enable-plugin

Ativar o plugin com nome

--help, -h

mostrar esta mensagem de ajuda e sair

--list-plugins, -l

Listar todos os plugins instalados

--remove-plugin, -r

Remover um plugin personalizado por nome. Não tem efeito em plugins nativos

--version

mostrar número da versão do programa e sair

13.1.3 calibre-debug

```
calibre-debug [options]
```

Diversas interfaces de linha de comando úteis para depurar o calibre. Sem opções, esse comando inicia um intérprete Python incorporado. Você também pode executar a interface principal do calibre, o Visualizador de e-books e o editor do calibre no modo de depuração.

Também contém interfaces para várias partes do calibre que não possuem

ferramentas de linha de comando dedicadas, como subconjuntos de fontes, a ferramenta de comparação de e-books, etc.

Você também pode usar o **calibre-debug** para executar scripts isolados. Para isso, use-o assim:

calibre-debug -e myscript.py -- `--option1` `--option2` file1 file2 …

Tudo após o -- é passado ao script. Você também pode usar o **calibre-debug** com um shebang em scripts, assim:

#!/usr/bin/env -S **calibre-debug** -e -- –

Sempre que você for passar argumentos para **calibre-debug** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--add-simple-plugin

Adiciona um plugin (por exemplo, um plugin que consiste de apenas um arquivo .py), pela especificação do caminho de um arquivo py que contenha o código do plugin.

--command, -c

Executar código Python.

--debug-device-driver, -d

Depurar a detecção de dispositivo

--default-programs

(Des)cadastre o calibre da lista dos Programas Padrão do Windows. *--default-programs* (página 305) = `(register|unregister)`

--diff

Executar a ferramenta diff do calibre. Por exemplo: calibre-debug *--diff* (página 305) arquivo1 arquivo2

--edit-book

Carregar a ferramenta `"Editar Livro"` do calibre em modo de depuração.

--exec-file, -e

Executar o código Python no arquivo.

--explode-book, -x

Extrai o livro dentro da pasta especificada. Use: -x arquivo.epub pasta_destino Exporta o livro como uma coleção de arquivos e metadados em HTML, onde você pode editar usando algum editor HTML padrão. Funciona com arquivos EPUB, AZW3, HTMLZ e DOCX.

--export-all-calibre-data

Exportar todos os dados do calibre (livros/configurações/plugins). Normalmente, você será questionado sobre a pasta de exportação e as bibliotecas a serem exportadas. Você pode também pode especificá-los como argumentos da linha de comando para pular as perguntas. Use caminhos absolutos para a pasta de exportação e bibliotecas. A palavra chave especial "all" pode ser utilizada para exportar todas as bibliotecas. Exemplos: calibre-debug *--export-all-calibre-data* (página 306) # para uso interativo calibre-debug *--export-all-calibre-data* (página 306) /caminho/para/pasta/de/exportacao/vazia /caminho/para/pasta/da/biblioteca1 /caminho/para/pasta/da/biblioteca2 calibre-debug *--export-all-calibre-data* (página 306) /pasta/de/exportacao all # exporta todas a bibliotecas conhecidas

--gui, -g

Executa a interface gráfica com a depuração ativada. A saída de depuração é impressa para stdout e stderr.

--gui-debug

Executa a interface gráfica com console de depuração, conectando ao caminho especificado. Para uso interno apenas use a opção -g para executar a interface gráfica no modo de depuração

--help, -h

mostrar esta mensagem de ajuda e sair

--implode-book, -i

Reúne um livro préviamente extraído. USe: -i diretório aquivo.epub Importa o livro dos arquivos dentro do diretório que foram criados pela opção explode-book. Tenha certeza de especifiar os mesmo tipos de arquivo que foram usados quando estraídos.

--import-calibre-data

Importar dados do calibre exportados anteriormente

--inspect-mobi, -m

Inspeciona arquivo(s) MOBI no(s) caminho(s) especificado(s)

--kepubify

Converte o arquivo EPUB especificado para KEPUB sem fazer uma conversão completa. É isto que o driver Kobo faz ao enviar arquivos para o dispositivo.

--paths

Saída dos caminhos necessários para configurar o ambiente do calibre

--run-plugin, -r

Executa um plugin que provê uma interface de linha de comando. Por exemplo: calibre-debug -r "Nome do plugin" -- file1 `--option1` Tudo após o -- será passado ao 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 inicie com um ponto, será considerado como o nome de um módulo. Caso o nome do teste comece com @, será considerado o nome de uma categoria.

--run-without-debug

Não executar com o sinalizador DEBUG definido

--shutdown-running-calibre, -s

Faz com que uma instância ativa do calibre, se houver uma, seja desligada. Note que se houver trabalhos ativos, estes serão abortados silenciosamente, então, use com cuidado.

--subset-font, -f

Subconjunto da fonte especificada. Use -- depois dessa opção para passar opções ao programa de subdivisão de fontes.

--test-build

Testar módulos binários em construção

--un-kepubify

Converte o arquivo KEPUB especificado para EPUB sem fazer uma conversão completa. É isto que o driver Kobo faz ao importar arquivo do dispositivo.

--version

mostrar número da versão do programa e sair

--viewer, -w

Executar o Visualizador de e-books em modo de depuração

13.1.4 calibre-server

```
calibre-server [options] [path to library folder ...]
```

Inicie o servidor de conteúdo calibre. O servidor de conteúdo do calibre apresenta sua biblioteca calibre na internet. Você pode especificar o caminho para as pastas da biblioteca com os argumentos para **calibre-server**. Se você não especificar algum caminho, todas as bibliotecas do programa calibre principal conhecidas serão usadas..

Sempre que você for passar argumentos para **calibre-server** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--access-log

Caminho para o arquivo de registro de acessos. Este registro contém informações sobre os clientes que se conectam e fazem requisições. Por padrão nenhum registro de acesso é feito.

--ajax-timeout

Tempo (em segundos) para esperar uma resposta do servidor quando rodando pesquisas.

--auth-mode

Escolha o tipo de autenticação usada. Define o modo de autenticação HTTP usado pelo servidor. Defina como "basic" se usar o servidor por meio 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 mínimo para o tempo de reinicio do servidor.

--ban-after

Número de falhas de login para banimento. O número de falhas de login depois que um endereço IP é banido

--ban-for

Banir endereços IP que tem falhas de logins repetidas. Proíbe temporariamente o acesso a endereços IP que tenham falhas de login repetidas pelo número de minutos especificado. Útil para evitar tentativas de adivinhar senhas. Se definido como zero, nenhum banimento é feito.

--book-list-mode

Escolha o modo de lista de livros padrão. Define o modo de lista de livros padrão que será utilizado para novos usuários. Usuários individuais podem sobrescrever o padrão em suas próprias configurações. O padrão é utilizar uma grade de capas.

--compress-min-size

Tamanho mínimo para que resposta utilize compressão de dados (em bytes).

--custom-list-template

Caminho para um arquivo JSON contendo um modelo para o modo de lista de livros personalizados. A maneira mais fácil de criar esse arquivo de modelo é ir para Preferências-> Compartilhamento pela rede-> Modelo de lista de livros em calibre, criar o modelo e exportá-lo.

--daemonize

Rodar o processo em segundo plano como um daemon. (Linux somente)

--displayed-fields

Somente exibir campos definidos pelo usuário. Lista separada por vírgulas de campos de metadados definidos pelo usuário que serão exibidos pelo servidor de conteúdo nas exibições /opds e /mobile. Se você definir esta opção, quaisquer campos que não estiverem nessa lista não serão exibidos. Por exemplo: my_rating,my_tags

--enable-allow-socket-preallocation, --disable-allow-socket-preallocation

Pré-alocação do soquete, por exemplo, com ativação do soquete systemd. Por padrão, esta opção está habilitada

--enable-auth, --disable-auth

Autenticação baseada em senha para acessar o servidor. Por padrão o servidor é irrestrito, permitindo qualquer pessoa acessá-lo. Você pode restringir o acesso a usuários pré-definidos com esta opção. Por padrão, esta opção está desabilitada

--enable-fallback-to-detected-interface, --disable-fallback-to-detected-interface

Fallback para a interface auto detectada. Se o servidor não conseguir associar a interface especificada na opção "listen_on", ele tentará detectar uma interface que se conecte com o mundo exterior e se associará a ela. Por padrão, esta opção está habilitada

--enable-local-write, --disable-local-write

Permitir que conexões locais não autenticadas façam alterações. Normalmente, se você não ativar a autenticação, o servidor operará no modo somente leitura, para não permitir que usuários anônimos façam alterações em suas bibliotecas de calibre. Essa opção permite que qualquer pessoa conecte-se no mesmo computador em que o servidor está sendo executado para fazer alterações. Isso é útil se você deseja executar o servidor sem autenticação, mas ainda usar o calibredb para fazer alterações em suas bibliotecas de calibre. Observe que ativar essa opção significa que qualquer programa em execução no computador pode fazer alterações em suas bibliotecas de calibre. Por padrão, esta opção está desabilitada

--enable-log-not-found, --disable-log-not-found

Registrar requisições HTTP 404 (Not Found). Por padrão, o servidor loga todas as requisições HTTP para recursos que não são encontrados. Isto pode gerar muito log indesejado se seu servidor for alvo de bots. Utilize essa opção para desativá-lo. Por padrão, esta opção está habilitada

--enable-use-bonjour, --disable-use-bonjour

Anunciar feeds OPDS via BonJour. Anuncia os feeds OPDS através do serviço BonJour, de maneira que os aplicativos de leitura baseados em OPDS possam detectar e conectar ao servidor automaticamente. Por padrão, esta opção está habilitada

--enable-use-sendfile, --disable-use-sendfile

Ativar ou desativar a transferência de arquivos zero-copy para aumento de performance. Isto usará transferências zero-copy no kernel ao enviar arquivos pela rede, aumentado a performance. Entretanto, pode causar transferência de arquivos corrompidas em alguns sistemas de arquivos falhos. Se você experimentar transferências de arquivos corrompidas, desative-o. Por padrão, esta opção está habilitada

--help, -h

mostrar esta mensagem de ajuda e sair

--ignored-fields

Campos de metadados definidos pelo usuário ignorados. Lista separada por vírgulas de campos de metadados definidos pelo usuário que não serão exibidos pelo servidor de conteúdo nas exibições de /opds e de /mobile. Por exemplo: my_rating,my_tags

--listen-on

A interface que escurará por conexões. O padrão é escutar em todas as interfaces IPv6 e IPv4 disponíveis. Você pode mudar isto para, por exemplo, `"127.0.0.1"` para somente escutar por conexões IPv4 da máquina local, ou para `"0.0.0.0"` para escutar por todas as conexões de entrada IPv4.

--log

Caminho para o arquivo de registros do servidor. Este arquivo contém informações e erros do servidor, não contém informações de acesso. Por padrão é escrito no stdout.

--manage-users

Gerencia a base de dados de usuários com permissão para conectar a este servidor. Você pode usá-lo no modo automatizado ao adicionar um –. Veja calibre-server *--manage-users* (página 309) -- help para detalhes. Veja também a opção *--userdb* (página 310).

--max-header-line-size

Tamanho máximo de cabeçalho HTTP único (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. Processos de trabalho são criados e utilizados conforme necessário para tarefas maiores como a preparação de um livro para exibição, adição de livros, conversão, etc. Por padrão, o número máximo de tais processos é baseado no número de núcleos da CPU. Você pode controlá-lo por meio desta opção.

--max-log-size

Tamanho máximo do arquivo de log (em MB). Tamanho máximo dos arquivos de log gerado pelo servidor. Quando o log se tornar maior que este tamanho, ele é rotacionado automaticamente. Defina para zero para desabilitar rotação de log.

--max-opds-items

Número máximo de livros nos feeds OPDS. Número máximo de livros retornados pelo servidor em uma única aquisição de feed OPDS.

--max-opds-ungrouped-items

Número máximo de itens desagrupados nos feeds OPDS. Agrupa itens em categorias como autor/etiquetas pela primeira letra quando houver mais que esse número de itens. Insira zero para desativar.

--max-request-body-size

Tamanho máximo de arquivos enviados ao servidor (em MB).

--num-per-page

Número de livros a serem exibidos em uma página. Número de livros a serem exibidos em uma página do navegador.

--pidfile

Escreva o processo com o PID para o arquivo especificado

--port

A porta que escutará por conexões.

--search-the-net-urls

Caminho para um arquivo JSON contendo URLs para a função "Pesquisar na internet". O jeito mais fácil de criar este arquivo é ir em Preferências->Compartilhar pela rede->Pesquisar na internet, criar as URLs e exportá-las.

--shutdown-timeout

Tempo total em segundos de espera para um encerramento limpo.

--ssl-certfile

Caminho para o arquivo de certificado SSL.

--ssl-keyfile

Caminho para o arquivo de chave privada SSL.

--timeout

Tempo (em segundos) de espera para que uma conexão ociosa seja fechada.

--trusted-ips

Permitir que conexões não autenticadas de endereços IP específicos façam alterações. Normalmente, se você não ativar a autenticação, o servidor opera no modo somente leitura, para não permitir que usuários anônimos façam alterações em suas bibliotecas do calibre. Esta opção permite que qualquer pessoa de enderenços IP específicos façam alterações. Deve ser um lista de endereços ou redes separada por vírgula. Isto é útil se você deseja executar o servidor sem autenticação mas ainda utilizar o calibredb para fazer alterações em sua suas bibliotecas do calibre. Observe que ativar essa opção significa que qualquer pessoa se conectando dos endereços IP especificados podem fazer alterações em suas bibliotecas do calibre.

--url-prefix

Um prefixo para adicionar à todas as URLs. Útil se você deseja rodar este servidor por meio de um proxy reverso.

--userdb

Caminho para o banco de dados do usuário usado para autenticação. O banco de dados é um arquivo SQLite. Para criá-lo use *--manage-users* (página 309). Você pode ler mais sobre como gerenciar usuários em: https://manual.calibre-ebook.com/pt/server.html#managing-user-accounts-from-the-command-line-only

--version

mostrar número da versão do programa e sair

--worker-count

Número de threads de trabalho utilizadas para processar requisições.

13.1.5 calibre-smtp

```
calibre-smtp [options] [a partir do texto]
```

Enviar mail usando o protocolo SMTP. **calibre-smtp** tem dois modos de operação. No modo de composição que você especifica de para e texto e estes são usados para construir e enviar uma mensagem de e-mail. No modo de filtro, **calibre-smtp** lê uma mensagem completa de e-mail do STDIN e envia-o.

texto é o corpo da mensagem de e-mail. Se o texto não for especificado, uma mensagem completa de e-mail é lida de STDIN. é o endereço de e-mail do remetente e é o endereço de e-mail do destinatário. Quando um e-mail completo é lido de STDIN, de e para apenas são usados na negociação SMTP, os cabeçalhos das mensagens não são modificados.

Sempre que você for passar argumentos para **calibre-smtp** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--fork, -f

Bifurcar e entregar a mensagem em segundo plano. Se você usar esta opção, você também deve usar *--outbox* (página 311) para lidar com falhas de entrega.

--help, -h

mostrar esta mensagem de ajuda e sair

--localhost, -l

Nome do domínio do domínio local. Usado ao conectar ao servidor SMTP.

--outbox, -o

Caminho para a pasta maildir para armazenamento de mensagens de e-mail com falha.

--timeout, -t

Tempo de espera para conexão

--verbose, -v

Seja mais detalhado

--version

mostrar número da versão do programa e sair

COMPOSE MAIL

Opções para compor um e-mail. Ignorado se o texto não for especificado

--attachment, -a

Arquivo para ser anexado ao e-mail

--subject, -s

Assunto do e-mail

SMTP RELAY

Opções para usar um servidor de retransmissão SMTP para enviar mensagens. O calibre tentará enviar o e-mail diretamente a menos –relay é especificado.

--cafile

Caminho para um arquivo de certificados de CA concatenados no formato PEM, usado para verificar o certificado do servidor ao usar o TLS. Por padrão, os certificados de autoridade de certificação do sistema são usados

--dont-verify-server-certificate

Não verifique o certificado do servidor ao se conectar usando o TLS. Este costumava ser o comportamento padrão em versões de calibre antes de 3.27. Se você estiver usando um relay com um certificado autoassinado ou inválido, poderá usar essa opção para restaurar o comportamento anterior à 3.27.

--encryption-method, -e

Método de criptografia para usar ao conectar a retransmissão. As opções são TLS, SSL e NONE. O padrão é TLS. AVISO: Escolher NONE é altamente inseguro

--password, -p

Senha para retransmissão

--port

Porta para conectar no servidor de retransmissão. O padrão é usar 465 se o método de criptografia é SSL e 25 caso contrário.

--relay, -r

Um servidor de retransmissão SMTP para ser usado no envio de e-mails.

--username, -u

Nome do usuário da retransmissão

13.1.6 calibredb

```
calibredb command [options] [arguments]
```

calibredb é a interface de linha de comando para a base de dados do calibre. Ele tem vários sub-comandos, documentados abaixo.

calibredb pode ser usado para manipular uma base de dados do calibre especificada pelo caminho ou um *Servidor de conteúdo* rodando na máquina local ou na internet. Você pode iniciar um *Servidor de conteúdo* usando o programa command:*calibre-server* ou na interface principal do calibre, clicando em *Conectar/compartilhar → Iniciar o Servidor de conteúdo*. Uma vez que o **calibredb** pode fazer alterações nas suas bibliotecas do calibre, você deve configurar a autenticação no servidor primeiro. Há duas maneiras de se fazer isso:

- Se você planeja se conectar somente a um servidor rodando no mesmo computador, você pode simplesmente usar a opção `--enable-local-write` do Servidor de conteúdo, para permitir que qualquer programa, incluindo o calibredb, rodando no computador local, faça alterações aos seus dados do calibre. Ao iniciar, o servidor através da interface principal do calibre, esta opção está em *Preferências → Compartilhar pela rede → Avançado*.
- Se você deseja habilitar o acesso pela internet, então você deve configurar contas de usuários no servidor e utilizar as opções `--username` (página 313) e `--password` (página 313) para o **calibredb** conceder acesso. Você pode configurar a autenticação do usuário no **calibre-server** utilizando a opção `--enable-auth` e usando `--manage-users` para criar contas de usuários. Se você estiver rodando o servidor a partir da interface principal do calibre, use *Preferências → Compartilhar pela rede → Exigir usuário e senha para acessar o servidor de conteúdo*.

Para se conectar a um Servidor de conteúdo rodando, passe a URL do servidor para a opção `--with-library` (página 313). Veja a documentação dessa opção para detalhes e exemplos.

- *OPÇÕES GLOBAIS* (página 313)
- *list* (página 314)
- *add* (página 315)
  - *ADICIONAR A PARTIR DE PASTAS* (página 315)
- *remove* (página 316)
- *add_format* (página 316)
- *remove_format* (página 316)

- *show_metadata* (página 317)
- *set_metadata* (página 317)
- *export* (página 317)
- *catalog* (página 318)
  - *EPUB OPÇÕES* (página 319)
- *saved_searches* (página 320)
- *add_custom_column* (página 320)
- *custom_columns* (página 321)
- *remove_custom_column* (página 321)
- *set_custom* (página 321)
- *restore_database* (página 322)
- *check_library* (página 322)
- *list_categories* (página 322)
- *backup_metadata* (página 323)
- *clone* (página 323)
- *embed_metadata* (página 323)
- *search* (página 324)
- *fts_index* (página 324)
- *fts_search* (página 325)

OPÇÕES GLOBAIS

`--help, -h`

mostrar esta mensagem de ajuda e sair

`--library-path, --with-library`

Caminho para a biblioteca calibre. O padrão é usar o caminho armazenado nas configurações. Você também pode se conectar a um servidor de conteúdo do calibre para executar ações em bibliotecas remotas. Para fazer isso, use uma URL de formulário: http://hostname:port/#library_id por exemplo, http://localhost:8080/#mylibrary. library_id é o ID da biblioteca da biblioteca à qual você deseja se conectar no Servidor de Conteúdo. Você pode usar o valor de library_id - para obter uma lista de IDs de bibliotecas disponíveis no servidor. Para detalhes sobre como configurar o acesso através de um Servidor de Conteúdo, consulte https://manual.calibre-ebook.com/pt/generated/pt/calibredb.html.

`--password`

Senha para se conectar a um Servidor de Conteúdo Calibre. Para ler a senha da entrada padrão, use o valor especial: <stdin>. Para ler a senha de um arquivo, use: <f:/path/to/file> (ex: <f: followed by the full path to the file and a trailing >). Os colchetes acima são necessários, lembre-se de escapar deles ou usar aspas para o seu shell.

`--timeout`

O tempo limite, em segundos, ao conectar-se a uma biblioteca do calibre pela rede. O padrão é dois minutos.

`--username`

Nome de usuário para conectar ao servidor de conteúdo do Calibre

--version

mostrar número da versão do programa e sair

list

```
calibredb list [opções]
```

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--ascending

Ordenar resultados em ordem ascendente

--fields, -f

Campos que serão exibidos ao listar livros no banco de dados. Deve ser uma lista de campos separada 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 Padrão: title,authors. O campo especial "all" pode ser usado para selecionar todos os campos. Em complemento aos campos acima, campos personalizados também estão disponíveis como *field_name. Por exemplo, para um campo personalizado #rating, use o nome: *rating.

--for-machine

Gerar saída no modo JSON, o que é mais adequado para a análise automática. Faz com que a largura de linha e opções de separação sejam ignoradas.

--limit

O número máximo de resultados para exibir. Padrão: tudo

--line-width, -w

A largura máxima de uma única linha na saída. O padrão é detectar o tamanho da tela.

--prefix

O prefixo do caminho de todos arquivos. O padrão é o caminho completo até a pasta da biblioteca.

--search, -s

Filtrar os resultados pela frase de pesquisa. Para o formato da frase de pesquisa, consulte a documentação à respeito de pesquisa no Manual do Usuário. O padrão é não filtrar.

--separator

O Caractere usado para separar os campos. O padrão é um espaço.

--sort-by

O campo pelo qual ordenar os resultados. Você 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 Padrão: id

--template

O modelo a executar se "template" estiver na lista de campos. Note que os modelos são ignorados ao se conectar com o servidor do calibre. Padrão: Nenhum

--template_file, -t

Caminho para um arquivo contendo o modelo a ser executado se "template" estiver na lista de campos. Padrão: Nenhum

--template_heading

Cabeçalho para a coluna do modelo. Padrão: template. Esta opção é ignorada se a opção *--for-machine* (página 314) estiver definida

add

```
calibredb add [opções] arquivos1 arquivo2 arquivo3 ...
```

Adicionar os arquivos especificados como livros para a base de dados. Você pode também especificar pastas, consulte abaixo as opções relacionadas com pastas.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--authors, -a

Configure o(s) autor(es) do(s) livro(s) adicionado(s)

--automerge, -m

Se livros com títulos e autores similares forem encontrados, mescla os formatos (arquivos) de entrada aos registros de livros existentes. O valor `"ignore"` significa que formatos duplicados são descartados. O valor `"overwrite"` significa que os formatos duplicados na biblioteca são sobrescritos com os novos arquivos adicionados. O valor `"new_record"` significa que formatos duplicados serão inseridos em um novo registro de livro.

--cover, -c

Caminho para a capa para usar no livro adicionado

--duplicates, -d

Adicionar livros para a base de dados mesmo se eles já existirem. A comparação é feita baseada nos títulos e autores dos livros. Note que a opção *--automerge* (página 315) tem precedência.

--empty, -e

Adicionar um livro vazio (um livro sem formatos)

--identifier, -I

Definir identificadores para este livro, por exemplo -I asin:XXX -I isbn:YYY

--isbn, -i

Configure o ISBN do(s) livro(s) adicionado(s)

--languages, -l

Uma lista de idiomas separados por vírgulas (melhor usar códigos de idioma ISO639, assim alguns nomes de idiomas possam também ser reconhecidos)

--series, -s

Definir a série do(s) livro(s) adicionado(s)

--series-index, -S

Definir o número de série do(s) livro(s) adicionado(s)

--tags, -T

Configure as etiquetas do(s) livro(s) adicionado(s)

--title, -t

Configure o título do(s) livro(s) adicionado(s)

ADICIONAR A PARTIR DE PASTAS

Opções para controlar a adição de livros a partir de pastas. Por padrão, apenas os arquivos que tem uma extensão de e-book conhecida são adicionados.

--add

Um padrão de nome de arquivo (glob). Arquivos que correspondam a este padrão serão adicionados ao escanear pastas por arquivos, mesmo se forem de um tipo de e-book desconhecido. Pode ser especificado múltiplas vezes para múltiplos padrões.

--ignore

Um padrão (glob) de nome de arquivo. Arquivos correspondentes a este padrão serão ignorados ao escanear pastas por arquivos. Pode ser especificado múltiplas vezes para múltiplos padrões. Por exemplo: *.pdf irá ignorar todos os arquivos PDF

--one-book-per-directory, -1

Assumir que cada pasta possui apenas um e-book e que todos os arquivos nela sejam formatos diferentes do mesmo e-book

--recurse, -r

Processar pastas recursivamente

remove

```
calibredb remove ids
```

Remove os livros identificados por IDs do banco de dados. IDs devem ser uma lista de números id separados por vírgulas (você pode pegar os números de ids utilizando o comando de busca). Por exemplo, 23 ,34,57-85 (quando especificar um alcance, o último número não é incluído).

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--permanent

Não usar a Lixeira

add_format

```
calibredb add_format [options] id ebook_file
```

Adiciona o livro em ebook_file para os formatos disponíveis de livro lógico identificados por id. Você pode pegar um id usando o comando de busca. Se o formato já existe, ele é substituído, a menos que a opção de não substituir seja marcada.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--as-extra-data-file

Adicionar como um arquivo de dados extra do livro, não como um formato de e-book

--dont-replace

Não substituir o formato se ele já existir

remove_format

```
calibredb remove_format [options] id fmt
```

remove o formato fmt de um livro lógico identificado pelo id. Você pode pegar o id utilizando o comando de busca. fmt pode ser uma extensão de arquivo como LRF ou TXT ou EPUB. Se o arquivo lógica não tem fmt disponível, não faz nada.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

show_metadata

```
calibredb show_metada[options] id
```

Apresenta os metadados armazenados na base de dados do calibre para o livro identificado pelo id. Id é um número de id do comando de busca.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--as-opf

Imprimir metadados num formulário OPF (XML)

set_metadata

```
calibredb set_metadata [opções] book_id [/caminho/para/metadata.opf]
```

Define os metadados armazenados na base de dados do calibre para o livro identificado por book_id a partir do arquivo OPF metadata.opf. book_id é um número de identificação do comando de pesquisa. Você pode ter uma ideia do formato OPF utilizando a opção –as-opf 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 arquivo OPF.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--field, -f

O campo a ser definido. O formato é field_name:valor, por exemplo: *--field* (página 317) tags:etiqueta1,etiqueta2. Use *--list-fields* (página 317) para obter uma lista de todos os nomes de campo. Você pode especificar essa opção várias vezes para definir vários campos. Nota: Para idiomas que você deve usar os códigos de linguagem ISO639 (por exemplo, en para Inglês, fr para Francês e assim por diante). Para identificadores, a sintaxe é *--field* (página 317) identifiers:isbn:XXXX,doi:YYYYY. Para campos booleanos (sim/não) usam verdadeiro e falso ou sim e não.

--list-fields, -l

Listar os nomes dos campos de metadados que podem ser usados com a opção *--field* (página 317)

export

```
calibredb export [opções] ids
```

Exportar os livros especificados pelos ids (uma lista separada por vírgula) para o sistema de arquivos. A operação de exportação salva todos os formatos do livro, sua capa e metadados (em um arquivo OPF). Quaisquer arquivos de dados extras associados com o livro também serão salvos. Você pode obter os números de id através do comando de busca.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--all

Exportar todos os livros do banco de dados, ignorando a lista de IDs.

--dont-asciiize

Permitir que o calibre converta todos os caracteres não Ingleses para seus equivalentes nos nomes dos arquivos. Isto é útil ao salvar em sistemas antigos sem suporte a nomes de arquivo Unicode. Especificar esta mudança desligará este comportamento

--dont-save-cover

Normalmente, o calibre irá salvar a capa em um arquivo separado juntamente com os atuais arquivos do e-book Especificar esta mudança desligará este comportamento

--dont-save-extra-files

Salva quaisquer arquivos extras associados ao salvar o livro Especificar esta mudança desligará este comportamento

--dont-update-metadata

Normalmente, o calibre atualizará os metadados nos arquivos salvos que estejam na biblioteca do calibre. Faz com que salvamentos no disco sejam mais lentos. Especificar esta mudança desligará este comportamento

--dont-write-opf

Normalmente, calibre escreverá os metadados dentro de arquivos OPF separados com os arquivos de e-book atuais. Especificar esta mudança desligará este comportamento

--formats

Lista separada por vírgulas de formatos para salvar cada livro. Por padrão todos formatos disponíveis são salvos.

--progress

Relatar progresso

--replace-whitespace

Substituir espaços em branco por `"_"`.

--single-dir

Exportar todos os livros para uma única pasta

--template

Modelo para controlar a estrutura de nome do arquivo e pasta dos arquivos salvos. O padrão é `"{author_sort}/{title}/{title} - {authors}"`, que vai salvar os livros em uma subpasta por autor com nomes de arquivos contendo título e autor. Os controles 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. O padrão é: %b, %Y

--to-dir

Exportar livros para a pasta especificada. O padrão é .

--to-lowercase

Converter caminhos para letras minúsculas

catalog

```
calibredb catalog /path/to/destination.(csv|epub|mobi|xml...) [options]
```

Exporta um catálogo em formato especificado pela extensão path/to/destination. Opções controlam como as entradas são exibidas no catálogo gerado. Note que formatos de catálogos diferentes suportam diferentes conjuntos de opções. Para ver as diferentes opções, especifique o nome do arquivo de saída e então a opção –help.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--ids, -i

Lista separada por vírgulas dos IDs da base de dados do catálogo. Se declarado, *--search* (página 318) será ignorado. Padrão: todos

--search, -s

Filtre os resultados por uma frase de pesquisa. Para o formato da frase de pesquisa, consulte a documentação relacionada a pesquisa no Manual do Usuário. Padrão: sem filtros

--verbose, -v

Mostrar informação detalhada de saída. Útil para depuração

EPUB OPÇÕES

--catalog-title

Título do catálogo gerado usado como título em metadados. Padrão: `'My Books'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--cross-reference-authors

Criar referências cruzadas na seção Autores de livros com vários autores. Padrão: `'False'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--debug-pipeline

Salvar a saída de diferentes estágios do processo de conversão para a pasta especificada. Útil se você não tem certeza em que fase do processo de conversão um erro está ocorrendo. Padrão: `'None'` Aplica-se a: formatos de saída AZW3, EPUB, MOBI

--exclude-genre

Expressão regular descrevendo etiquetas para excluir como gêneros. Padrão: `'[.+]|^+$'` exclui etiquetas entre colchetes, por exemplo, `'[Projeto Gutenberg]'`, e `'+'`, a etiqueta padrão para livros de leitura. 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>','Etiquetas','<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 várias regras são definidas, todas as regras serão aplicadas. Padrão: `"(('Catalogs','Tags','Catalog'),)"` Aplica-se aos formatos de saída AZW3, ePub, MOBI

--generate-authors

Incluir a seção `'Autores'` no catálogo. Padrão: `'False'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--generate-descriptions

Incluir a seção `'Descrições'` no catálogo. Padrão: `'False'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--generate-genres

Incluir a seção `'Gêneros'` no catálogo. Padrão: `'False'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--generate-recently-added

Incluir a seção `'Adicionados Recentemente'` no catálogo. Padrão: `'False'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--generate-series

Incluir a seção `'Série'` no catálogo. Padrão: `'False'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--generate-titles

Incluir a seção `'Títulos'` no catálogo. Padrão: `'False'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--genre-source-field

Campo recurso para a seção `'Gêneros'`. Padrão: `'Etiquetas'` Aplica-se a: formatos de saída AZW3, ePub, MOBI

--header-note-source-field

Campo personalizado que contém o texto da nota para inserir no cabeçalho da Descrição. Padrão: '' 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 mesclar com os comentários [before|after] Espaços reservados de notas com os respectivos comentários [True|False] - Uma régua horizontal é inserida entre notas e comentários padrão: '::' Aplica-se a formatos de saída AZW3, ePub, MOBI

--output-profile

Especifica o perfil de saída. Em alguns casos, um perfil de saída é necessário para otimizar o catálogo para o dispositivo. Por exemplo, 'kindle' ou 'kindle_dx' cria uma tabela estruturada de conteúdo com seções e artigos. Padrão: 'None' Aplica-se a: formatos de saída AZW3, ePub, MOBI

--prefix-rules

Especifica as regras usadas para incluir prefixos que indicam livros de leitura, lista de sonhos e outros prefixos especificados pelo usuário. O modelo para uma regra de prefixo é ('<rule name>','<source field>','<pattern>','<prefix>'). Quando várias regras são definidas, será usada a primeira regra correspondente. Padrão: "(('Read books','tags','+','✓'),('Wishlist item','tags','Wishlist','×'))" Aplica-se a formatos de saída AZW3, ePub, MOBI

--preset

Usar um nome predefinido criado com a GUI do construtor de catálogo. Uma predefinição especifica todos os ajustes para construir um catálogo. Padrão: 'None' Aplica-se a formatos de saída AZW3, ePub, MOBI

--thumb-width

Sugestão de tamanho (polegadas) para capas de livro no catálogo. Intervalo: 1.0 - 2.0 Padrão: '1.0' Aplicado para formatos de saída AZW3, ePub, MOBI

--use-existing-cover

Substituir capas existentes ao gerar o catálogo. Padrão: 'False' Aplica-se a: formatos de saída AZW3, ePub, MOBI

saved_searches

```
calibredb saved_searches [options] (list|add|remove)
```

Gerencie as pesquisas salvas no banco de dados. Se você tentar adicionar uma pesquisa com o nome que já existe, ele será sobrescristo.

Sintaxe para adicionar:

calibredb **saved_searches** add search_name search_expression

Sintaxe para remover:

calibredb **saved_searches** remove search_name

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

add_custom_column

```
calibredb add_custom_column [opções] identificador nome tipo-de-dado
```

Criar uma coluna personalizada. "identificador" é o nome de máquina para coluna. Não pode conter espaços ou dois--pontos. "nome" é o nome da coluna que será exibido. tipo-de-dado pode ser um dos seguintes: bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de 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 321)"{\ "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 legais 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). Você vai ver o JSON da "exibição" para a nova coluna em OPF.

--is-multiple

Este campo armazena dados como etiquetas (ou seja, vários valores separados por vírgulas). Somente se aplica se o tipo de dados for texto.

custom_columns

```
calibredb custom_columns [options]
```

Lista colunas personalizadas disponíveis. Exibe os nomes dos campos e seus IDs.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--details, -d

Mostrar detalhes para cada campo.

remove_custom_column

```
calibredb remove_custom_column[options]label
```

Remover a coluna customizada identificada pela etiqueta. Você poderá ver as colunas costumizadas com o comando custom_columns

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--force, -f

Não pedir confirmação

set_custom

```
calibredb set_custom [options] column id value
```

Marque o valor da coluna customizada para o livro identificado pelo id. Você pode pegar a lista de ids usando o comando de pesquisa. Você pode pegar a lista de colunas costumizadas usando o custom_columns command.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--append, -a

Se o campo possibilitar armazenar diversos valores, não sobrescreva os valores adicione os valores especificados aos já existentes.

restore_database

```
calibredb restore_database [opções]
```

Restaurar esse banco de dados dos metadados armazenados em arquivos OPF em cada pasta de biblioteca do calibre. Isto é útil se o seu arquivo metadata.db foi corrompido.

AVISO: Este comando regenera completamente o seu banco de dados. Você vai perder todas as buscas salvas, as categorias de usuários, quadros de conexões, configurações armazenadas por conversão de livros, e receitas personalizadas. Os metadados restaurados serão apenas tão precisos quanto aos encontrados nos arquivos OPF .

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--really-do-it, -r

Realmente faz a recuperação. O comando não será executado, a menos que esta opção seja especificada.

check_library

```
calibredb check_library [options]
```

Executar algumas verificações no sistema de arquivos 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 você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--csv, -c

Saída em CSV

--ignore_extensions, -e

Lista de extensões separadas por vírgula para ignorar. Padrão: tudo

--ignore_names, -n

Lista de nomes separados por vírgula para ignorar. Padrão: tudo

--report, -r

Lista de relatórios separados por vírgula. Padrão: tudo

--vacuum-fts-db

Limpar a base de dados da pesquisa no texto completo. Isto pode ser muito lento e utilizar muita memória, dependendo do tamanho da base de dados.

list_categories

```
calibredb list_categories [opções]
```

Produz um relatório das informações da categoria no banco de dados. A informação é equivalente à que é mostrada no Navegador de etiquetas.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--categories, -r

Lista de categorias separadas por vírgula para pesquisar nomes. Padrão: tudo

--csv, -c

Saída em CSV

--dialect

Tipo de arquivo CSV que será produzido. Escolhas: excel, excel-tab, unix

--item_count, -i

Processa apenas o número de itens em uma categoria, em vez das contagens por item dentro da categoria

--width, -w

A largura máxima de uma única linha na saída. O padrão é detectar o tamanho da tela.

backup_metadata

```
calibredb backup_metadata [options]
```

Backup dos metadados armazenados no banco de dados em arquivos OPF individuais em cada pasta de livros. Isso normalmente acontece automaticamente, mas você pode executar este comando para forçar uma nova geração dos arquivos OPF, com a opção –all.

Note que normalmente não há necessidade de fazer isso, pois os arquivos OPF são copiados automaticamente, toda vez que os metadados são alterados.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--all

Normalmente, este comando apenas opera em livros que tenham arquivos OPF desatualizados. Esta opção irá criar operações em todos os livros.

clone

```
calibredb clone path/to/new/library
```

Criar um **clone** da biblioteca atual. Isto cria uma nova biblioteca vazia que contém as mesmas colunas personalizadas, Bibliotecas Virtuais e outros ajustes como a biblioteca atual.

A biblioteca clonada não contêm livros. Se você quiser criar uma cópia completa, incluindo todos os livros, use em seguida suas ferramentas de sistema de arquivos para copiar a pasta da biblioteca.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

embed_metadata

```
calibredb embed_metadata [options] book_id
```

Atualizar os metadados nos arquivos de livros atuais armazenados na biblioteca do calibre dos metadados no banco de dados do calibre. Normalmente, os metadados são atualizadas apenas

quando exportar arquivos do calibre, este comando é útil se você deseja que os arquivos

sejam atualizados no local. Observe que diferentes formatos de arquivos suportam diferentes

quantidades de metadados. Você pode usar o valor especial 'all' para book_id para atualizar os metadados em todos os livros. Você também pode especificar várias ids de livro separado

por espaços e intervalos de ID separados por hífens. Por exemplo: calibredb **embed_metadata** 1 2 10-15 23

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--only-formats, -f

Apenas atualizar os metadados em arquivos do formato especificado. Especifique-o várias vezes para vários formatos. Por padrão, todos os formatos são atualizados.

search

```
calibredb search [options] search expression
```

Procura a biblioteca para o termo de pesquisa especificado, retornando uma lista separada por vírgula de ids de livros resultantes da expressão de busca. O formato de saída é útil para alimentar outros comandos que aceitam uma lista de ids como entrada.

O comando de expressão poder ser qualquer coisa da poderosa consulta de busca do calibre por exemplo: calibredb **search** author:asimov 'title:"i robot"'

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--limit, -l

O número máximo de resultados para retornar. O padrão é todos os resultados.

fts_index

```
calibredb fts_index [opções] enable/disable/status/reindex
```

Controla o processo de indexação da pesquisa no texto completo.

enable

Habilita a indexação nesta biblioteca

disable

Desabilita a indexação nesta biblioteca

status

Mostra o status atual da indexação

reindex

Pode ser usado para reindexar livros específicos ou a biblioteca inteira. Para reindexar livros específicos, especifique os ids como um argumento adicional após o comando reindex. Se nenhum id for especificado, a biblioteca inteira é reindexada.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

--indexing-speed

A velocidade da indexação. Use `'fast'` para indexação rápida, utilizando todos os recursos de seu computador, ou `'slow'` para indexação menos intensa, consumindo menos recursos. Note que a velocidade é redefinida para lenta após cada invocação.

--wait-for-completion

Aguarde até que todos os livros sejam indexados, exibindo periodicamente o progresso da indexação

fts_search

```
calibredb fts_search [opções] expressão de busca
```

Pesquisa no texto completo de toda a biblioteca ou em um subgrupo.

Sempre que você for passar argumentos para calibredb que possuem espaços em branco, coloque o argumento dentro de 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 ao redor de cada correspondência. Note que isso torna a pesquisa muito mais lenta.

--indexing-threshold

O quanto da biblioteca deve estar indexado antes de permitir a pesquisa, como porcentagem. O padrão 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 para a saída de resultado de pesquisa. Pode ser `"text"` para texto simples ou `"json"` para saída JSON.

--restrict-to

Restringir os livros pesquisados, seja usando uma expressão de busca ou ids. Por exemplo: ids:1,2,3 para restringir por ids ou search:tag:foo para restringir a livros com a etiqueta `'foo'`.

13.1.7 ebook-convert

```
ebook-convert arquivo_entrada arquivo_saída [opções]
```

Converta um e-book de um formato para outro.

arquivo_entrada é a entrada e arquivo_saída é a saída. Ambos devem ser especificados como os dois primeiros argumentos para o comando.

O formato de e-book de saída é adivinhado a partir da extensão de arquivo do arquivo_saída. O arquivo_saída também pode ser do formato especial .EXT, em que EXT é a extensão do arquivo de saída. Nesse caso, o nome do arquivo de saída é derivado do nome do arquivo de entrada. Observe que os nomes dos arquivos não devem começar com um hífen. Finalmente, se o arquivo_saida não tiver nenhuma extensão, ele será tratado como uma pasta e um "e-book aberto" (OEB), consistindo de arquivos HTML, será gravado nessa pasta. Esses arquivos são os arquivos que normalmente teriam sido passados para o plugin de saída.

Depois de especificar o arquivo de entrada e saída, você pode personalizar a conversão especificando várias opções. As opções disponíveis dependem dos tipos de arquivo de entrada e saída. Para obter ajuda sobre eles, especifique o arquivo de entrada e saída e, em seguida, use a opção -h.

Para documentação completa do sistema de conversão, consulte *Conversão de E-book* (página 61)

Sempre que você for passar argumentos para **ebook-convert** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

As opções e valores padrão para as opções alteram dependendo dos formatos de entrada e saída, por isso você deve sempre verificar com:

```
ebook-convert myfile.input_format myfile.output_format -h
```

Abaixo estão as opções que são comuns para todas as conversões, seguido pelas opções específicas para cada formato de entrada e saída.

- *OPÇÕES DE ENTRADA* (página 326)
- *OPÇÕES DE SAÍDA* (página 327)
- *APARÊNCIA* (página 328)
- *PROCESSAMENTO HEURÍSTICO* (página 330)
- *LOCALIZAR E SUBSTITUIR* (página 331)
- *DETECÇÃO DE ESTRUTURA* (página 331)
- *ÍNDICE* (página 332)
- *METADADOS* (página 333)
- *DEPURAÇÃO* (página 334)

--help, -h

mostrar esta mensagem de ajuda e sair

--input-profile

Especificar o perfil de origem. O perfil de origem dá informações ao sistema de conversão sobre como interpretar várias informações no documento de origem. Por exemplo, comprimento dependente da resolução (ou seja, comprimento em pixels). As escolhas são: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes

Listar nomes de receita incorporados. Você pode criar um e-book a partir de uma receita embutida como esta: ebook-convert "Nome da receita.recipe" output.epub

--output-profile

Especifica o perfil de saída. O perfil de saída diz ao sistema de conversão como otimizar o documento criado para o dispositivo especificado. 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_large, generic_eink_hd, hanlinv3, hanlinv5, illiad, ipad, ipad3, irexdr1000, irexdr800, jetbook5, kindle, kindle_dx, kindle_fire, kindle_oasis, kindle_pw, kindle_pw3, kindle_scribe, kindle_voyage, kobo, msreader, mobipocket, nook, nook_color, nook_hd_plus, pocketbook_inkpad3, pocketbook_lux, pocketbook_hd, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet

--version

mostrar número da versão do programa e sair

OPÇÕES DE ENTRADA

Opções para controlar o processamento do arquivo de entrada mobi

--input-encoding

Especificar a codificação de caracteres do documento de entrada. Se escolhida, esta opção irá substituir qualquer outra codificação declarada pelo próprio documento. Particularmente útil para documentos que não declaram uma codificação ou que possuem declarações de codificação incorretas.

OPÇÕES DE SAÍDA

Opções para controlar o processamento do arquivo de saída epub

--dont-split-on-page-breaks

Desativar a divisão nas quebras de página. Normalmente, os arquivos de entrada são divididos automaticamente a cada quebra de página em dois arquivos. Isso fornece um e-book de saída que pode ser analisado mais rapidamente e com menos recursos. No entanto, a divisão é lenta e, se o arquivo de origem contiver um número muito grande de quebras de página, você deverá desativar a divisão nas quebras de página.

--epub-flatten

Esta opção só é necessária se você pretende utilizar o formato EPUB no FBReaderJ. Ele vai achatar o sistema de arquivos dentro do EPUB, colocando todos os arquivos no nível superior.

--epub-inline-toc

Insira um índice em linha que irão aparecer como parte do conteúdo do livro principal.

--epub-max-image-size

O tamanho máximo de imagem (largura x altura). Um valor de none significa utilizar o tamanho da tela do perfil de saída. Um valor de profile significa que nenhum tamanho máximo foi especificado. Por exemplo, um valor de 100x200 fará com que todas as imagens sejam redimensionadas de tal maneira que sua largura tenha não mais que 100 pixels e sua altura tenha não mais que 200 pixels. Note que isto somente afeta o tamanho real dos arquivos de imagem. Qualquer imagem pode ser exibida com um tamanho diferente dependendo do estilo aplicado a ela no documento.

--epub-toc-at-end

Colocar o índice em linha inserida no final do livro, em vez do começo.

--epub-version

A versão do arquivo EPUB para ser gerado. O EPUB 2 é o mais compatível, use apenas o EPUB 3 se você sabe que realmente precisa dele.

--extract-to

Extraia os conteúdo do livro gerado para a pasta especificada. O conteúdo da pasta é apagado primeiro, então tenha cuidado.

--flow-size

Divida todos os arquivos HTML maiores que este tamanho (em KB). Isto é necessário já que a maioria dos leitores de EPUB não dão conta de arquivos grandes. O padrão de 260KB é o tamanho exigido para o Adobe Digital Editions. Configure para 0 para desabilitar a divisão baseada no tamanho.

--no-default-epub-cover

Normalmente, se o arquivo de entrada não tem capa e você não especificou uma, uma capa padrão é gerada com o título, autores, etc. Esta opção desabilita a geração desta capa.

--no-svg-cover

Não use SVG para a capa do livro. Use esta opção se seu EPUB irá ser usado em um dispositivo que não suporta SVG, como o iPhone ou o JetBook Lite. Sem esta opção, ambos dispositivos irão mostrar a capa como uma página em branco.

--preserve-cover-aspect-ratio

Quando estiver usando uma capa SVG, esta opção irá fazer a capa se ajustar ao tamanho da área da tela, mas preservando seu aspecto sem deformação (largura e comprimento). Isto quer dizer que pode haver bordas brancas nos lados ou na parte de cima e em baixo da imagem, mas ela nunca ficará deformada. Sem esta opção a imagem pode ser visivelmente deformada, mas não haverão bordas brancas.

--pretty-print

Se especificado, o plugin de saída tentará produzir uma saída que seja tão legível por pessoas quanto possível. Pode não ter efeito para alguns plugins de saída.

--toc-title

Título para qualquer índice gerado em linha.

APARÊNCIA

Opções para controlar a aparência do arquivo de saída

--asciiize

Transformar caracteres Unicode em uma representação ASCII. Use com cuidado, porque isso irá substituir caracteres Unicode por ASCII. Por exemplo, ele irá substituir "Pelé" com "Pele". Além disso, observe que nos casos em que existem várias representações de um caractere (caracteres compartilhados por chineses e japoneses, por exemplo), será usada a representação com base no idioma atual da interface do calibre.

--base-font-size

O tamanho da fonte base em pts. Todos os tamanhos de fonte no livro produzido serão redimensionados com base nesse tamanho. Ao escolher um tamanho maior, você pode aumentar as fontes na saída e vice-versa. Por padrão, quando o valor é zero, o tamanho da fonte base é escolhido com base no perfil de saída escolhido.

--change-justification

Mudar a justificação do texto. Um valor de "Esquerda" converte todo o texto justificado na origem para texto alinhado a esqueda (por ex. sem justificação). Um valor de "Justificado" converte todo o texto não justificado para justificado. Um valor de "Original" (o padrão) não muda a justificação do arquivo de origem. Note que somente alguns formatos de saída suportam justificação.

--disable-font-rescaling

Desativar redimensionamento do tamanho das letras.

--embed-all-fonts

Incorporar todas as fontes que referenciam o documento de entrada pode não estar atualmente incorporado. Isso irá procurar no seu sistema por fontes, e se encontradas, elas serão incorporadas. Incorporar somente irá funcionar no formato que você está convertendo que suporte fontes incorporadas, como EPUB, AZW3, DOCX ou PDF. Por favor, tenha a certeza de que você possui a licensa apropriada para incorporar as fontes utilizadas neste documento.

--embed-font-family

Coloque a fonte especificada no livro. Isto especifica a fonte "base" usada no livro. Se o documento de entrada especifica suas próprias fontes, elas podem sobrepor esta fonte base. Você pode usar a opção "Filtrar informação de estilo" para remover fontes do documento de entrada. Note que isso funciona somente com alguns formatos de saída, principalmente EPUB, AZW3 e AZW3.

--expand-css

Por padrão, o calibre usará o formato abreviado para várias propriedades CSS, como margem, preenchimento, borda etc. Essa opção fará com que ele use o formulário expandido completo. Observe que o CSS é sempre expandido ao gerar arquivos EPUB com o perfil de saída configurado para um dos perfis do Nook, pois o Nook não pode manipular o CSS abreviado.

--extra-css

O caminho para uma folha de estilo CSS ou CSS puro. Este CSS será acrescentado às regras de estilo do arquivo de origem, de modo a substituir aquelas regras.

--filter-css

Uma lista de propriedades CSS separadas por vírgulas que será removida de todas as regras de estilo do CSS. Isto é útil se a presença de informações dos estilo previna-o de ser sobrescrito no seu dispositivo. Por exemplo: font-family,color,margin-left,margin-right

--font-size-mapping

Mapeamento de nomes de fonte em CSS para tamanhos de fonte em pontos. Um exemplo de configuração é 12,12,14,16,18,20,22,24. Estes são os mapeamentos para os tamanhos de muito pequeno (xx-small) a muito grande (xx-large), com o último tamanho sendo para fontes enormes (huge). O algoritmo de redimensionamento de fonte usa estes tamanhos para redimensionar as fontes de forma inteligente. O padrão é usar um mapeamento com base no perfil de saída que você escolheu.

--insert-blank-line

Insere uma linha em branco entre parágrafos. Não irá funcionar se o arquivo de origem não usa parágrafos (tags <p> ou).

--insert-blank-line-size

Definir a altura das linhas em branco inseridas (em `'em'`). a altura das linhas entre paragrafos será duas vezes o tamanho definido aqui.

--keep-ligatures

Preserva as ligaduras presentes no documento de entrada. Uma ligadura é um formato especial para mostrar um par de caracteres tal como ff, fi, fl, etc. A maioria dos leitores não têm suporte para ligaduras nas sua fonte padrão e não conseguem exibi-las corretamente. Por padrão, o calibre irá transformar uma ligadura no seu par correspondente de caracteres normais. Note que as ligaduras aqui significam apenas ligaduras unicode e não ligaduras criadas via CSS ou por estilos de fontes. Esta opção irá preservar as ligaduras.

--line-height

A altura da linha em pts. Controla o espaçamento entre linhas consecutivas de texto. Só se aplica a elementos que não definem a sua altura de linha própria. Na maioria dos casos, a opção de altura mínima de linha é mais útil. Por padrão nenhuma manipulação de altura de linha é executada.

--linearize-tables

Alguns documentos mal modelados usam tabelas para controlar a disposição do texto na página. Quando convertidos estes documentos geralmente ficam com texto fora da página e outros problemas. Esta opção irá extrair o conteúdo das tabelas e mostra ele forma linear.

--margin-bottom

Definir a margem inferior em pts. O padrão é %d. Definir isso para menos de zero fará com que nenhuma margem seja definida (a configuração de margem no documento original será preservada). Nota: Formatos orientados a páginas, como PDF e DOCX, têm suas próprias configurações de margem que têm precedência.

--margin-left

Definir a margem esquerda em pts. O padrão é %d. Definir isso para menos de zero fará com que nenhuma margem seja definida (a configuração de margem no documento original será preservada). Nota: Formatos orientados a páginas, como PDF e DOCX, têm suas próprias configurações de margem que têm precedência.

--margin-right

Definir a margem direita em pts. O padrão é %d. Definir isso para menos de zero fará com que nenhuma margem seja definida (a configuração de margem no documento original será preservada). Nota: Formatos orientados a páginas, como PDF e DOCX, têm suas próprias configurações de margem que têm precedência.

--margin-top

Definir a margem superior em pts. O padrão é %d. Definir isso para menos de zero fará com que nenhuma margem seja definida (a configuração de margem no documento original será preservada). Nota: Formatos orientados a páginas, como PDF e DOCX, têm suas próprias configurações de margem que têm precedência.

--minimum-line-height

A altura mínima de linha, como uma percentagem do tamanho do elemento de fonte calculado. O calibre irá garantir que cada elemento tenha uma altura de linha de pelo menos esta configuração, independentemente do que o documento de entrada especifica. Configure para zero para desativar. O padrão é 120%. Use esta configuração de

preferência para a especificação direta de altura de linha, a menos que você saiba o que está fazendo. Por exemplo, você pode conseguir "espaço duplo" texto definindo isso para 240.

--remove-paragraph-spacing

Remove espaço entre parágrafos. Também configura um espaçamento de 1,5 linhas nos parágrafos. A remoção de espaços não irão funcionar se o arquivo de origem não usa parágrafos (tags <p> ou).

--remove-paragraph-spacing-indent-size

Quando calibre remove linhas em branco entre parágrafos, ele insere automaticamente um recuo de linha, para garantir que parágrafos sejam facilmente distinguíveis. Esta opção controla o tamanho da quebra de linha (em "em"). Se você inserir um valor negativo, o recuo especificado no documento será utilizado, isso significa que calibre não modificará o recuo.

--smarten-punctuation

Converta aspas, travessões e reticências em seus equivalentes tipograficamente corretos. Para detalhes, veja https://daringfireball.net/projects/smartypants.

--subset-embedded-fonts

Subconjunto de todas as fontes incorporadas. Cada fonte incorporada é reduzida para conter apenas os glifos usados neste documento. Isto diminui o tamanho dos arquivos de fonte. Útil se você estiver incorporando fontes particularmente grandes com muitos glifos não utilizados.

--transform-css-rules

Caminho para um arquivo contendo regras para transformar os estilos CSS neste livro. A maneira mais fácil de criar esse arquivo é usar o assistente para criar regras na GUI do calibre. Acesse-o na seção "Aparência-> Transformar estilos" da caixa de diálogo de conversão. Depois de criar as regras, você pode usar o botão "Exportar" para salvá-las em um arquivo.

--transform-html-rules

Caminho para um arquivo contendo regras para transformar o HTML deste livro. A maneira mais fácil de criar um arquivo como este é usar o Assistente para criar regras na interface gráfica do calibre. Acesse-o na seção "Aparência->Transformar HTML" no diálogo de conversão. Uma vez criadas as regras, você pode usar o botão "Exportar" para salvá-las em um arquivo.

--unsmarten-punctuation

Converter citações extravagantes, traços e reticências aos seus equivalentes simples.

PROCESSAMENTO HEURÍSTICO

Modifique o documento de texto e sua estrutura usando padrões comuns. Inicialmente desativado. Use –enable-heuristics para ativar. Ações individuais podem ser desabilitaras com as –disable-* opções.

--disable-dehyphenate

Analiza palavras que contenham hifén no documento. O documento será utilizado como um dicionário para determinar se hiféns devem ser mantidos ou removidos.

--disable-delete-blank-paragraphs

Remova parágrafos vazios do documento quando eles existem entre outros parágrafos.

--disable-fix-indents

Transforma a identação criada a partir de múltiplas tabulações em identações CSS

--disable-format-scene-breaks

Marcadores de quebra de cena alinhados à esquerda são centralizados. Substituir quebras de cenas suaves que usam várias linhas em branco por linhas horizontais.

--disable-italicize-common-cases

Procure por palavras e padrões comuns que deveriam ser itálicos e os coloque em itálico.

--disable-markup-chapter-headings

Detectar títulos e subtítulos de capítulos sem formatação. Altera-os para tags h2 e h3. Esta opção não irá criar um índice, mas pode ser utilizada junto com a estrutura de detecção para criar um.

--disable-renumber-headings

Procura ocorrências sequenciais de tags <h1> ou <h2>. As tags serão renumeradas para evitar quebra no meio do título dos capítulos.

--disable-unwrap-lines

Eliminar quebra de linhas usando pontuação e outras formas de formatação.

--enable-heuristics

Permitir processamento heurístico. Esta opção deverá estar ativada para que qualquer processamento heurístico seja feito.

--html-unwrap-factor

Escala usada para determinar o tamanho que a linha deve ter quando as quebras forem eliminadas. Valores válidos são decimais entre 0 e 1. O padrão é 0.4, um pouco abaixo da média do tamanho da linha. Se apenas algumas linhas do documento necessitam ter suas linhas eliminadas, este valor deve ser reduzido.

--replace-scene-breaks

Substituir quebras de cena com o texto especificado. Por padrão, o texto do documento de entrada será usado.

LOCALIZAR E SUBSTITUIR

Modificar o texto do documento e estrutura usando os padrões definidos pelo usuário.

--search-replace

Caminho para um arquivo que contém pesquisa e substitui expressões regulares. O arquivo deve conter linhas alternadas de expressão regular seguidas por padrão de substituição (que pode ser uma linha vazia). A expressão regular deve estar na sintaxe de regex do Python e o arquivo deve ser codificado em UTF-8.

--sr1-replace

Substituição para substituir o texto encontrado com a busca sr1.

--sr1-search

Padrão de busca (expressão regular) para ser substituído pelo substituto P&S-1.

--sr2-replace

Substituição para substituir o texto encontrado com a busca sr2.

--sr2-search

Padrão de busca (expressão regular) para ser substituído pelo substituto P&S-2.

--sr3-replace

Substituição para substituir o texto encontrado com a busca sr3.

--sr3-search

Padrão de busca (expressão regular) para ser substituído por sr2-replace.

DETECÇÃO DE ESTRUTURA

Auto-detecção do controle da estrutura de documento.

--add-alt-text-to-img

Quando uma etiqueta <img> não possui um atributo Alt, verifica os metadados de texto alternativo da imagem associada e usa isso para preencher o atributo Alt. O atributo Alt melhora a acessibilidade, fornecendo descrições de texto para usuários que não conseguem ver ou interpretar completamente o conteúdo visual.

--chapter

Uma expressão XPath para detectar títulos de capítulos. O padrão é considerar tags <h1> ou <h2> que contenham as palavras "capítulo", "livro", "sessão", "prólogo", "epílogo" ou "parte" como títulos de capítulo, bem como quaisquer tags que tenham class = "chapter ". A expressão usada deve avaliar uma lista de elementos. Para desativar a detecção de capítulos, use a expressão "/". Consulte o tutorial do XPath no manual do usuário do calibre para obter mais ajuda sobre como usar esse recurso.

--chapter-mark

Especifique como marcar os capítulos detectados. O valor "pagebreak" irá inserir uma quebra de página antes dos capítulos. "Rule" irá inserir uma linha antes de cada capítulo. "none" irá desligar a marcação de capítulos, enquanto "both" irá inserir uma quebra de página e uma linha para marcar os capítulos.

--disable-remove-fake-margins

Alguns documentos especificam os tamanhos de margem direita e esquerda individualmente para cada parágrafo. O calibre tentará detectar e remover estas margens. Alguma vezes, isso pode causar a remoção indevida de margens. Neste caso, você pode desativar a remoção.

--insert-metadata

Insira os metadados do livro no início do livro. Isso é útil se o seu e-book reader não suporta a exibição / pesquisa de metadados diretamente.

--page-breaks-before

Uma expressão XPath. Quebras de página são inseridas antes dos elementos especificados. Para desativar, utilize a expressão: /

--prefer-metadata-cover

Usar a capa detectada no arquivo fonte em relação à capa especificada.

--remove-first-image

Remova a primeira imagem do e-book de entrada. Útil se o documento de entrada tiver uma imagem de capa não identificada como capa. Nesse caso, se você definir uma capa em calibre, o documento de saída terminará com duas imagens de capa, se você não especificar essa opção.

--start-reading-at

Uma expressão XPath para detectar o local no documento no qual começar a ler. Alguns programas de leitura de livros eletrônicos (mais proeminentemente o Kindle) usam esse local como a posição na qual abrir o livro. Consulte o tutorial do XPath no Manual do Usuário do calibre para obter ajuda adicional ao usar essa função.

ÍNDICE

Controlar a geração automática do Índice. Por padrão, se o arquivo fonte tem um índice, ele será usado no lugar do que seria gerado automaticamente.

--duplicate-links-in-toc

Ao criar de um índice a partir de links contidos no documento de entrada, é permitido entradas duplicadas, isto é: permitir mais de uma entrada com o mesmo texto, desde que eles apontem para locais diferentes.

--level1-toc

Expressão XPath que especifica todas as tags que deverão ser adicionadas no primeiro nível do índice. Se estiver especificada, terá precedência em relação a outras formas de detecção automática. Consulte o tutorial do XPath no Manual do Usuário do calibre para exemplos.

--level2-toc

Expressão XPath que especifica todas as tags que deverão ser adicionadas no segundo nível do índice. Cada entrada será adicionada abaixo da entrada anterior no nível um. Consulte o tutorial do XPath no Manual do Usuário do calibre para exemplos.

--level3-toc

Expressão XPath que especifica todas as tags que deverão ser adicionadas no terceiro nível do índice. Cada entrada será adicionada abaixo da entrada anterior no nível dois. Consulte o tutorial do XPath no Manual do Usuário do calibre para exemplos.

--max-toc-links

Número máximo de links para inserir no índice. Use 0 para desativar. O padrão é: 50. Os links serão adicionados ao índice apenas se o número encontrado for menor que o limite máximo de capítulos.

--no-chapters-in-toc

Não adicionar para o índice os capítulos detectados automaticamente.

--toc-filter

Remover os itens do índice cujos títulos correspondem à expressão regular especificada. Os itens correspondentes e subitens serão removidos.

--toc-threshold

Se um número de capítulos menor do que este for detectado, links a eles serão adicionados para o índice. Padrão: 6.

--use-auto-toc

Normalmente, se o arquivo fonte já tem um índice, ele é usado com preferência em relação ao que é gerado automaticamente. Com esta opção, o que é gerado automaticamente será sempre usado.

METADADOS

Opções para escolher os metadados no arquivo de saída

--author-sort

O texto a ser usado quando ordenando por autor.

--authors

Configura os autores. Múltiplos autores devem ser separados por `"&"`.

--book-producer

Defina o produtor do livro.

--comments

Defina a descrição do e-book.

--cover

Configura a capa para o arquivo específico ou URL

--isbn

Defina o ISBN do livro.

--language

Defina o idioma.

--pubdate

Definir a data de publicação (supostamente no fuso horário local, a menos que o fuso horário seja especificado explicitamente)

--publisher

Defina o editor do e-book.

--rating

Defina a avaliação. Deve ser um número entre 1 e 5.

--read-metadata-from-opf, --from-opf, -m

Ler metadados de um arquivo OPF específico. Metadados lidos deste arquivo irão substituir quaisquer metadados no arquivo de origem.

--series

Defina a série a que este e-book pertence.

--series-index

Define o índice do livro nesta série.

--tags

Defina etiquetas para o livro. Deve ser uma lista separada por vírgulas.

--timestamp

Definir a marca temporal do livro (não é mais usada em lugar algum)

--title

Defina o título.

--title-sort

A versão do título a ser usada na ordenação.

DEPURAÇÃO

Opções para auxiliar no debugging da conversão

--debug-pipeline, -d

Salvar o resultado de diferentes etapas do processo de conversão para a pasta especificada. Útil se você não tem certeza em que etapa do processo de conversão está ocorrendo um erro.

--verbose, -v

Nível de verbosidade. Especifique várias vezes para maior verbosidade. Especificando-o duas vezes resultará em verbosidade completa, enquanto a verbosidade média e zero vezes menos verbosidade.

13.1.8 ebook-edit

```
ebook-edit [opts] [path_to_ebook] [name_of_file_inside_book ...]
```

Inicia a ferramenta de edição de livros do calibre. Você pode, opcionalmente, especificar também os nomes dos arquivos dentro do livro que será aberto para editá-los automaticamente.

Sempre que você for passar argumentos para **ebook-edit** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--detach

Desanexar do terminal de controle, se houver algum (somente Linux)

--help, -h

mostrar esta mensagem de ajuda e sair

--select-text

O texto a ser selecionado no livro quando for aberto para edição

--version

mostrar número da versão do programa e sair

13.1.9 ebook-meta

```
ebook-meta ebook_file [options]
```

Leitura/Gravação de metadados de/para arquivos de e-book.

Formatos suportados para leitura de metadados: azw, azw1, azw3, azw4, cb7, cbc, cbr, cbz, chm, docx, epub, fb2, fbz, html, htmlz, imp, kepub, lit, lrf, lrx, mobi, odt, oebzip, opf, pdb, pdf, pml, pmlz, pobi, prc, rar, rb, rtf, snb, tpz, txt, txtz, updb, zip

Formatos suportados para gravação de metadados: azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, kepub, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

Diferentes formatos de arquivos suportam diferentes tipos de metadados. Ao tentar inserir metadados não suportados pelo formato do arquivo, os metadados serão ignorados silenciosamente.

Sempre que você for passar argumentos para **ebook-meta** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--author-sort

Sequência de caracteres para ser usada quando for usada a ordenação por autor. Se não especificada, e o(s) autor(es) for especificado, ela será gerada automaticamente a partir do(s) autor(es).

--authors, -a

Definir autores. Autores múltiplos devem ser separados pelo caractere `"&"`. Nomes de autores devem estar na sequência PrimeiroNome ÚltimoNome.

--book-producer, -k

Defina o produtor do livro.

--category

Escolher a categoria do livro.

--comments, -c

Defina a descrição do e-book.

--cover

Defina a capa do arquivo especificado.

--date, -d

Escolher a data de publicação

--disallow-rendered-cover

Para formatos como EPUB, que usam uma `"capa padrão"` criada a partir da primeira página renderizada, desabilita essas capas padrão

--from-opf

Leia os metadados do arquivo OPF especificado e use-o para definir os metadados no e-book. Os metadados especificados na linha de comando substituirão os metadados lidos do arquivo OPF

--get-cover

Obter a capa do e-book e salve-o como o arquivo especificado.

--help, -h

mostrar esta mensagem de ajuda e sair

--identifier

Defina identificadores para o livro. Pode ser especificado múltiplas vezes. Por exemplo: *--identifier* (página 336) uri:https://acme.com *--identifier* (página 336) isbn:12345. Para remover um identificador, não especifique nenhum valor, *--identifier* (página 336) isbn:. Note que para arquivos EPUB, um identificador marcado como "package identifier" não pode ser removido.

--index, -i

Define o índice do livro nesta série.

--isbn

Defina o ISBN do livro.

--language, -l

Defina o idioma.

--lrf-bookid

Altere o BookID nos arquivos LRF

--publisher, -p

Defina o editor do e-book.

--rating, -r

Defina a avaliação. Deve ser um número entre 1 e 5.

--series, -s

Defina a série a que este e-book pertence.

--tags

Defina etiquetas para o livro. Deve ser uma lista separada por vírgulas.

--title, -t

Defina o título.

--title-sort

A versão do título para ser usado na ordenação. Se não especificado, e o título for especificado, ele será gerado automaticamente a partir do título.

--to-opf

Especificar o nome de um arquivo OPF. O metadado irá ser escrito no arquivo OPF.

--version

mostrar número da versão do programa e sair

13.1.10 ebook-polish

```
ebook-polish [options] input_file [output_file]
```

Polimento de livros tem a ver com buscar o brilho da perfeição em seus e-books cuidadosamente criados. Polir tenta minimizar as alterações no código interno do seu e-book. Diferente da conversão, ele nãoachata CSS, renomeia arquivos, muda tamanho de fonte, ajusta margens, etc. Cada ação realiza apenas o mínimo de mudanças necessárias para o efeito desejado.

Você deveria usar essa ferramenta como último passo na criação de e-books.

Note que embelezar funciona apenas em arquivos no formato AZW3 ou EPUB ou KEPUB.

Sempre que você for passar argumentos para **ebook-polish** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--add-soft-hyphens, -H

Adicionar hifens condicionais a todas as palavras do livro. Isto permite uma melhor exibição de texto justificado em leitores que não suportam hifenização.

--compress-images, -i

Comprime imagens no livro sem perda, para reduzir o seu tamanho sem afetar a qualidade da imagem.

--cover, -c

Caminho para uma imagem de capa. Altera a capa especificada no e-book. Se não houver capa ou a capa não estiver identificada corretamente, insere uma nova capa.

--download-external-resources, -d

Baixar recursos externos como imagens, folhas de estilo, etc, que apontam para URLs em vez de arquivos no livro. Todos esses recursos serão baixados e adicionados ao livro, de tal forma que o livro não faça mais referência a recursos externos.

--embed-fonts, -e

incorporar todas as fontes que são referenciadas no documento e ainda não estão incorporadas. Isto irá analisar o seu computador pelas fontes, e se eles forem encontradas, elas serão incorporadas no documento. Certifique-se que você tenha a licença adequada para incorporar as fontes usadas neste documento.

--help, -h

mostrar esta mensagem de ajuda e sair

--jacket, -j

Inserir uma "sobrecapa de livro" no começo do livro que contenha todos os metadados do livro, como título, etiquetas, autores, série, comentários, etc. Qualquer sobrecapa anterior será substituída.

--opf, -o

Caminho para um arquivo OPF. Os metadados no livro é atualizado do arquivo OPF.

--remove-jacket

Remover uma página de sobrecapa de livro inserida anteriormente.

--remove-soft-hyphens

Remover hifens condicionais de todo o texto do livro.

--remove-unused-css, -u

Remove todas as regras CSS não utilizadas de folhas de estilo e elementos <style>. Alguns livros criados a partir de modelos de produção podem ter um grande número de regras CSS extras que não correspondem a qualquer conteúdo real. Estas regras extras podem retardar leitores que precisam processar todas elas.

--smarten-punctuation, -p

Converter traços simples texto, reticências, aspas, vários hífens, etc em seus equivalentes tipograficamente corretos. Observe que o algoritmo pode, algumas vezes produzir resultados incorretos, especialmente quando aspas simples no começo das contrações estão envolvidas.

--subset-fonts, -f

Criar um subconjunto de fontes significa reduzir uma fonte incorporada para conter apenas os caracteres utilizados desta fonte no livro. Isso reduz consideravelmente o tamanho dos arquivos de fonte (reduzir o tamanho do arquivo da fonte pela metade é comum). Por exemplo, se o livro usa uma fonte específica para os cabeçalhos, então o subconjunto irá reduzir esse tipo de fonte para conter apenas os caracteres presentes nos cabeçalhos atuais no livro. Ou, se o livro incorpora as versões em negrito e itálico de uma fonte, mas o texto em negrito e itálico é relativamente raro, ou ausente por completo, então as fontes em negrito e itálico podem ser reduzidas para apenas alguns caracteres ou completamente removida. A única desvantagem para o subconjunto de fontes é que se, algum dia mais tarde, você decidir adicionar mais texto aos seus livros, o texto recém-adicionado pode não ser abrangido pela subconjunto da fonte.

--upgrade-book, -U

Atualize as estruturas internas do livro, se possível. Por exemplo, atualiza livros EPUB 2 para livros EPUB 3.

--verbose

Produz saída mais detalhada, útil para depuração.

--version

mostrar número da versão do programa e sair

13.1.11 ebook-viewer

```
ebook-viewer [opções] arquivo
```

Exibir um eBook.

Sempre que você for passar argumentos para **ebook-viewer** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--continue

Continuar lendo o último livro aberto

--detach

Desanexar do terminal de controle, se houver algum (somente Linux)

--force-reload

Forçar o recarregamento de todos os livros abertos

--full-screen, --fullscreen, -f

Se especificado, a janela do Visualizador de e-books tentará abrir em tela cheia ao iniciar.

--help, -h

mostrar esta mensagem de ajuda e sair

--new-instance

Abrir uma nova janela do visualizador mesmo quando a opção de usar apenas uma única instância está definida

--open-at

A posição em que será aberto o livro especificado. A posição é um local ou posição que você pode obter usando a ação Ir Para->Local nos controles do visualizador. Alternativamente, você pode usar a forma toc:algumacoisa e ele abrirá no local da primeira entrada do índice que contenha o texto "algumacoisa". A forma toc-href:algumacoisa irá corresponder ao href (link de destino interno) das entradas do índice. A correspondência é exata. Se você quiser combinar com uma parte do texto, use a forma toc-href-contains:algumacoisa. A forma ref:algumacoisa usará posições do Modo de Referência. A forma search:algumacoisa irá buscar "algumacoisa" após abrir o livro. A forma regex:algumacoisa irá buscar a expressão regular "algumacoisa" após abrir o livro.

--raise-window

Se especificado, a janela do Visualizador de e-books tentará vir para frente quando iniciada.

--version

mostrar número da versão do programa e sair

13.1.12 fetch-ebook-metadata

```
fetch-ebook-metadata [opções]
```

Buscar metadados do livro de fontes on-line. Você deve especificar pelo menos um de título, autores ou ISBN.

Sempre que você for passar argumentos para **fetch-ebook-metadata** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--allowed-plugin, -p

Especifique o nome de um plugin de download de metadados para usar. Por padrão, todos os plugins de metadados serão usados. Pode ser especificado várias vezes para vários plugins. Todos os nomes de plugin: Google, Google Images, Amazon.com, Edelweiss, Open Library, Big Book Search

--authors, -a

Autor(es) do(s) livro(s)

--cover, -c

Especifique um nome de arquivo. A capa, se disponível, será salva com este nome. Sem esta opção, nenhuma capa será baixada.

--help, -h

mostrar esta mensagem de ajuda e sair

--identifier, -I

Identificadores como ASIN/Goodreads id, etc. Pode ser especificado múltiplas vezes para múltiplos identificadores. Por exemplo:*--identifier* (página 339) asin:B0082BAJA0

--isbn, -i

ISBN do Livro

--opf, -o

Saída de metadados em formato OPF em vez de texto legível humano.

--timeout, -d

Tempo de inatividade em segundos. O padrão é 30

--title, -t

Título do livro

--verbose, -v

Imprimir o registro para o console (stderr)

--version

mostrar número da versão do programa e sair

13.1.13 lrf2lrs

```
lrf2lrs livro.lrf
```

Converte um arquivo LRF em um arquivo LRS (XML de codificação UTF-8)

Sempre que você for passar argumentos para **lrf2lrs** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--dont-output-resources
Não salve a imagem embutida e os arquivos fonte no o disco

--help, -h
mostrar esta mensagem de ajuda e sair

--output, -o
Arquivo de saída LRS

--verbose
Seja mais detalhado

--version
mostrar número da versão do programa e sair

13.1.14 lrfviewer

```
lrfviewer [opções] meulivro.lrf
```

Ler o e-book LRF meulivro.lrf.

Sempre que você for passar argumentos para **lrfviewer** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--disable-hyphenation
Desativar a hifenização. Deve acelerar significativamente a renderização.

--help, -h
mostrar esta mensagem de ajuda e sair

--profile
Falha ao renderizar LRF

--verbose
Imprimir mais informações sobre o processo de renderização

--version
mostrar número da versão do programa e sair

--visual-debug
Ativar recursos visuais para a depuração do mecanismo de renderização

--white-background
Por padrão, o fundo é esbranquiçado, pois eu acho isso mais fácil aos olhos. Utilize esta opção para tornar o fundo branco puro.

13.1.15 lrs2lrf

```
lrs2lrf [opções] arquivo.lrs
```

Compila um arquivo LRS em um arquivo LRF.

Sempre que você for passar argumentos para **lrs2lrf** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--help, -h

mostrar esta mensagem de ajuda e sair

--lrs

Converte LRS para LRS (útil para depuração).

--output, -o

Caminho para o arquivo de saída

--verbose

Processamento detalhado

--version

mostrar número da versão do programa e sair

13.1.16 web2disk

```
web2disk URL
```

Onde URL é, por exemplo, https://google.com

Sempre que você for passar argumentos para **web2disk** que possuem espaços em branco, coloque o argumento dentro de aspas. Por exemplo: "/some path/with spaces"

[opções]

--base-dir, -d

Pasta base no qual a URL é salva. O padrão é .

--delay

Intervalo mínimo, em segundos, entre buscas consecutivas. O padrão é de 0 s

--dont-download-stylesheets

Não baixar folhas de estilo CSS.

--encoding

A codificação de caracteres para os sites que você está tentando baixar. O padrão é tentar e adivinhar a codificação.

--filter-regexp

Qualquer link que corresponda a esta expressão regular será ignorado. Esta opção pode ser especificada várias vezes, neste caso, qualquer expressão regular que corresponda a um link, ele será ignorado. Por padrão, nenhum link será ignorado. Se filtrar expressão regular e combinar expressão regular forem especificados, então filtrar expressão regular é aplicado primeiro.

--help, -h

mostrar esta mensagem de ajuda e sair

--match-regexp

Somente links que correspondem a esta expressão regular serão seguidos. Esta opção pode ser especificada multiplas vezes, no caso, toda vez que um link corresponder a uma das expressões regulares, ele será seguido. Por padrão todos os links são seguidos.

--max-files, -n

O número máximo de arquivos para baixar. Isto só se aplica a arquivos de tags <a href>. O padrão é 9223372036854775807

--max-recursions, -r

Máximo número de níveis na recursividade, ou seja a profundidade de links para seguir. Padrão 1

--timeout, -t

Tempo em segundos para esperar uma resposta do servidor. Padrão: 10.0 s

--verbose

Mostrar informação detalhada de saída. Útil para depuração

--version

mostrar número da versão do programa e sair

13.2 Comandos não Documentados

- ebook-device
- markdown-calibre

Você pode ver o uso dos comandos não documentados executando-os sem argumentos em um terminal.

CAPÍTULO 14

Configurando um ambiente de desenvolvimento do calibre

calibre é completamente open source, licenciado sob o 'GNU GPL v3 <https://www.gnu.org/licenses/gpl.html>'_. Isso significa que você é livre para baixar e modificar o programa para o conteúdo do seu coração. Nesta seção, você aprenderá como configurar um ambiente de desenvolvimento de calibre no sistema operacional de sua escolha. calibre é escrito principalmente em 'Python <https://www.python.org>'_ com algum código C/C++ para velocidade e interface do sistema. Note que o calibre requer pelo menos Python 3.8.

Conteúdo

- *Filosofia de desenvolvimento* (página 344)
  - *Layout do código* (página 344)
- *Obtendo o código* (página 345)
  - *Enviando suas alterações para serem incluídas* (página 345)
- *Ambiente de Desenvolvimento Windows* (página 346)
- *ambiente de desenvolvimento no macOS* (página 347)
- *Ambiente de Desenvolvimento Linux* (página 348)
- *Ter instalações separadas "normal" e "desenvolvimento" do Calibre no mesmo computador* (página 348)
- *Debugging tips* (página 349)
  - *Usando declarações* `print()` (página 349)
  - *Usando um interpretador* Python *interativo* (página 349)
  - *Usando o depurador* Python *como depurador remoto* (página 349)
  - *Usar o depurador no seu IDE Python favorito* (página 350)
  - *Executing arbitrary scripts in the calibre Python environment* (página 350)
- *Usando o* Calibre *em seus projetos* (página 350)

- *Instalação binária de* Calibre (página 351)
- *Instalação de Origem no Linux* (página 351)

- *API documentation for various parts of calibre* (página 351)

14.1 Filosofia de desenvolvimento

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 generated/pt/cli-index.

O design modular do calibre é expresso através de "Plugins". Há um :ref:'tutorial <customize>' sobre a escrita de plugins de calibre. Por exemplo, adicionar suporte para um novo dispositivo ao calibre normalmente envolve escrever menos de 100 linhas de código na forma de um plug-in de driver de dispositivo. Você pode navegar pelo 'built-in drivers <https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices>'_. Da mesma forma, adicionar suporte para novos formatos de conversão envolve escrever plugins de formato de entrada/saída. Outro exemplo do design modular é o sistema :ref:'recipe <news>' para buscar notícias. Para obter mais exemplos de plug-ins projetados para adicionar recursos ao calibre, consulte o 'Índice de plug-ins <https://www.mobileread.com/forums/showthread.php?p=1362767#post1362767>'_.

14.1.1 Layout do código

Todo o código Python calibre está no pacote "calibre". Este pacote contém os seguintes subpacotes principais

- dispositivos - Todos os drivers de dispositivo. Olhe alguns dos drivers embutidos para ter uma ideia de como eles funcionam.
  - Para obter detalhes, consulte: "devices.interface" que define a interface suportada por drivers de dispositivo e "devices.usbms" que define um driver genérico que se conecta a um dispositivo USBMS. Todos os drivers baseados em USBMS em calibre herdam dele.
- e-books - Todo o código de conversão/metadados do e-book. Um bom ponto de partida é "calibre.ebooks.conversion.cli", que é o módulo que alimenta o comando :command:'ebook-convert'. O processo de conversão é controlado através de "conversion.plumber". O código independente de formato está todo em "ebooks.oeb" e o código dependente de formato está em "ebooks.format_name".
  - A leitura, escrita e download de metadados estão em `ebooks.metadata`
  - A conversão acontece em um pipeline, para a estrutura do pipeline, veja :ref:'conversion-introduction'. O pipeline consiste em um plugin de entrada, várias transformações e um plugin de saída. O código que constrói e conduz o pipeline está em :file:'plumber.py'. O pipeline funciona em uma representação de um e-book que é como um epub descompactado, com manifesto, coluna, toc, guia, conteúdo html, etc. A classe que gerencia essa representação é OEBBook em "ebooks.oeb.base". As várias transformações que são aplicadas ao livro durante as conversões vivem em :file:'oeb/transforms/*.py'. E os plugins de entrada e saída vivem em :file:'conversion/plugins/*.py'.
  - A edição de e-book acontece usando um objeto de contêiner diferente. Está documentado em :ref:'polish_api'.
- db - O back-end do banco de dados. Consulte :ref:'db_api' para a interface com a biblioteca de calibre.
- Servidor de conteúdo: "srv" é o servidor de conteúdo de calibre.
- gui2 - A Interface Gráfica do Usuário. A inicialização da GUI acontece em "gui2.main" e "gui2.ui". O e-book-viewer está em "gui2.viewer". O editor de e-book está em "gui2.tweak_book".

Se você quiser localizar os pontos de entrada para todos os vários executáveis de calibre, observe a estrutura "entry_points" em 'linux.py <https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.py>'_.

Se você precisar de ajuda para entender o código, poste no 'fórum de desenvolvimento <https://www.mobileread.com/forums/forumdisplay.php?f=240>'_ e você provavelmente receberá ajuda de um dos muitos desenvolvedores do calibre.

14.2 Obtendo o código

Você pode obter o código-fonte do calibre de duas maneiras, usando um sistema de controle de versão ou baixando diretamente um :website_base:'tarball <dist/src>'.

calibre usa 'Git <https://www.git-scm.com/>'_, um sistema de controle de versão distribuído. O Git está disponível em todas as plataformas suportadas. Depois de instalar o Git, você pode obter o código-fonte do calibre com o comando:

```
git clone https://github.com/kovidgoyal/calibre.git
```

No Windows, você precisará do nome completo do caminho, que será algo como :file:'C:\Arquivos de Programas\Git\git.exe'.

O calibre é um projeto muito grande com um histórico de controle de fonte muito longo, então o acima pode levar um tempo (10 minutos a uma hora, dependendo da velocidade da sua internet).

If you want to get the code faster, the source code for the latest release is always available as an archive[115].

To update a branch to the latest code, use the command:

```
git pull --no-edit
```

Também é possível navegar pelo código no GitHub[116].

14.2.1 Enviando suas alterações para 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[117]. 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[118]. Note that this will include *all* the commits you have made. If you only want to send some commits, you have to change `origin/master` above. To send only the last commit, use:

```
git format-patch HEAD~1 --stdout > my-changes
```

To send the last *n* commits, replace *1* with *n*, for example, for the last 3 commits:

```
git format-patch HEAD~3 --stdout > my-changes
```

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[119] 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:

[115] https://calibre-ebook.com/dist/src
[116] https://github.com/kovidgoyal/calibre
[117] https://bugs.launchpad.net/calibre
[118] https://bugs.launchpad.net/calibre
[119] https://github.com

- Configure git na sua máquina como descrito neste artigo: Configurar Git[120]
- Configure chaves ssh para autenticação no GitHub, como descrito aqui: Gerar Chaves SSH[121]
- Ir a https://github.com/kovidgoyal/calibre e clique no botão *Fork*.
- Em um Terminal faça:

```
git clone git@github.com:<username>/calibre.git
git remote add upstream https://github.com/kovidgoyal/calibre.git
```

  Substitua <username> em cima com o seu nome de utilizador do GitHub. Isto fará com que o seu "fork" seja verificado localmente.

- Pode efetuar alterações e confirmá-las sempre que desejar. Quando estiver pronto para mesclar o seu trabalho, faça um:

```
git push
```

  e vá para `https://github.com/<username>/calibre` e clique no botão *Pull Request* para gerar um pedido de submissão que pode ser unido.

- Pode atualizar a sua cópia local com o código do repositório principal a qualquer momento, fazendo:

```
git pull upstream
```

You should also keep an eye on the calibre development forum[122]. 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

Também deve obter o código fonte do Calibre separadamente, conforme descrito em cima.

Install calibre normally, using the Windows installer[123]. Then open a Command Prompt and change to the previously checked out calibre code folder. For example:

```
cd C:\Users\kovid\work\calibre
```

calibre é a pasta que contém os ficheiros SRC e as subpastas dos recursos.

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[124] to setting environment variables on Windows.

Once you have set the environment variable, open a new command prompt and check that it was correctly set by using the command:

```
echo %CALIBRE_DEVELOP_FROM%
```

[120] https://help.github.com/articles/set-up-git
[121] https://help.github.com/articles/generating-ssh-keys
[122] https://www.mobileread.com/forums/forumdisplay.php?f=240
[123] https://calibre-ebook.com/download_windows
[124] https://docs.python.org/using/windows.html#excursus-setting-environment-variables

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[125].

14.4 ambiente de desenvolvimento no macOS

> **Nota**
>
> Também deve obter o código fonte do Calibre separadamente, conforme descrito em cima.

Install calibre normally using the provided .dmg[126]. 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.

Crie um arquivo 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
```

Depois de fazer isso, 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.

[125] https://www.mobileread.com/forums/showthread.php?t=251201
[126] https://calibre-ebook.com/download_osx

14.5 Ambiente de Desenvolvimento Linux

> **Nota**
>
> Também deve obter o código fonte do Calibre separadamente, conforme descrito em cima.

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[127]. Then open a terminal and change to the previously checked out calibre code folder, for example:

```
cd /home/kovid/work/calibre
```

calibre é a pasta que contém os ficheiros SRC e as subpastas dos recursos.

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. You can get these by running `calibre-debug --paths`. But note that distribution calibre packages are often terminally broken and are completely unsupported.

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 Ter instalações separadas "normal" e "desenvolvimento" do Calibre no mesmo computador

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)

[127] https://calibre-ebook.com/download_linux

Para iniciar a versão de lançamento do Calibre com a sua biblioteca diária:

calibre-normal.bat:

```
calibre.exe "--with-library=C:\path\to\everyday\library folder"
```

calibre-dev.bat:

```
set CALIBRE_DEVELOP_FROM=C:\path\to\calibre\checkout\src
calibre.exe "--with-library=C:\path\to\test\library folder"
```

14.7 Debugging tips

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 Usando declarações `print()`

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
```

Da mesma forma, pode iniciar o visualizador de ebooks como:

```
calibre-debug -w /path/to/file/to/be/viewed
```

The e-book editor can be started as:

```
calibre-debug --edit-book /path/to/be/edited
```

14.7.2 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[128].

> **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.

14.8 Usando o *Calibre* em seus projetos

É possível usar funções/código do *Calibre* diretamente em seu projeto *Python*. Existem duas maneiras de fazer isso:

[128] https://docs.python.org/library/pdb.html#debugger-commands

14.8.1 Instalação binária de *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 no 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(**values*)

book_created = 4

When a new book record is created in the database, with the book id as the only argument

book_edited = 8

When a book format is edited, with arguments: (book_id, fmt)

books_removed = 5

When books are removed from the database with the list of book ids as the only argument

format_added = 2

When a format is added to a book, with arguments: (book_id, format)

formats_removed = 3

When formats are removed from a book, with arguments: (mapping of book id to set of formats removed from the book)

indexing_progress_changed = 9

When the indexing progress changes

items_removed = 7

When items such as tags or authors are removed from some books. Arguments: (field_name, affected book ids, ids of removed items)

items_renamed = 6

When items such as tags or authors are renamed in some or all books. Arguments: (field_name, affected book ids, map of old item id to new item id)

links_changed = 11

When the links associated with items(s) are changed, with arguments: (field_name, item_ids)

metadata_changed = 1

When some metadata is changed for some books, with arguments: (name of changed field, set of affected book ids)

notes_changed = 10

When the notes associated with item(s) are changed, with arguments: (field_name, item_ids)

add_books(*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)

Add the specified books to the library. Books should be an iterable of 2-tuples, each 2-tuple of the form `(mi, format_map)` where mi is a Metadata object and format_map is a dictionary of the form `{fmt: path_or_stream}`, for example: `{'EPUB': '/path/to/file.epub'}`.

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 357).

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_extra_files(*book_id*, *map_of_relpath_to_stream_or_path*, *replace=True*, *auto_rename=False*)

Add extra data files

add_format(*book_id*, *fmt*, *stream_or_path*, *replace=True*, *run_hooks=True*, *dbapi=None*)

Add a format to the specified book. Return True if the format was added successfully.

Parâmetros

- **replace** – Se Verdadeiro substitui o formato existente, caso contrário, se o formato já existe, retorna Falso.
- **run_hooks** – Se verdadeiro, os plugins de tipo de ficheiro são executados no formato antes e depois de serem adicionados.
- **dbapi** – Uso interno apenas.

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 351), library_id, event_type_specific_data)

add_notes_resource(*path_or_stream_or_data*, *name: str*, *mtime: float = None*) → int

Add the specified resource so it can be referenced by notes and return its content hash

all_annotation_types()

Return a tuple of all annotation types in the database.

all_annotation_users()

Return a tuple of all (user_type, user name) that have annotations.

all_annotations(*restrict_to_user=None*, *limit=None*, *annotation_type=None*, *ignore_removed=False*, *restrict_to_book_ids=None*)

Return a tuple of all annotations matching the specified criteria. *ignore_removed* controls whether removed (deleted) annotations are also returned. Removed annotations are just a skeleton used for merging of annotations.

all_annotations_for_book(*book_id*)

Return a tuple containing all annotations for the specified book_id as a dict with keys: *format*, *user_type*, *user*, *annotation*. Here, annotation is the annotation data.

all_book_ids(*type=<class 'frozenset'>*)

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)

annotation_count_for_book(*book_id*)

Return the number of annotations for the specified book available in the database.

annotations_map_for_book(*book_id*, *fmt*, *user_type='local'*, *user='viewer'*)

Return a map of annotation type -> annotation data for the specified book_id, format, user and user_type.

author_data(*author_ids=None*)

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*)

Comprime as imagens de capa dos livros especificados. Uma qualidade de compressão de 100 executará uma compressão sem perdas, caso contrário, terá perdas.

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 355) 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 357). This can be used for an implementation of *has_book()* (página 357) in a worker process without access to the db.

delete_annotations(*annot_ids*)

Delete annotations with the specified ids.

delete_custom_book_data(*name*, *book_ids=()*)

Delete data for name. By default deletes all data, if you only want to delete data for some book ids, pass in a list of book ids.

delete_trash_entry(*book_id*, *category*)

Delete an entry from the trash. Here category is 'b' for books and 'f' for formats.

embed_metadata(*book_ids*, *only_fmts=None*, *report_error=None*, *report_progress=None*)

Update metadata in all formats of the specified book_ids to current metadata in the database.

expire_old_trash()

Expire entries from the trash that are too old

export_note(*field*, *item_id*) → str

Export the note as a single HTML document with embedded images as data: URLs

fast_field_for(*field_obj*, *book_id*, *default_value=None*)

Same as field_for, except that it avoids the extra lookup to get the field object

field_for(*name*, *book_id*, *default_value=None*)

Return the value of the field `name` for the book identified by `book_id`. If no such book exists or it has no defined value for the field `name` or no such field exists, then `default_value` is returned.

`default_value` is not used for title, title_sort, authors, author_sort and series_index. This is because these always have values in the db. `default_value` is used for all custom columns.

The returned value for is_multiple fields are always tuples, even when no values are found (in other words, default_value is ignored). The exception is identifiers for which the returned value is always a dictionary. The returned tuples are always in link order, that is, the order in which they were created.

field_ids_for(*name*, *book_id*)

Return the ids (as a tuple) for the values that the field `name` has on the book identified by `book_id`. If there are no values, or no such book, or no such field, an empty tuple is returned.

field_supports_notes(*field=None*) → bool

Return True iff the specified field supports notes. If field is None return frozenset of all fields that support notes.

find_identical_books(*mi*, *search_restriction=''*, *book_ids=None*)

Finds books that have a superset of the authors in mi and the same title (title is fuzzy matched). See also *data_for_find_identical_books()* (página 354).

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 354) 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 354).

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 354) 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_all_items_that_have_notes(*field_name=None*) → set[int] | dict[str, set[int]]

Return all item_ids for items that have notes in the specified field or all fields if field_name is None

get_all_link_maps_for_book(*book_id*)

Returns all links for all fields referenced by book identified by book_id. If book_id doesn't exist then the method returns {}.

Example: Assume author A has link X, author B has link Y, tag S has link F, and tag T has link G. If book 1 has author A and tag T, this method returns {'authors':{'A':'X'}, 'tags':{'T', 'G'}}. If book 2's author is neither A nor B and has no tags, this method returns {}.

Parâmetros

book_id – the book id in question.

Retorna

{field: {field_value, link_value}, … for all fields with a field_value having a non-empty link value for that book

get_categories(*sort='name'*, *book_ids=None*, *already_fixed=None*, *first_letter_sort=False*, *uncollapsed_categories=None*)

Used internally to implement the Tag Browser

get_custom_book_data(*name*, *book_ids=()*, *default=None*)

Get data for name. By default returns data for all book_ids, pass in a list of book ids if you only want some data. Returns a map of book_id to values. If a particular value could not be decoded, uses default for it.

get_id_map(*field*)

Return a mapping of id numbers to values for the specified field. The field must be a many-one or many-many field, otherwise a ValueError is raised.

get_ids_for_custom_book_data(*name*)

Return the set of book ids for which name has data.

get_item_id(*field*, *item_name*, *case_sensitive=False*)

Return the item id for item_name or None if not found. This function is very slow if doing lookups for multiple names use either get_item_ids() or get_item_name_map(). Similarly, case sensitive lookups are faster than case insensitive ones.

get_item_ids(*field*, *item_names*, *case_sensitive=False*)

Return a dict mapping item_name to the item id or None

get_item_name(*field*, *item_id*)

Return the item name for the item specified by item_id in the specified field. See also `get_id_map()` (página 356).

get_item_name_map(*field*, *normalize_func=None*)

Return mapping of item values to ids

get_link_map(*for_field*)

Return a dictionary of links for the supplied field.

Parâmetros

for_field – the lookup name of the field for which the link map is desired

Retorna

{field_value:link_value, …} for non-empty links

get_metadata(*book_id*, *get_cover=False*, *get_user_categories=True*, *cover_as_data=False*)

Return metadata for the book identified by book_id as a `calibre.ebooks.metadata.book.base.Metadata` (página 211) 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_notes_resource(*resource_hash*) → dict | None

Return a dict containing the resource data and name or None if no resource with the specified hash is found

get_proxy_metadata(*book_id*)

Like `get_metadata()` (página 357) 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 354).

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.

import_note(*field*, *item_id*, *path_to_html_file*, *path_is_data=False*)

Import a previously exported note or an arbitrary HTML file as the note for the specified item

init()

Initialize this cache with data from the backend.

items_with_notes_in_book(*book_id: int*) → dict[str, dict[int, str]]

Return a dict of field to items that have associated notes for that field for the specified book

link_for(*field*, *item_id*)

Return the link, if any, for the specified item or None if no link is found

list_extra_files(*book_id*, *use_cache=False*, *pattern=''*) → tuple[ExtraFile, ...]

Get information about extra files in the book's directory.

Parâmetros

- **book_id** – the database book id for the book
- **pattern** – the pattern of filenames to search for. Empty pattern matches all extra files. Patterns must use / as separator. Use the DATA_FILE_PATTERN constant to match files inside the data directory.

Retorna

A tuple of all extra files matching the specified pattern. Each element of the tuple is ExtraFile(relpath, file_path, stat_result). Where relpath is the relative path of the file to the book directory using / as a separator. stat_result is the result of calling os.stat() on the file.

merge_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

Merge the specified annotations into the existing annotations for book_id, fm, user_type, and user.

merge_extra_files(*dest_id*, *src_ids*, *replace=False*)

Merge the extra files from src_ids into dest_id. Conflicting files are auto-renamed unless replace=True in which case they are replaced.

move_book_from_trash(*book_id*)

Undelete a book from the trash directory

move_format_from_trash(*book_id*, *fmt*)

Undelete a format from the trash directory

multisort(*fields*, *ids_to_sort=None*, *virtual_fields=None*)

Return a list of sorted book ids. If ids_to_sort is None, all book ids are returned.

fields must be a list of 2-tuples of the form (field_name, ascending=True or False). The most significant field is the first 2-tuple.

notes_data_for(*field*, *item_id*) → str

Return all notes data as a dict or None if note does not exist

notes_for(*field*, *item_id*) → str

Return the notes document or an empty string if not found

notes_resources_used_by(*field*, *item_id*)

Return the set of resource hashes of all resources used by the note for the specified item

pref(*name*, *default=None*, *namespace=None*)

Return the value for the specified preference or the value specified as `default` if the preference is not set.

read_backup(*book_id*)

Return the OPF metadata backup for the book as a bytestring or None if no such backup exists.

remove_books(*book_ids*, *permanent=False*)

Remove the books specified by the book_ids from the database and delete their format files. If `permanent` is False, then the format files are placed in the per-library trash directory.

remove_extra_files(*book_id: int*, *relpaths: Iterable[str]*, *permanent=False*) → dict[str, Exception | None]

Delete the specified extra files, either to Recycle Bin or permanently.

remove_formats (*formats_map*, *db_only=False*)

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.

Retorna

A map of book id to set of formats actually deleted from the filesystem for that book

remove_items (*field*, *item_ids*, *restrict_to_book_ids=None*)

Delete all items in the specified field with the specified ids. Returns the set of affected book ids. `restrict_to_book_ids` is an optional set of books ids. If specified the items will only be removed from those books.

rename_extra_files (*book_id*, *map_of_relpath_to_new_relpath*, *replace=False*)

Rename extra data files

rename_items (*field*, *item_id_to_new_name_map*, *change_index=True*, *restrict_to_book_ids=None*)

Rename items from a many-one or many-many field such as tags or series.

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.

search_annotations(*fts_engine_query*, *use_stemming=True*, *highlight_start=None*, *highlight_end=None*, *snippet_size=None*, *annotation_type=None*, *restrict_to_book_ids=None*, *restrict_to_user=None*, *ignore_removed=False*)

Return of a tuple of annotations matching the specified Full-text query.

search_notes(*fts_engine_query=''*, *use_stemming=True*, *highlight_start=None*, *highlight_end=None*, *snippet_size=None*, *restrict_to_fields=()*, *return_text=True*, *result_type=<class 'tuple'>*, *process_each_result=None*, *limit=None*)

Search the text of notes using an FTS index. If the query is empty return all notes.

set_annotations_for_book(*book_id*, *fmt*, *annots_list*, *user_type='local'*, *user='viewer'*)

Set all annotations for the specified book_id, fmt, user_type and user.

set_conversion_options(*options*, *fmt='PIPE'*)

options must be a map of the form {book_id:conversion_options}

set_cover(*book_id_data_map*)

Set the cover for this book. The data can be either a QImage, QPixmap, file object or bytestring. It can also be None, in which case any existing cover is removed.

set_field(*name*, *book_id_to_val_map*, *allow_case_change=True*, *do_path_update=True*)

Set the values of the field specified by `name`. Returns the set of all book ids that were affected by the change.

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_link_map(*field*, *value_to_link_map*, *only_set_if_no_existing_link=False*)

Sets links for item values in field. Note: this method doesn't change values not in the value_to_link_map

Parâmetros

- **field** – the lookup name
- **value_to_link_map** – dict(field_value:link, …). Note that these are values, not field ids.

Retorna

books changed by setting the link

set_metadata(*book_id*, *mi*, *ignore_errors=False*, *force_changes=False*, *set_title=True*, *set_authors=True*, *allow_case_change=False*)

Set metadata for the book *id* from the *Metadata* object *mi*

Setting force_changes=True will force set_metadata to update fields even if mi contains empty values. In this case, 'None' is distinguished from 'empty'. If mi.XXX is None, the XXX is not replaced, otherwise it is. The tags, identifiers, and cover attributes are special cases. Tags and identifiers cannot be set to None so they will always be replaced if force_changes is true. You must ensure that mi contains the values you want the book to have. Covers are always changed if a new cover is provided, but are never deleted. Also note that force_changes has no effect on setting title or authors.

set_notes_for(*field*, *item_id*, *doc: str*, *searchable_text: str = ''*, *resource_hashes=()*, *remove_unused_resources=False*) → int

Set the notes document. If the searchable text is different from the document, specify it as searchable_text. If the document references resources their hashes must be present in resource_hashes. Set remove_unused_resources to True to cleanup unused resources, note that updating a note automatically cleans up resources pertaining to that note anyway.

set_pref(*name*, *val*, *namespace=None*)

Set the specified preference to the specified value. See also `pref()` (página 358).

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).

unretire_note_for(*field*, *item_id*) → int

Unretire a previously retired note for the specified item. Notes are retired when an item is removed from the database

update_annotations(*annot_id_map*)

Update annotations.

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 ferramentas de edição de e-book

As ferramentas de edição de e-book consistem em um objeto `calibre.ebooks.oeb.polish.container.Container` (página 362) que representa um livro como uma coleção de arquivos HTML + de recursos e várias ferramentas que podem ser usadas para executar operações no contêiner. Todas as ferramentas estão na forma de funções de nível de módulo nos vários módulos `calibre.ebooks.oeb.polish.*`. Você obtém um conteiner object para um livro com um caminho desta forma:

```
from calibre.ebooks.oeb.polish.container import get_container
container = get_container('Path to book file', tweak_mode=True)
```

Se você estiver escrevendo um plugin para o editor de E-book, você obterá o contêiner atual para o livro que está sendo editado assim:

```
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 Container

class calibre.ebooks.oeb.polish.container.**Container**(*rootpath=None*, *opfpath=None*, *log=<calibre.utils.logging.Log object>*, *clone_data=None*)

A container represents an open e-book as a folder full of files and an OPF file. There are two important concepts:

- The root folder. This is the base of the e-book. All the e-books files are inside this folder or in its sub-folders.
- Names: These are paths to the books' files relative to the root folder. They always contain POSIX separators and are unquoted. They can be thought of as canonical identifiers for files in the book. Most methods on the container object work with names. Names are always in the NFC Unicode normal form.
- Clones: o objecto 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 directamente aos ficheiros no sistema de ficheiros. Em vez disso, use `raw_data()` (página 365) ou `open()` (página 364) 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=b''*, *media_type=None*, *spine_index=None*, *modify_name_if_needed=False*, *process_manifest_item=None*, *suggested_id=''*)

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*, *suggested_id=''*)

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 arquivos todos os objetos analisados que tenham sido identificados com problemas e escrever o arquivo 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*)

Efetiva a escrita de um objeto analisado para o disco (será serializado e escrito no arquivo subjacente). Se `keep_parsed for Verdadeiro a representação analisada é mantida na Cache. Ver também: *parsed()* (página 365)

dirty(*name*)

Marcar o objecto analisado correspondente ao nome como impuro. Ver também: *parsed()* (página 365).

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 363)

filesize(*name*)

Devolve o tamanho em bytes do ficheiro representado pelo nome canónico especificado. Manipula automaticamente os objectos analisados como impuros. Ver também: *parsed()* (página 365)

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 objecto 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 363) 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 ascendente (ou acrescenta se Index for None), ajustando a indentação. Apenas funciona com itens que 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 arquivo apontado por nome para leitura/escrita direta. Note que isto efetivará as alterações ao arquivo se o mesmo se encontrar deteriorado e irá removê-lo do cache do interpretador. É necessário terminar o trabalho neste arquivo antes de acessar novamente a 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 363) de modo a que o contentor saiba que tem de atualizar a cache. Veja também `replace()` (página 365).

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 objecto 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 362) 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 objectos analisados correspondentes ao nome por obj, que tem de ser um objecto 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 365).

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 366) and *spine_items* (página 365).

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 366) and *spine_items* (página 365).

property spine_names

An iterator yielding name and is_linear for every item in the books' spine. See also: *spine_iter* (página 366) and *spine_items* (página 365).

Gerenciando arquivos de componentes em um contêiner

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

Gerenciando capas 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.

Dividindo e mesclando arquivos

`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 367). 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.

Gerenciando 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

Trabalhando 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.

Trabalhando 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*)

Criar um índice incorporado (HTML) a partir de um índice NCX existente.

Parâmetros

`title` – O título para este índice.

Editar ferramenta de livro

`class` `calibre.gui2.tweak_book.plugin.`**`Tool`**

Base: `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 256) object to which this tool belongs.
- self. *`boss`* (página 369)
- self. *`gui`* (página 369)

Methods that must be overridden in sub classes:

- *`create_action()`* (página 370)
- *`register_shortcut()`* (página 369)

`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 370) object. Used to control the user interface.

`property gui`

The main window of the user interface

`property current_container`

Devolve o *`calibre.ebooks.oeb.polish.container.Container`* (página 362) objeto atual que representa o livro a ser editado.

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:
        # 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
```

Ver também

Método *register_shortcut()* (página 369).

Controlando a interface do usuário 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()

Efetive quaisquer alterações que o usuário tenha feito aos arquivos abertos nos editores para o contêiner. Deve-se chamar este método antes de executar quaisquer ações no contêiner atual

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

Gerenciamento de Direitos Digitais (DRM)

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[129].

15.1 O que a GDD implica para mim pessoalmente?

Quando você compra um e-book com DRM, você realmente não o possui, mas comprou a permissão para usá-lo da maneira ditada pelo vendedor. O DRM limita o que você pode fazer com os e-books que você "comprou". Frequentemente, as pessoas que compram livros com DRM não estão cientes da extensão dessas restrições. Essas restrições evitam que você reformate o e-book de acordo com sua preferência, incluindo alterações estilísticas, como ajustar os tamanhos das fontes, embora haja um software que permite que você faça essas coisas para livros que não sejam DRM. As pessoas costumam se surpreender com o fato de um e-book que compraram em um formato específico não poder ser convertido para outro formato se o e-book tiver DRM. Portanto, se você tem um Amazon Kindle e compra um livro vendido pela *Barnes and Nobles*, você deve saber que, se o e-book tiver DRM, você não poderá lê-lo em seu *Kindle*. Observe que estou falando sobre um livro que você compra, não rouba ou pirateia, mas COMPRA.

15.2 O que a GDD faz para os autores?

Editores de e-books com DRM argumentam que o DRM é para o bem dos autores e para proteger sua integridade artística e prevenir a pirataria. Mas o DRM NÃO impede a pirataria. Pessoas que desejam piratear conteúdo ou usar conteúdo pirateado ainda o fazem e têm sucesso. Os três principais esquemas de DRM para e-books hoje são executados pela *Amazon*, *Adobe* e *Barnes and Noble* e todos os três esquemas de DRM foram quebrados. Tudo o que o DRM faz é incomodar usuários legítimos. Pode-se argumentar que isso realmente prejudica os autores, pois as pessoas que

[129] https://en.wikipedia.org/wiki/Digital_rights_management

teriam comprado o livro optam por encontrar uma versão pirata, pois não estão dispostas a tolerar o DRM. Aqueles que piratearíam na ausência de DRM também o fazem em sua presença. Para reiterar, o ponto chave é que o DRM *não impede a pirataria*. Portanto, o DRM não é apenas inútil e prejudicial aos compradores de e-books, mas também um desperdício de dinheiro.

15.3 GDD 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 Por que o calibre não apoia a GDD?

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 visão do calibre em relação aos provedores de conteúdo?

Acreditamos firmemente que os autores e outros provedores de conteúdo devem ser compensados por seus esforços, mas o DRM não é a melhor maneira de fazer isso. Estamos desenvolvendo este banco de dados de e-books sem DRM de várias fontes para ajudá-lo a encontrar alternativas sem DRM e para ajudar os autores e editores independentes de e-books sem DRM a divulgar seu conteúdo. Esperamos que isso seja útil e solicitamos que não pirateie o conteúdo disponibilizado para você aqui.

15.6 Como eu posso ajudar a lutar contra a GDD?

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 *(do inglês Really Simple Syndication, 'agregação verdadeiramente simples')* é um formato de web feed para publicações cujo conteúdo é atualizado com frequência, como notícias de jornais e postagens de blogs. Trata-se de um formato específico para leitura de máquina e é a melhor forma para capturar conteúdo da internet em um e-book. Existem vários formatos de feed em uso na internet e o calibre oferece suporte à maioria, especialmente ao *ATOM*, comumente utilizado em blogs.

Receita

Receita é um conjunto de instruções que orienta o calibre na conversão de uma fonte de notícias online, como uma revista ou um blog, em um e-book. Uma receita é essencialmente um código Python[130] e, por isso, é capaz de converter em e-book fontes de notícia arbitrariamente complexas. De uma perspectiva mais simples, trata-se de um um conjunto de variáveis, como URLs, que dão ao calibre informação suficiente para buscar na internet as notícias a serem baixadas.

Linguagem de marcação de hipertexto.

HTML * (Linguagem de marcação de hipertexto)*, um subconjunto da Linguagem padrão de marcação generalizada (SGML) para publicação eletrônica, é o padrão específico usado para a World Wide Web.

Folhas de estilo em cascata.

CSS *(Folhas de estilo em cascata)* é uma linguagem usada para descrever como um documento HTML deveria ser escrito (estilo visual).

Interface de programação de aplicativos.

API *(Interface de Programação de Aplicativos)* é uma interface de código-fonte que fornece uma biblioteca para suportar pedidos de apoio para serviços a serem feitos por programas de computador.

Extensão de arquivos utilizada em e-readers da Sony

LRF Formato de e-book específico para dispositivos Sony.

Localizador padrão de recursos.

URL * (Localizador Padrão de Recursos)* por exemplo: "http://example.com"

expressão regular

Expressões regulares fornece uma forma concisa e flexível de identificar cadeias de caracteres de texto com inte-

[130] https://www.python.org

resse, tais como caracteres especiais, palavras ou padrões de caracteres. Ver *tutorial* (página 215) para introdução a expressões regulares.

Índice de Módulos Python

C

calibre.customize, 256
calibre.customize.conversion, 265
calibre.db.cache, 351
calibre.devices.interface, 268
calibre.ebooks.metadata.book.base, 211
calibre.ebooks.metadata.sources.base, 262
calibre.ebooks.oeb.polish.container, 361
calibre.ebooks.oeb.polish.cover, 367
calibre.ebooks.oeb.polish.css, 368
calibre.ebooks.oeb.polish.jacket, 367
calibre.ebooks.oeb.polish.pretty, 366
calibre.ebooks.oeb.polish.replace, 366
calibre.ebooks.oeb.polish.split, 367
calibre.ebooks.oeb.polish.toc, 369
calibre.gui2.tweak_book.boss, 370
calibre.utils.formatter_functions, 185
calibre.web.feeds.news, 41

Índice

Símbolos

-1
    calibredb-add opção de linha de comando, 316
-H
    ebook-polish opção de linha de comando, 337
-I
    calibredb-add opção de linha de comando, 315
    fetch-ebook-metadata opção de linha de comando, 339
-S
    calibredb-add opção de linha de comando, 315
-T
    calibredb-add opção de linha de comando, 315
-U
    ebook-polish opção de linha de comando, 338
--access-log
    calibre-server opção de linha de comando, 307
--add
    calibredb-add opção de linha de comando, 315
--add-alt-text-to-img
    ebook-convert opção de linha de comando, 331
--add-plugin
    calibre-customize opção de linha de comando, 304
--add-simple-plugin
    calibre-debug opção de linha de comando, 305
--add-soft-hyphens
    ebook-polish opção de linha de comando, 337
--ajax-timeout
    calibre-server opção de linha de comando, 307
--all
    calibredb-backup_metadata opção de linha de comando, 323
    calibredb-export opção de linha de comando, 317
--allowed-plugin
    fetch-ebook-metadata opção de linha de comando, 339
--append
    calibredb-set_custom opção de linha de comando, 321
--as-extra-data-file
    calibredb-add_format opção de linha de comando, 316
--as-opf
    calibredb-show_metadata opção de linha de comando, 317
--ascending
    calibredb-list opção de linha de comando, 314
--asciiize
    ebook-convert opção de linha de comando, 328
--attachment
    calibre-smtp opção de linha de comando, 311
--auth-mode
    calibre-server opção de linha de comando, 307
--author-sort
    ebook-convert opção de linha de comando, 333
    ebook-meta opção de linha de comando, 335
--authors
    calibredb-add opção de linha de comando, 315
    ebook-convert opção de linha de comando,

333
ebook-meta opção de linha de comando, 335
fetch-ebook-metadata opção de linha de comando, 339
--auto-reload
calibre-server opção de linha de comando, 307
--automerge
calibredb-add opção de linha de comando, 315
--ban-after
calibre-server opção de linha de comando, 307
--ban-for
calibre-server opção de linha de comando, 307
--base-dir
web2disk opção de linha de comando, 341
--base-font-size
ebook-convert opção de linha de comando, 328
--book-list-mode
calibre-server opção de linha de comando, 308
--book-producer
ebook-convert opção de linha de comando, 333
ebook-meta opção de linha de comando, 335
--build-plugin
calibre-customize opção de linha de comando, 304
--cafile
calibre-smtp opção de linha de comando, 311
--catalog-title
calibredb-catalog opção de linha de comando, 319
--categories
calibredb-list_categories opção de linha de comando, 322
--category
ebook-meta opção de linha de comando, 335
--change-justification
ebook-convert opção de linha de comando, 328
--chapter
ebook-convert opção de linha de comando, 332
--chapter-mark
ebook-convert opção de linha de comando, 332
--command
calibre-debug opção de linha de comando, 305
--comments
ebook-convert opção de linha de comando, 333
ebook-meta opção de linha de comando, 335
--compress-images
ebook-polish opção de linha de comando, 337
--compress-min-size
calibre-server opção de linha de comando, 308
--continue
ebook-viewer opção de linha de comando, 338
--cover
calibredb-add opção de linha de comando, 315
ebook-convert opção de linha de comando, 333
ebook-meta opção de linha de comando, 335
ebook-polish opção de linha de comando, 337
fetch-ebook-metadata opção de linha de comando, 339
--cross-reference-authors
calibredb-catalog opção de linha de comando, 319
--csv
calibredb-check_library opção de linha de comando, 322
calibredb-list_categories opção de linha de comando, 323
--custom-list-template
calibre-server opção de linha de comando, 308
--customize-plugin
calibre-customize opção de linha de comando, 304
--daemonize
calibre-server opção de linha de comando, 308
--date
ebook-meta opção de linha de comando, 335
--debug-device-driver
calibre-debug opção de linha de comando, 305
--debug-pipeline
calibredb-catalog opção de linha de comando, 319
ebook-convert opção de linha de comando, 334
--default-programs
calibre-debug opção de linha de comando, 305
--delay

web2disk opção de linha de comando, 341
--detach
calibre opção de linha de comando, 304
ebook-edit opção de linha de comando, 334
ebook-viewer opção de linha de comando, 338
--details
calibredb-custom_columns opção de linha de comando, 321
--dialect
calibredb-list_categories opção de linha de comando, 323
--diff
calibre-debug opção de linha de comando, 305
--disable-allow-socket-preallocation
calibre-server opção de linha de comando, 308
--disable-auth
calibre-server opção de linha de comando, 308
--disable-dehyphenate
ebook-convert opção de linha de comando, 330
--disable-delete-blank-paragraphs
ebook-convert opção de linha de comando, 330
--disable-fallback-to-detected-interface
calibre-server opção de linha de comando, 308
--disable-fix-indents
ebook-convert opção de linha de comando, 330
--disable-font-rescaling
ebook-convert opção de linha de comando, 328
--disable-format-scene-breaks
ebook-convert opção de linha de comando, 330
--disable-hyphenation
lrfviewer opção de linha de comando, 340
--disable-italicize-common-cases
ebook-convert opção de linha de comando, 330
--disable-local-write
calibre-server opção de linha de comando, 308
--disable-log-not-found
calibre-server opção de linha de comando, 308
--disable-markup-chapter-headings
ebook-convert opção de linha de comando, 331
--disable-plugin
calibre-customize opção de linha de comando, 304
--disable-remove-fake-margins
ebook-convert opção de linha de comando, 332
--disable-renumber-headings
ebook-convert opção de linha de comando, 331
--disable-unwrap-lines
ebook-convert opção de linha de comando, 331
--disable-use-bonjour
calibre-server opção de linha de comando, 308
--disable-use-sendfile
calibre-server opção de linha de comando, 309
--disallow-rendered-cover
ebook-meta opção de linha de comando, 335
--display
calibredb-add_custom_column opção de linha de comando, 321
--displayed-fields
calibre-server opção de linha de comando, 308
--do-not-match-on-related-words
calibredb-fts_search opção de linha de comando, 325
--dont-asciiize
calibredb-export opção de linha de comando, 317
--dont-download-stylesheets
web2disk opção de linha de comando, 341
--dont-output-resources
lrf2lrs opção de linha de comando, 340
--dont-replace
calibredb-add_format opção de linha de comando, 316
--dont-save-cover
calibredb-export opção de linha de comando, 317
--dont-save-extra-files
calibredb-export opção de linha de comando, 318
--dont-split-on-page-breaks
ebook-convert opção de linha de comando, 327
--dont-update-metadata
calibredb-export opção de linha de comando, 318
--dont-verify-server-certificate
calibre-smtp opção de linha de comando, 311
--dont-write-opf

calibredb-export opção de linha de comando, 318
--download-external-resources
ebook-polish opção de linha de comando, 337
--duplicate-links-in-toc
ebook-convert opção de linha de comando, 332
--duplicates
calibredb-add opção de linha de comando, 315
--edit-book
calibre-debug opção de linha de comando, 305
--embed-all-fonts
ebook-convert opção de linha de comando, 328
--embed-font-family
ebook-convert opção de linha de comando, 328
--embed-fonts
ebook-polish opção de linha de comando, 337
--empty
calibredb-add opção de linha de comando, 315
--enable-allow-socket-preallocation
calibre-server opção de linha de comando, 308
--enable-auth
calibre-server opção de linha de comando, 308
--enable-fallback-to-detected-interface
calibre-server opção de linha de comando, 308
--enable-heuristics
ebook-convert opção de linha de comando, 331
--enable-local-write
calibre-server opção de linha de comando, 308
--enable-log-not-found
calibre-server opção de linha de comando, 308
--enable-plugin
calibre-customize opção de linha de comando, 304
--enable-use-bonjour
calibre-server opção de linha de comando, 308
--enable-use-sendfile
calibre-server opção de linha de comando, 309
--encoding
web2disk opção de linha de comando, 341
--encryption-method
calibre-smtp opção de linha de comando, 311
--epub-flatten
ebook-convert opção de linha de comando, 327
--epub-inline-toc
ebook-convert opção de linha de comando, 327
--epub-max-image-size
ebook-convert opção de linha de comando, 327
--epub-toc-at-end
ebook-convert opção de linha de comando, 327
--epub-version
ebook-convert opção de linha de comando, 327
--exclude-genre
calibredb-catalog opção de linha de comando, 319
--exclusion-rules
calibredb-catalog opção de linha de comando, 319
--exec-file
calibre-debug opção de linha de comando, 305
--expand-css
ebook-convert opção de linha de comando, 328
--explode-book
calibre-debug opção de linha de comando, 306
--export-all-calibre-data
calibre-debug opção de linha de comando, 306
--extra-css
ebook-convert opção de linha de comando, 328
--extract-to
ebook-convert opção de linha de comando, 327
--field
calibredb-set_metadata opção de linha de comando, 317
--fields
calibredb-list opção de linha de comando, 314
--filter-css
ebook-convert opção de linha de comando, 328
--filter-regexp
web2disk opção de linha de comando, 341

--flow-size
ebook-convert opção de linha de comando, 327
--font-size-mapping
ebook-convert opção de linha de comando, 329
--for-machine
calibredb-list opção de linha de comando, 314
--force
calibredb-remove_custom_column opção de linha de comando, 321
--force-reload
ebook-viewer opção de linha de comando, 338
--fork
calibre-smtp opção de linha de comando, 311
--formats
calibredb-export opção de linha de comando, 318
--from-opf
ebook-convert opção de linha de comando, 334
ebook-meta opção de linha de comando, 335
--full-screen
ebook-viewer opção de linha de comando, 338
--fullscreen
ebook-viewer opção de linha de comando, 338
--generate-authors
calibredb-catalog opção de linha de comando, 319
--generate-descriptions
calibredb-catalog opção de linha de comando, 319
--generate-genres
calibredb-catalog opção de linha de comando, 319
--generate-recently-added
calibredb-catalog opção de linha de comando, 319
--generate-series
calibredb-catalog opção de linha de comando, 319
--generate-titles
calibredb-catalog opção de linha de comando, 319
--genre-source-field
calibredb-catalog opção de linha de comando, 319
--get-cover
ebook-meta opção de linha de comando, 335
--gui
calibre-debug opção de linha de comando, 306
--gui-debug
calibre-debug opção de linha de comando, 306
--header-note-source-field
calibredb-catalog opção de linha de comando, 319
--help
calibre opção de linha de comando, 304
calibre-customize opção de linha de comando, 304
calibre-debug opção de linha de comando, 306
calibre-server opção de linha de comando, 309
calibre-smtp opção de linha de comando, 311
ebook-convert opção de linha de comando, 326
ebook-edit opção de linha de comando, 334
ebook-meta opção de linha de comando, 336
ebook-polish opção de linha de comando, 337
ebook-viewer opção de linha de comando, 338
fetch-ebook-metadata opção de linha de comando, 339
lrf2lrs opção de linha de comando, 340
lrfviewer opção de linha de comando, 340
lrs2lrf opção de linha de comando, 341
opção de linha de comando, 313
web2disk opção de linha de comando, 341
--html-unwrap-factor
ebook-convert opção de linha de comando, 331
--identifier
calibredb-add opção de linha de comando, 315
ebook-meta opção de linha de comando, 336
fetch-ebook-metadata opção de linha de comando, 339
--ids
calibredb-catalog opção de linha de comando, 318
--ignore
calibredb-add opção de linha de comando, 316
--ignore_extensions
calibredb-check_library opção de linha de comando, 322
--ignore_names
calibredb-check_library opção de linha

de comando, 322
--ignore-plugins
calibre opção de linha de comando, 304
--ignored-fields
calibre-server opção de linha de comando, 309
--implode-book
calibre-debug opção de linha de comando, 306
--import-calibre-data
calibre-debug opção de linha de comando, 306
--include-snippets
calibredb-fts_search opção de linha de comando, 325
--index
ebook-meta opção de linha de comando, 336
--indexing-speed
calibredb-fts_index opção de linha de comando, 324
--indexing-threshold
calibredb-fts_search opção de linha de comando, 325
--input-encoding
ebook-convert opção de linha de comando, 326
--input-profile
ebook-convert opção de linha de comando, 326
--insert-blank-line
ebook-convert opção de linha de comando, 329
--insert-blank-line-size
ebook-convert opção de linha de comando, 329
--insert-metadata
ebook-convert opção de linha de comando, 332
--inspect-mobi
calibre-debug opção de linha de comando, 306
--is-multiple
calibredb-add_custom_column opção de linha de comando, 321
--isbn
calibredb-add opção de linha de comando, 315
ebook-convert opção de linha de comando, 333
ebook-meta opção de linha de comando, 336
fetch-ebook-metadata opção de linha de comando, 339
--item_count
calibredb-list_categories opção de linha de comando, 323
--jacket
ebook-polish opção de linha de comando, 337
--keep-ligatures
ebook-convert opção de linha de comando, 329
--kepubify
calibre-debug opção de linha de comando, 306
--language
ebook-convert opção de linha de comando, 333
ebook-meta opção de linha de comando, 336
--languages
calibredb-add opção de linha de comando, 315
--level1-toc
ebook-convert opção de linha de comando, 332
--level2-toc
ebook-convert opção de linha de comando, 332
--level3-toc
ebook-convert opção de linha de comando, 333
--library-path
opção de linha de comando, 313
--limit
calibredb-list opção de linha de comando, 314
calibredb-search opção de linha de comando, 324
--line-height
ebook-convert opção de linha de comando, 329
--line-width
calibredb-list opção de linha de comando, 314
--linearize-tables
ebook-convert opção de linha de comando, 329
--list-fields
calibredb-set_metadata opção de linha de comando, 317
--list-plugins
calibre-customize opção de linha de comando, 305
--list-recipes
ebook-convert opção de linha de comando, 326
--listen-on
calibre-server opção de linha de comando,

309
--localhost
calibre-smtp opção de linha de comando, 311
--log
calibre-server opção de linha de comando, 309
--lrf-bookid
ebook-meta opção de linha de comando, 336
--lrs
lrs2lrf opção de linha de comando, 341
--manage-users
calibre-server opção de linha de comando, 309
--margin-bottom
ebook-convert opção de linha de comando, 329
--margin-left
ebook-convert opção de linha de comando, 329
--margin-right
ebook-convert opção de linha de comando, 329
--margin-top
ebook-convert opção de linha de comando, 329
--match-end-marker
calibredb-fts_search opção de linha de comando, 325
--match-regexp
web2disk opção de linha de comando, 341
--match-start-marker
calibredb-fts_search opção de linha de comando, 325
--max-files
web2disk opção de linha de comando, 342
--max-header-line-size
calibre-server opção de linha de comando, 309
--max-job-time
calibre-server opção de linha de comando, 309
--max-jobs
calibre-server opção de linha de comando, 309
--max-log-size
calibre-server opção de linha de comando, 309
--max-opds-items
calibre-server opção de linha de comando, 309
--max-opds-ungrouped-items
calibre-server opção de linha de comando, 309
--max-recursions
web2disk opção de linha de comando, 342
--max-request-body-size
calibre-server opção de linha de comando, 309
--max-toc-links
ebook-convert opção de linha de comando, 333
--merge-comments-rule
calibredb-catalog opção de linha de comando, 320
--minimum-line-height
ebook-convert opção de linha de comando, 329
--new-instance
ebook-viewer opção de linha de comando, 338
--no-chapters-in-toc
ebook-convert opção de linha de comando, 333
--no-default-epub-cover
ebook-convert opção de linha de comando, 327
--no-svg-cover
ebook-convert opção de linha de comando, 327
--no-update-check
calibre opção de linha de comando, 304
--num-per-page
calibre-server opção de linha de comando, 309
--one-book-per-directory
calibredb-add opção de linha de comando, 316
--only-formats
calibredb-embed_metadata opção de linha de comando, 324
--open-at
ebook-viewer opção de linha de comando, 338
--opf
ebook-polish opção de linha de comando, 337
fetch-ebook-metadata opção de linha de comando, 339
--outbox
calibre-smtp opção de linha de comando, 311
--output
lrf2lrs opção de linha de comando, 340
lrs2lrf opção de linha de comando, 341
--output-format
calibredb-fts_search opção de linha de comando, 325

--output-profile
calibredb-catalog opção de linha de comando, 320
ebook-convert opção de linha de comando, 326
--page-breaks-before
ebook-convert opção de linha de comando, 332
--password
calibre-smtp opção de linha de comando, 312
opção de linha de comando, 313
--paths
calibre-debug opção de linha de comando, 306
--permanent
calibredb-remove opção de linha de comando, 316
--pidfile
calibre-server opção de linha de comando, 310
--port
calibre-server opção de linha de comando, 310
calibre-smtp opção de linha de comando, 312
--prefer-metadata-cover
ebook-convert opção de linha de comando, 332
--prefix
calibredb-list opção de linha de comando, 314
--prefix-rules
calibredb-catalog opção de linha de comando, 320
--preserve-cover-aspect-ratio
ebook-convert opção de linha de comando, 327
--preset
calibredb-catalog opção de linha de comando, 320
--pretty-print
ebook-convert opção de linha de comando, 327
--profile
lrfviewer opção de linha de comando, 340
--progress
calibredb-export opção de linha de comando, 318
--pubdate
ebook-convert opção de linha de comando, 333
--publisher
ebook-convert opção de linha de comando, 333
ebook-meta opção de linha de comando, 336
--raise-window
ebook-viewer opção de linha de comando, 339
--rating
ebook-convert opção de linha de comando, 334
ebook-meta opção de linha de comando, 336
--read-metadata-from-opf
ebook-convert opção de linha de comando, 334
--really-do-it
calibredb-restore_database opção de linha de comando, 322
--recurse
calibredb-add opção de linha de comando, 316
--relay
calibre-smtp opção de linha de comando, 312
--remove-first-image
ebook-convert opção de linha de comando, 332
--remove-jacket
ebook-polish opção de linha de comando, 337
--remove-paragraph-spacing
ebook-convert opção de linha de comando, 330
--remove-paragraph-spacing-indent-size
ebook-convert opção de linha de comando, 330
--remove-plugin
calibre-customize opção de linha de comando, 305
--remove-soft-hyphens
ebook-polish opção de linha de comando, 337
--remove-unused-css
ebook-polish opção de linha de comando, 337
--replace-scene-breaks
ebook-convert opção de linha de comando, 331
--replace-whitespace
calibredb-export opção de linha de comando, 318
--report
calibredb-check_library opção de linha de comando, 322
--restrict-to
calibredb-fts_search opção de linha de comando, 325

--run-plugin
calibre-debug opção de linha de comando, 306
--run-test
calibre-debug opção de linha de comando, 306
--run-without-debug
calibre-debug opção de linha de comando, 306
--search
calibredb-catalog opção de linha de comando, 318
calibredb-list opção de linha de comando, 314
--search-replace
ebook-convert opção de linha de comando, 331
--search-the-net-urls
calibre-server opção de linha de comando, 310
--select-text
ebook-edit opção de linha de comando, 334
--separator
calibredb-list opção de linha de comando, 314
--series
calibredb-add opção de linha de comando, 315
ebook-convert opção de linha de comando, 334
ebook-meta opção de linha de comando, 336
--series-index
calibredb-add opção de linha de comando, 315
ebook-convert opção de linha de comando, 334
--shutdown-running-calibre
calibre opção de linha de comando, 304
calibre-debug opção de linha de comando, 307
--shutdown-timeout
calibre-server opção de linha de comando, 310
--single-dir
calibredb-export opção de linha de comando, 318
--smarten-punctuation
ebook-convert opção de linha de comando, 330
ebook-polish opção de linha de comando, 337
--sort-by
calibredb-list opção de linha de comando, 314
--sr1-replace
ebook-convert opção de linha de comando, 331
--sr1-search
ebook-convert opção de linha de comando, 331
--sr2-replace
ebook-convert opção de linha de comando, 331
--sr2-search
ebook-convert opção de linha de comando, 331
--sr3-replace
ebook-convert opção de linha de comando, 331
--sr3-search
ebook-convert opção de linha de comando, 331
--ssl-certfile
calibre-server opção de linha de comando, 310
--ssl-keyfile
calibre-server opção de linha de comando, 310
--start-in-tray
calibre opção de linha de comando, 304
--start-reading-at
ebook-convert opção de linha de comando, 332
--subject
calibre-smtp opção de linha de comando, 311
--subset-embedded-fonts
ebook-convert opção de linha de comando, 330
--subset-font
calibre-debug opção de linha de comando, 307
--subset-fonts
ebook-polish opção de linha de comando, 337
--tags
calibredb-add opção de linha de comando, 315
ebook-convert opção de linha de comando, 334
ebook-meta opção de linha de comando, 336
--template
calibredb-export opção de linha de comando, 318
calibredb-list opção de linha de comando, 314
--template_file
calibredb-list opção de linha de comando,

314
--template_heading
calibredb-list opção de linha de comando, 314
--test-build
calibre-debug opção de linha de comando, 307
--thumb-width
calibredb-catalog opção de linha de comando, 320
--timefmt
calibredb-export opção de linha de comando, 318
--timeout
calibre-server opção de linha de comando, 310
calibre-smtp opção de linha de comando, 311
fetch-ebook-metadata opção de linha de comando, 339
opção de linha de comando, 313
web2disk opção de linha de comando, 342
--timestamp
ebook-convert opção de linha de comando, 334
--title
calibredb-add opção de linha de comando, 315
ebook-convert opção de linha de comando, 334
ebook-meta opção de linha de comando, 336
fetch-ebook-metadata opção de linha de comando, 339
--title-sort
ebook-convert opção de linha de comando, 334
ebook-meta opção de linha de comando, 336
--to-dir
calibredb-export opção de linha de comando, 318
--to-lowercase
calibredb-export opção de linha de comando, 318
--to-opf
ebook-meta opção de linha de comando, 336
--toc-filter
ebook-convert opção de linha de comando, 333
--toc-threshold
ebook-convert opção de linha de comando, 333
--toc-title
ebook-convert opção de linha de comando, 328
--transform-css-rules
ebook-convert opção de linha de comando, 330
--transform-html-rules
ebook-convert opção de linha de comando, 330
--trusted-ips
calibre-server opção de linha de comando, 310
--un-kepubify
calibre-debug opção de linha de comando, 307
--unsmarten-punctuation
ebook-convert opção de linha de comando, 330
--upgrade-book
ebook-polish opção de linha de comando, 338
--url-prefix
calibre-server opção de linha de comando, 310
--use-auto-toc
ebook-convert opção de linha de comando, 333
--use-existing-cover
calibredb-catalog opção de linha de comando, 320
--userdb
calibre-server opção de linha de comando, 310
--username
calibre-smtp opção de linha de comando, 312
opção de linha de comando, 313
--vacuum-fts-db
calibredb-check_library opção de linha de comando, 322
--verbose
calibre opção de linha de comando, 304
calibre-smtp opção de linha de comando, 311
calibredb-catalog opção de linha de comando, 319
ebook-convert opção de linha de comando, 334
ebook-polish opção de linha de comando, 338
fetch-ebook-metadata opção de linha de comando, 339
lrf2lrs opção de linha de comando, 340
lrfviewer opção de linha de comando, 340
lrs2lrf opção de linha de comando, 341
web2disk opção de linha de comando, 342
--version

calibre opção de linha de comando, 304
calibre-customize opção de linha de comando, 305
calibre-debug opção de linha de comando, 307
calibre-server opção de linha de comando, 310
calibre-smtp opção de linha de comando, 311
ebook-convert opção de linha de comando, 326
ebook-edit opção de linha de comando, 335
ebook-meta opção de linha de comando, 336
ebook-polish opção de linha de comando, 338
ebook-viewer opção de linha de comando, 339
fetch-ebook-metadata opção de linha de comando, 339
lrf2lrs opção de linha de comando, 340
lrfviewer opção de linha de comando, 340
lrs2lrf opção de linha de comando, 341
opção de linha de comando, 313
web2disk opção de linha de comando, 342
--viewer
calibre-debug opção de linha de comando, 307
--visual-debug
lrfviewer opção de linha de comando, 340
--wait-for-completion
calibredb-fts_index opção de linha de comando, 324
--white-background
lrfviewer opção de linha de comando, 340
--width
calibredb-list_categories opção de linha de comando, 323
--with-library
calibre opção de linha de comando, 304
opção de linha de comando, 313
--worker-count
calibre-server opção de linha de comando, 310
-a
calibre-customize opção de linha de comando, 304
calibre-smtp opção de linha de comando, 311
calibredb-add opção de linha de comando, 315
calibredb-set_custom opção de linha de comando, 321
ebook-meta opção de linha de comando, 335
fetch-ebook-metadata opção de linha de comando, 339
-b
calibre-customize opção de linha de comando, 304
-c
calibre-debug opção de linha de comando, 305
calibredb-add opção de linha de comando, 315
calibredb-check_library opção de linha de comando, 322
calibredb-list_categories opção de linha de comando, 323
ebook-meta opção de linha de comando, 335
ebook-polish opção de linha de comando, 337
fetch-ebook-metadata opção de linha de comando, 339
-d
calibre-debug opção de linha de comando, 305
calibredb-add opção de linha de comando, 315
calibredb-custom_columns opção de linha de comando, 321
ebook-convert opção de linha de comando, 334
ebook-meta opção de linha de comando, 335
ebook-polish opção de linha de comando, 337
fetch-ebook-metadata opção de linha de comando, 339
web2disk opção de linha de comando, 341
-e
calibre-debug opção de linha de comando, 305
calibre-smtp opção de linha de comando, 311
calibredb-add opção de linha de comando, 315
calibredb-check_library opção de linha de comando, 322
ebook-polish opção de linha de comando, 337
-f
calibre-debug opção de linha de comando, 307
calibre-smtp opção de linha de comando, 311
calibredb-embed_metadata opção de linha de comando, 324
calibredb-list opção de linha de comando, 314
calibredb-remove_custom_column opção de

linha de comando, 321
calibredb-set_metadata opção de linha de comando, 317
ebook-polish opção de linha de comando, 337
ebook-viewer opção de linha de comando, 338
-g
calibre-debug opção de linha de comando, 306
-h
calibre opção de linha de comando, 304
calibre-customize opção de linha de comando, 304
calibre-debug opção de linha de comando, 306
calibre-server opção de linha de comando, 309
calibre-smtp opção de linha de comando, 311
ebook-convert opção de linha de comando, 326
ebook-edit opção de linha de comando, 334
ebook-meta opção de linha de comando, 336
ebook-polish opção de linha de comando, 337
ebook-viewer opção de linha de comando, 338
fetch-ebook-metadata opção de linha de comando, 339
lrf2lrs opção de linha de comando, 340
lrfviewer opção de linha de comando, 340
lrs2lrf opção de linha de comando, 341
opção de linha de comando, 313
web2disk opção de linha de comando, 341
-i
calibre-debug opção de linha de comando, 306
calibredb-add opção de linha de comando, 315
calibredb-catalog opção de linha de comando, 318
calibredb-list_categories opção de linha de comando, 323
ebook-meta opção de linha de comando, 336
ebook-polish opção de linha de comando, 337
fetch-ebook-metadata opção de linha de comando, 339
-j
ebook-polish opção de linha de comando, 337
-k
ebook-meta opção de linha de comando, 335
-l
calibre-customize opção de linha de comando, 305
calibre-smtp opção de linha de comando, 311
calibredb-add opção de linha de comando, 315
calibredb-search opção de linha de comando, 324
calibredb-set_metadata opção de linha de comando, 317
ebook-meta opção de linha de comando, 336
-m
calibre-debug opção de linha de comando, 306
calibredb-add opção de linha de comando, 315
ebook-convert opção de linha de comando, 334
-n
calibredb-check_library opção de linha de comando, 322
web2disk opção de linha de comando, 342
-o
calibre-smtp opção de linha de comando, 311
ebook-polish opção de linha de comando, 337
fetch-ebook-metadata opção de linha de comando, 339
lrf2lrs opção de linha de comando, 340
lrs2lrf opção de linha de comando, 341
-p
calibre-smtp opção de linha de comando, 312
ebook-meta opção de linha de comando, 336
ebook-polish opção de linha de comando, 337
fetch-ebook-metadata opção de linha de comando, 339
-r
calibre-customize opção de linha de comando, 305
calibre-debug opção de linha de comando, 306
calibre-smtp opção de linha de comando, 312
calibredb-add opção de linha de comando, 316
calibredb-check_library opção de linha de comando, 322
calibredb-list_categories opção de linha de comando, 322
calibredb-restore_database opção de li-

nha de comando, 322
ebook-meta opção de linha de comando, 336
web2disk opção de linha de comando, 342
-s
calibre opção de linha de comando, 304
calibre-debug opção de linha de comando, 307
calibre-smtp opção de linha de comando, 311
calibredb-add opção de linha de comando, 315
calibredb-catalog opção de linha de comando, 318
calibredb-list opção de linha de comando, 314
ebook-meta opção de linha de comando, 336
-t
calibre-debug opção de linha de comando, 306
calibre-smtp opção de linha de comando, 311
calibredb-add opção de linha de comando, 315
calibredb-list opção de linha de comando, 314
ebook-meta opção de linha de comando, 336
fetch-ebook-metadata opção de linha de comando, 339
web2disk opção de linha de comando, 342
-u
calibre-smtp opção de linha de comando, 312
ebook-polish opção de linha de comando, 337
-v
calibre opção de linha de comando, 304
calibre-smtp opção de linha de comando, 311
calibredb-catalog opção de linha de comando, 319
ebook-convert opção de linha de comando, 334
fetch-ebook-metadata opção de linha de comando, 339
-w
calibre-debug opção de linha de comando, 307
calibredb-list opção de linha de comando, 314
calibredb-list_categories opção de linha de comando, 323
-x
calibre-debug opção de linha de comando, 306

A

abort_article() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
abort_recipe_processing() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
abspath_to_name() (*método calibre.ebooks.oeb.polish.container.Container*), 362
accept_drag_move_event() (*método calibre.gui2.actions.InterfaceAction*), 283
accept_enter_event() (*método calibre.gui2.actions.InterfaceAction*), 283
accepts_drops (*atributo calibre.gui2.actions.InterfaceAction*), 283
action_add_menu (*atributo calibre.gui2.actions.InterfaceAction*), 282
action_menu_clone_qaction (*atributo calibre.gui2.actions.InterfaceAction*), 282
action_shortcut_name (*atributo calibre.gui2.actions.InterfaceAction*), 282
action_spec (*atributo calibre.gui2.actions.InterfaceAction*), 282
action_type (*atributo calibre.gui2.actions.InterfaceAction*), 282
add_annotation_to_library() (*método calibre.devices.usbms.device.Device*), 279
add_book() (*método calibre.devices.interface.BookList*), 276
add_books() (*método calibre.db.cache.Cache*), 352
add_books_to_metadata() (*método calibre.devices.usbms.driver.USBMS*), 281
add_books_to_metadata() (*método de classe calibre.devices.interface.DevicePlugin*), 272
add_custom_book_data() (*método calibre.db.cache.Cache*), 352
add_extra_files() (*método calibre.db.cache.Cache*), 352
add_file() (*método calibre.ebooks.oeb.polish.container.Container*), 362
add_format() (*método calibre.db.cache.Cache*), 352
add_listener() (*método calibre.db.cache.Cache*), 352
add_name_to_manifest() (*método calibre.ebooks.oeb.polish.container.Container*), 362
add_notes_resource() (*método calibre.db.cache.Cache*), 352
add_or_replace_jacket() (*no módulo calibre.ebooks.oeb.polish.jacket*), 367
add_properties() (*método calibre.ebooks.oeb.polish.container.Container*), 362
add_savepoint() (*método calibre.gui2.tweak_book.boss.Boss*), 370

add_toc_thumbnail() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
adeify_images() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 41
all_annotation_types() (*método calibre.db.cache.Cache*), 353
all_annotation_users() (*método calibre.db.cache.Cache*), 353
all_annotations() (*método calibre.db.cache.Cache*), 353
all_annotations_for_book() (*método calibre.db.cache.Cache*), 353
all_book_ids() (*método calibre.db.cache.Cache*), 353
all_field_for() (*método calibre.db.cache.Cache*), 353
all_field_ids() (*método calibre.db.cache.Cache*), 353
all_field_keys() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
all_field_names() (*método calibre.db.cache.Cache*), 353
all_non_none_fields() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
allowed_in_menu (*atributo calibre.gui2.tweak_book.plugin.Tool*), 369
allowed_in_toolbar (*atributo calibre.gui2.tweak_book.plugin.Tool*), 369
annotation_count_for_book() (*método calibre.db.cache.Cache*), 353
annotations_map_for_book() (*método calibre.db.cache.Cache*), 353
apply_container_update_to_gui() (*método calibre.gui2.tweak_book.boss.Boss*), 371
apply_unique_properties() (*método calibre.ebooks.oeb.polish.container.Container*), 362
articles_are_obfuscated (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
ASK_TO_ALLOW_CONNECT (*atributo calibre.devices.interface.DevicePlugin*), 269
author (*atributo calibre.customize.InterfaceActionBase*), 285
author (*atributo calibre.customize.MetadataReaderPlugin*), 260
author (*atributo calibre.customize.MetadataWriterPlugin*), 261
author (*atributo calibre.customize.Plugin*), 256
author (*atributo calibre.customize.PreferencesPlugin*), 285
author (*atributo calibre.devices.usbms.driver.USBMS*), 279
author (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
author_data() (*método calibre.db.cache.Cache*), 353
author_sort_from_authors() (*método calibre.db.cache.Cache*), 353
auto_cleanup (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
auto_cleanup_keep (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
auto_repeat (*atributo calibre.gui2.actions.InterfaceAction*), 282
auto_trim_covers (*atributo calibre.ebooks.metadata.sources.base.Source*), 262

B

BasicNewsRecipe (*classe em calibre.web.feeds.news*), 41
BCD (*atributo calibre.devices.interface.DevicePlugin*), 268
BCD (*atributo calibre.devices.usbms.device.Device*), 277
book_class (*atributo calibre.devices.usbms.driver.USBMS*), 280
book_created (*atributo calibre.db.cache.Cache.EventType*), 351
book_edited (*atributo calibre.db.cache.Cache.EventType*), 351
book_type (*atributo calibre.ebooks.oeb.polish.container.Container*), 362
BookList (*classe em calibre.devices.interface*), 275
booklist_class (*atributo calibre.devices.usbms.driver.USBMS*), 280
books() (*método calibre.devices.interface.DevicePlugin*), 272
books() (*método calibre.devices.usbms.driver.USBMS*), 280
books_for_field() (*método calibre.db.cache.Cache*), 353
books_in_virtual_library() (*método calibre.db.cache.Cache*), 353
books_removed (*atributo calibre.db.cache.Cache.EventType*), 351
Boss (*classe em calibre.gui2.tweak_book.boss*), 370
boss (*propriedade calibre.gui2.tweak_book.plugin.Tool*), 369
browser_type (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
BuiltinAdd (*classe em calibre.utils.formatter_functions*), 185
BuiltinAnd (*classe em calibre.utils.formatter_functions*), 186
BuiltinAnnotationCount (*classe em calibre.utils.formatter_functions*), 203
BuiltinApproximateFormats (*classe em calibre.utils.formatter_functions*), 203
BuiltinArguments (*classe em calibre.utils.formatter_functions*), 207
BuiltinAssign (*classe em calibre.utils.formatter_functions*), 207

BuiltinAuthorLinks (*classe em calibre.utils.formatter_functions*), 203
BuiltinAuthorSorts (*classe em calibre.utils.formatter_functions*), 203
BuiltinBookCount (*classe em calibre.utils.formatter_functions*), 190
BuiltinBooksize (*classe em calibre.utils.formatter_functions*), 204
BuiltinBookValues (*classe em calibre.utils.formatter_functions*), 191
BuiltinCapitalize (*classe em calibre.utils.formatter_functions*), 185
BuiltinCeiling (*classe em calibre.utils.formatter_functions*), 185
BuiltinCharacter (*classe em calibre.utils.formatter_functions*), 200
BuiltinCheckYesNo (*classe em calibre.utils.formatter_functions*), 200
BuiltinCmp (*classe em calibre.utils.formatter_functions*), 210
BuiltinConnectedDeviceName (*classe em calibre.utils.formatter_functions*), 204
BuiltinConnectedDeviceUUID (*classe em calibre.utils.formatter_functions*), 204
BuiltinContains (*classe em calibre.utils.formatter_functions*), 200
BuiltinCount (*classe em calibre.utils.formatter_functions*), 196
BuiltinCurrentLibraryName (*classe em calibre.utils.formatter_functions*), 204
BuiltinCurrentLibraryPath (*classe em calibre.utils.formatter_functions*), 204
BuiltinCurrentVirtualLibraryName (*classe em calibre.utils.formatter_functions*), 204
BuiltinDateArithmetic (*classe em calibre.utils.formatter_functions*), 193
BuiltinDaysBetween (*classe em calibre.utils.formatter_functions*), 194
BuiltinDivide (*classe em calibre.utils.formatter_functions*), 185
BuiltinEncodeForURL (*classe em calibre.utils.formatter_functions*), 194
BuiltinEval (*classe em calibre.utils.formatter_functions*), 209
BuiltinExtraFileModtime (*classe em calibre.utils.formatter_functions*), 191
BuiltinExtraFileNames (*classe em calibre.utils.formatter_functions*), 191
BuiltinExtraFileSize (*classe em calibre.utils.formatter_functions*), 192
BuiltinField (*classe em calibre.utils.formatter_functions*), 204
BuiltinFieldExists (*classe em calibre.utils.formatter_functions*), 201
BuiltinFieldListCount (*classe em calibre.utils.formatter_functions*), 196
BuiltinFinishFormatting (*classe em calibre.utils.formatter_functions*), 187
BuiltinFirstMatchingCmp (*classe em calibre.utils.formatter_functions*), 210
BuiltinFirstNonEmpty (*classe em calibre.utils.formatter_functions*), 210
BuiltinFloor (*classe em calibre.utils.formatter_functions*), 185
BuiltinFormatDate (*classe em calibre.utils.formatter_functions*), 187
BuiltinFormatDateField (*classe em calibre.utils.formatter_functions*), 188
BuiltinFormatDuration (*classe em calibre.utils.formatter_functions*), 188
BuiltinFormatNumber (*classe em calibre.utils.formatter_functions*), 190
BuiltinFormatsModtimes (*classe em calibre.utils.formatter_functions*), 205
BuiltinFormatsPaths (*classe em calibre.utils.formatter_functions*), 205
BuiltinFormatsPathSegments (*classe em calibre.utils.formatter_functions*), 205
BuiltinFormatsSizes (*classe em calibre.utils.formatter_functions*), 206
BuiltinFractionalPart (*classe em calibre.utils.formatter_functions*), 186
BuiltinGetLink (*classe em calibre.utils.formatter_functions*), 192
BuiltinGetNote (*classe em calibre.utils.formatter_functions*), 192
BuiltinGlobals (*classe em calibre.utils.formatter_functions*), 208
BuiltinHasCover (*classe em calibre.utils.formatter_functions*), 206
BuiltinHasExtraFiles (*classe em calibre.utils.formatter_functions*), 193
BuiltinHasNote (*classe em calibre.utils.formatter_functions*), 193
BuiltinHumanReadable (*classe em calibre.utils.formatter_functions*), 190
BuiltinIdentifierInList (*classe em calibre.utils.formatter_functions*), 208
BuiltinIfempty (*classe em calibre.utils.formatter_functions*), 201
BuiltinInList (*classe em calibre.utils.formatter_functions*), 208
BuiltinIsDarkMode (*classe em calibre.utils.formatter_functions*), 208
BuiltinIsMarked (*classe em calibre.utils.formatter_functions*), 206
BuiltinLanguageCodes (*classe em calibre.utils.formatter_functions*), 206

BuiltinLanguageStrings (*classe em calibre.utils.formatter_functions*), 206
BuiltinListCountMatching (*classe em calibre.utils.formatter_functions*), 197
BuiltinListDifference (*classe em calibre.utils.formatter_functions*), 197
BuiltinListEquals (*classe em calibre.utils.formatter_functions*), 197
BuiltinListIntersection (*classe em calibre.utils.formatter_functions*), 197
BuiltinListitem (*classe em calibre.utils.formatter_functions*), 209
BuiltinListJoin (*classe em calibre.utils.formatter_functions*), 197
BuiltinListRe (*classe em calibre.utils.formatter_functions*), 198
BuiltinListReGroup (*classe em calibre.utils.formatter_functions*), 198
BuiltinListRemoveDuplicates (*classe em calibre.utils.formatter_functions*), 198
BuiltinListSort (*classe em calibre.utils.formatter_functions*), 198
BuiltinListSplit (*classe em calibre.utils.formatter_functions*), 198
BuiltinListUnion (*classe em calibre.utils.formatter_functions*), 199
BuiltinLookup (*classe em calibre.utils.formatter_functions*), 211
BuiltinLowercase (*classe em calibre.utils.formatter_functions*), 185
BuiltinMakeUrl (*classe em calibre.utils.formatter_functions*), 194
BuiltinMakeUrlExtended (*classe em calibre.utils.formatter_functions*), 195
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*), 206
BuiltinOr (*classe em calibre.utils.formatter_functions*), 187
BuiltinPrint (*classe em calibre.utils.formatter_functions*), 208
BuiltinQueryString (*classe em calibre.utils.formatter_functions*), 195
BuiltinRange (*classe em calibre.utils.formatter_functions*), 199
BuiltinRatingToStars (*classe em calibre.utils.formatter_functions*), 190
BuiltinRawField (*classe em calibre.utils.formatter_functions*), 206
BuiltinRawList (*classe em calibre.utils.formatter_functions*), 207
BuiltinRe (*classe em calibre.utils.formatter_functions*), 201
BuiltinReGroup (*classe em calibre.utils.formatter_functions*), 201
BuiltinRound (*classe em calibre.utils.formatter_functions*), 186
BuiltinSelect (*classe em calibre.utils.formatter_functions*), 209
BuiltinSeriesSort (*classe em calibre.utils.formatter_functions*), 207
BuiltinSetGlobals (*classe em calibre.utils.formatter_functions*), 208
BuiltinShorten (*classe em calibre.utils.formatter_functions*), 201
BuiltinStrcat (*classe em calibre.utils.formatter_functions*), 202
BuiltinStrcatMax (*classe em calibre.utils.formatter_functions*), 202
BuiltinStrcmp (*classe em calibre.utils.formatter_functions*), 210
BuiltinStrcmpcase (*classe em calibre.utils.formatter_functions*), 210
BuiltinStrInList (*classe em calibre.utils.formatter_functions*), 209
BuiltinStrlen (*classe em calibre.utils.formatter_functions*), 202
BuiltinSubitems (*classe em calibre.utils.formatter_functions*), 199
BuiltinSublist (*classe em calibre.utils.formatter_functions*), 200
BuiltinSubstr (*classe em calibre.utils.formatter_functions*), 202
BuiltinSubtract (*classe em calibre.utils.formatter_functions*), 186
BuiltinSwapAroundArticles (*classe em calibre.utils.formatter_functions*), 202
BuiltinSwapAroundComma (*classe em calibre.utils.formatter_functions*), 202
BuiltinSwitch (*classe em calibre.utils.formatter_functions*), 211
BuiltinSwitchIf (*classe em calibre.utils.formatter_functions*), 211
BuiltinTemplate (*classe em calibre.utils.formatter_functions*), 209
BuiltinTest (*classe em calibre.utils.formatter_functions*), 202
BuiltinTitlecase (*classe em calibre.utils.formatter_functions*), 185
BuiltinToday (*classe em calibre.utils.formatter_functions*), 194
BuiltinToHex (*classe em calibre.utils.formatter_functions*), 196

BuiltinTransliterate (*classe em calibre.utils.formatter_functions*), 203
BuiltinUppercase (*classe em calibre.utils.formatter_functions*), 185
BuiltinUrlsFromIdentifiers (*classe em calibre.utils.formatter_functions*), 196
BuiltinUserCategories (*classe em calibre.utils.formatter_functions*), 207
BuiltinVirtualLibraries (*classe em calibre.utils.formatter_functions*), 207

C

Cache (*classe em calibre.db.cache*), 351
Cache.EventType (*classe em calibre.db.cache*), 351
cached_cover_url_is_reliable (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
calibre opção de linha de comando
    --detach, 304
    --help, 304
    --ignore-plugins, 304
    --no-update-check, 304
    --shutdown-running-calibre, 304
    --start-in-tray, 304
    --verbose, 304
    --version, 304
    --with-library, 304
    -h, 304
    -s, 304
    -v, 304
calibre.customize
    module, 256
calibre.customize.conversion
    module, 265
calibre.db.cache
    module, 351
calibre.devices.interface
    module, 268
calibre.ebooks.metadata.book.base
    module, 211
calibre.ebooks.metadata.sources.base
    module, 262
calibre.ebooks.oeb.polish.container
    module, 361
calibre.ebooks.oeb.polish.cover
    module, 367
calibre.ebooks.oeb.polish.css
    module, 368
calibre.ebooks.oeb.polish.jacket
    module, 367
calibre.ebooks.oeb.polish.pretty
    module, 366
calibre.ebooks.oeb.polish.replace
    module, 366
calibre.ebooks.oeb.polish.split
    module, 367
calibre.ebooks.oeb.polish.toc
    module, 369
calibre.gui2.tweak_book.boss
    module, 370
calibre.utils.formatter_functions
    module, 184
calibre.web.feeds.news
    module, 41
calibre-customize opção de linha de comando
    --add-plugin, 304
    --build-plugin, 304
    --customize-plugin, 304
    --disable-plugin, 304
    --enable-plugin, 304
    --help, 304
    --list-plugins, 305
    --remove-plugin, 305
    --version, 305
    -a, 304
    -b, 304
    -h, 304
    -l, 305
    -r, 305
calibre-debug opção de linha de comando
    --add-simple-plugin, 305
    --command, 305
    --debug-device-driver, 305
    --default-programs, 305
    --diff, 305
    --edit-book, 305
    --exec-file, 305
    --explode-book, 306
    --export-all-calibre-data, 306
    --gui, 306
    --gui-debug, 306
    --help, 306
    --implode-book, 306
    --import-calibre-data, 306
    --inspect-mobi, 306
    --kepubify, 306
    --paths, 306
    --run-plugin, 306
    --run-test, 306
    --run-without-debug, 306
    --shutdown-running-calibre, 307
    --subset-font, 307
    --test-build, 307
    --un-kepubify, 307
    --version, 307
    --viewer, 307
    -c, 305
    -d, 305
    -e, 305

-f, 307
-g, 306
-h, 306
-i, 306
-m, 306
-r, 306
-s, 307
-t, 306
-w, 307
-x, 306
calibre-server opção de linha de comando
--access-log, 307
--ajax-timeout, 307
--auth-mode, 307
--auto-reload, 307
--ban-after, 307
--ban-for, 307
--book-list-mode, 308
--compress-min-size, 308
--custom-list-template, 308
--daemonize, 308
--disable-allow-socket-preallocation, 308
--disable-auth, 308
--disable-fallback-to-detected--interface, 308
--disable-local-write, 308
--disable-log-not-found, 308
--disable-use-bonjour, 308
--disable-use-sendfile, 309
--displayed-fields, 308
--enable-allow-socket-preallocation, 308
--enable-auth, 308
--enable-fallback-to-detected--interface, 308
--enable-local-write, 308
--enable-log-not-found, 308
--enable-use-bonjour, 308
--enable-use-sendfile, 309
--help, 309
--ignored-fields, 309
--listen-on, 309
--log, 309
--manage-users, 309
--max-header-line-size, 309
--max-job-time, 309
--max-jobs, 309
--max-log-size, 309
--max-opds-items, 309
--max-opds-ungrouped-items, 309
--max-request-body-size, 309
--num-per-page, 309
--pidfile, 310
--port, 310
--search-the-net-urls, 310
--shutdown-timeout, 310
--ssl-certfile, 310
--ssl-keyfile, 310
--timeout, 310
--trusted-ips, 310
--url-prefix, 310
--userdb, 310
--version, 310
--worker-count, 310
-h, 309
calibre-smtp opção de linha de comando
--attachment, 311
--cafile, 311
--dont-verify-server-certificate, 311
--encryption-method, 311
--fork, 311
--help, 311
--localhost, 311
--outbox, 311
--password, 312
--port, 312
--relay, 312
--subject, 311
--timeout, 311
--username, 312
--verbose, 311
--version, 311
-a, 311
-e, 311
-f, 311
-h, 311
-l, 311
-o, 311
-p, 312
-r, 312
-s, 311
-t, 311
-u, 312
-v, 311
calibredb-add opção de linha de comando
-1, 316
-I, 315
-S, 315
-T, 315
--add, 315
--authors, 315
--automerge, 315
--cover, 315
--duplicates, 315
--empty, 315
--identifier, 315
--ignore, 316
--isbn, 315

--languages, 315
--one-book-per-directory, 316
--recurse, 316
--series, 315
--series-index, 315
--tags, 315
--title, 315
-a, 315
-c, 315
-d, 315
-e, 315
-i, 315
-l, 315
-m, 315
-r, 316
-s, 315
-t, 315
calibredb-add_custom_column opção de linha de comando
--display, 321
--is-multiple, 321
calibredb-add_format opção de linha de comando
--as-extra-data-file, 316
--dont-replace, 316
calibredb-backup_metadata opção de linha de comando
--all, 323
calibredb-catalog opção de linha de comando
--catalog-title, 319
--cross-reference-authors, 319
--debug-pipeline, 319
--exclude-genre, 319
--exclusion-rules, 319
--generate-authors, 319
--generate-descriptions, 319
--generate-genres, 319
--generate-recently-added, 319
--generate-series, 319
--generate-titles, 319
--genre-source-field, 319
--header-note-source-field, 319
--ids, 318
--merge-comments-rule, 320
--output-profile, 320
--prefix-rules, 320
--preset, 320
--search, 318
--thumb-width, 320
--use-existing-cover, 320
--verbose, 319
-i, 318
-s, 318
-v, 319
calibredb-check_library opção de linha de comando
--csv, 322
--ignore_extensions, 322
--ignore_names, 322
--report, 322
--vacuum-fts-db, 322
-c, 322
-e, 322
-n, 322
-r, 322
calibredb-custom_columns opção de linha de comando
--details, 321
-d, 321
calibredb-embed_metadata opção de linha de comando
--only-formats, 324
-f, 324
calibredb-export opção de linha de comando
--all, 317
--dont-asciiize, 317
--dont-save-cover, 317
--dont-save-extra-files, 318
--dont-update-metadata, 318
--dont-write-opf, 318
--formats, 318
--progress, 318
--replace-whitespace, 318
--single-dir, 318
--template, 318
--timefmt, 318
--to-dir, 318
--to-lowercase, 318
calibredb-fts_index opção de linha de comando
--indexing-speed, 324
--wait-for-completion, 324
calibredb-fts_search opção de linha de comando
--do-not-match-on-related-words, 325
--include-snippets, 325
--indexing-threshold, 325
--match-end-marker, 325
--match-start-marker, 325
--output-format, 325
--restrict-to, 325
calibredb-list opção de linha de comando
--ascending, 314
--fields, 314
--for-machine, 314
--limit, 314
--line-width, 314
--prefix, 314

--search, 314
--separator, 314
--sort-by, 314
--template, 314
--template_file, 314
--template_heading, 314
-f, 314
-s, 314
-t, 314
-w, 314
calibredb-list_categories opção de linha de comando
--categories, 322
--csv, 323
--dialect, 323
--item_count, 323
--width, 323
-c, 323
-i, 323
-r, 322
-w, 323
calibredb-remove opção de linha de comando
--permanent, 316
calibredb-remove_custom_column opção de linha de comando
--force, 321
-f, 321
calibredb-restore_database opção de linha de comando
--really-do-it, 322
-r, 322
calibredb-search opção de linha de comando
--limit, 324
-l, 324
calibredb-set_custom opção de linha de comando
--append, 321
-a, 321
calibredb-set_metadata opção de linha de comando
--field, 317
--list-fields, 317
-f, 317
-l, 317
calibredb-show_metadata opção de linha de comando
--as-opf, 317
can_be_disabled (*atributo calibre.customize.conversion.InputFormatPlugin*), 265
can_be_disabled (*atributo calibre.customize.conversion.OutputFormatPlugin*), 267
can_be_disabled (*atributo calibre.customize.InterfaceActionBase*), 285
can_be_disabled (*atributo calibre.customize.Plugin*), 257
can_be_disabled (*atributo calibre.customize.PreferencesPlugin*), 285
CAN_DO_DEVICE_DB_PLUGBOARD (*atributo calibre.devices.interface.DevicePlugin*), 269
can_get_multiple_covers (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
can_handle() (*método calibre.devices.interface.DevicePlugin*), 270
can_handle_windows() (*método calibre.devices.interface.DevicePlugin*), 270
can_handle_windows() (*método calibre.devices.usbms.device.Device*), 278
CAN_SET_METADATA (*atributo calibre.devices.interface.DevicePlugin*), 269
CAN_SET_METADATA (*atributo calibre.devices.usbms.driver.USBMS*), 280
canonicalize_internal_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
capabilities (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
card_prefix() (*método calibre.devices.interface.DevicePlugin*), 271
card_prefix() (*método calibre.devices.usbms.device.Device*), 278
CatalogPlugin (*classe em calibre.customize*), 261
category (*atributo calibre.customize.PreferencesPlugin*), 285
category_order (*atributo calibre.customize.PreferencesPlugin*), 285
center_navbar (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
change_font() (*no módulo calibre.ebooks.oeb.polish.fonts*), 368
changed_signal (*atributo calibre.gui2.preferences.ConfigWidgetBase*), 287
changed_signal (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 286
clean_downloaded_metadata() (*método calibre.ebooks.metadata.sources.base.Source*), 263
cleanup() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
CLI (*classe em calibre.devices.usbms.cli*), 279
cli_main() (*método calibre.customize.Plugin*), 258
cli_options (*atributo calibre.customize.CatalogPlugin*), 261
clone_browser() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42

close_editor() (*método calibre.gui2.tweak_book.boss.Boss*), 371
commit() (*método calibre.ebooks.oeb.polish.container.Container*), 362
commit() (*método calibre.gui2.preferences.ConfigWidgetBase*), 287
commit() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 286
commit_all_editors_to_container() (*método calibre.gui2.tweak_book.boss.Boss*), 371
commit_item() (*método calibre.ebooks.oeb.polish.container.Container*), 363
common_options (*atributo calibre.customize.conversion.InputFormatPlugin*), 266
common_options (*atributo calibre.customize.conversion.OutputFormatPlugin*), 267
compress_covers() (*método calibre.db.cache.Cache*), 353
compress_news_images (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 46
compress_news_images_auto_size (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
compress_news_images_max_size (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
config_help_message (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
config_widget (*atributo calibre.customize.PreferencesPlugin*), 285
config_widget() (*método calibre.customize.Plugin*), 257
config_widget() (*método calibre.ebooks.metadata.sources.base.Source*), 263
config_widget() (*método de classe calibre.devices.interface.DevicePlugin*), 273
ConfigWidgetBase (*classe em calibre.gui2.preferences*), 286
ConfigWidgetInterface (*classe em calibre.gui2.preferences*), 286
Container (*classe em calibre.ebooks.oeb.polish.container*), 362
contains(), **160**
conversion_options (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
convert() (*método calibre.customize.conversion.InputFormatPlugin*), 266
convert() (*método calibre.customize.conversion.OutputFormatPlugin*), 267
copy_cover_to() (*método calibre.db.cache.Cache*), 354
copy_format_to() (*método calibre.db.cache.Cache*), 354
core_usage (*atributo calibre.customize.conversion.InputFormatPlugin*), 266
cover() (*método calibre.db.cache.Cache*), 354
cover_margins (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
create_action() (*método calibre.gui2.tweak_book.plugin.Tool*), 370
create_inline_toc() (*no módulo calibre.ebooks.oeb.polish.toc*), 369
create_menu_action() (*método calibre.gui2.actions.InterfaceAction*), 283
create_widget() (*método calibre.customize.PreferencesPlugin*), 286
current_container (*propriedade calibre.gui2.tweak_book.plugin.Tool*), 369
currently_editing (*propriedade calibre.gui2.tweak_book.boss.Boss*), 371
custom_field_keys() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
customization_help() (*método calibre.customize.Plugin*), 258
customization_help() (*método calibre.ebooks.metadata.sources.base.Source*), 262

D

data_for_find_identical_books() (*método calibre.db.cache.Cache*), 354
data_for_has_book() (*método calibre.db.cache.Cache*), 354
debug_managed_device_detection() (*método calibre.devices.interface.DevicePlugin*), 270
deepcopy() (*método calibre.ebooks.metadata.book.base.Metadata*), 211
default_cover() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
delay (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
delete_annotations() (*método calibre.db.cache.Cache*), 354
delete_books() (*método calibre.devices.interface.DevicePlugin*), 273
delete_books() (*método calibre.devices.usbms.driver.USBMS*), 281
delete_custom_book_data() (*método calibre.db.cache.Cache*), 354

delete_trash_entry() (*método calibre.db.cache.Cache*), 354
description (*atributo calibre.customize.Plugin*), 256
description (*atributo calibre.customize.PreferencesPlugin*), 285
description (*atributo calibre.devices.usbms.driver.USBMS*), 279
description (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
description (*propriedade calibre.customize.conversion.OutputFormatPlugin*), 267
detect_managed_devices() (*método calibre.devices.interface.DevicePlugin*), 270
Device (*classe em calibre.devices.usbms.device*), 276
DevicePlugin (*classe em calibre.devices.interface*), 268
dirty() (*método calibre.ebooks.oeb.polish.container.Container*), 363
do_user_config() (*método calibre.customize.Plugin*), 257
dont_add_to (*atributo calibre.gui2.actions.InterfaceAction*), 282
dont_remove_from (*atributo calibre.gui2.actions.InterfaceAction*), 282
download() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42
download_cover() (*método calibre.ebooks.metadata.sources.base.Source*), 265
drop_event() (*método calibre.gui2.actions.InterfaceAction*), 283

E

ebook-convert opção de linha de comando
--add-alt-text-to-img, 331
--asciiize, 328
--author-sort, 333
--authors, 333
--base-font-size, 328
--book-producer, 333
--change-justification, 328
--chapter, 332
--chapter-mark, 332
--comments, 333
--cover, 333
--debug-pipeline, 334
--disable-dehyphenate, 330
--disable-delete-blank-paragraphs, 330
--disable-fix-indents, 330
--disable-font-rescaling, 328
--disable-format-scene-breaks, 330
--disable-italicize-common-cases, 330
--disable-markup-chapter-headings, 331
--disable-remove-fake-margins, 332
--disable-renumber-headings, 331
--disable-unwrap-lines, 331
--dont-split-on-page-breaks, 327
--duplicate-links-in-toc, 332
--embed-all-fonts, 328
--embed-font-family, 328
--enable-heuristics, 331
--epub-flatten, 327
--epub-inline-toc, 327
--epub-max-image-size, 327
--epub-toc-at-end, 327
--epub-version, 327
--expand-css, 328
--extra-css, 328
--extract-to, 327
--filter-css, 328
--flow-size, 327
--font-size-mapping, 329
--from-opf, 334
--help, 326
--html-unwrap-factor, 331
--input-encoding, 326
--input-profile, 326
--insert-blank-line, 329
--insert-blank-line-size, 329
--insert-metadata, 332
--isbn, 333
--keep-ligatures, 329
--language, 333
--level1-toc, 332
--level2-toc, 332
--level3-toc, 333
--line-height, 329
--linearize-tables, 329
--list-recipes, 326
--margin-bottom, 329
--margin-left, 329
--margin-right, 329
--margin-top, 329
--max-toc-links, 333
--minimum-line-height, 329
--no-chapters-in-toc, 333
--no-default-epub-cover, 327
--no-svg-cover, 327
--output-profile, 326
--page-breaks-before, 332
--prefer-metadata-cover, 332
--preserve-cover-aspect-ratio, 327
--pretty-print, 327
--pubdate, 333
--publisher, 333
--rating, 334
--read-metadata-from-opf, 334

--remove-first-image, 332
--remove-paragraph-spacing, 330
--remove-paragraph-spacing-indent-size, 330
--replace-scene-breaks, 331
--search-replace, 331
--series, 334
--series-index, 334
--smarten-punctuation, 330
--sr1-replace, 331
--sr1-search, 331
--sr2-replace, 331
--sr2-search, 331
--sr3-replace, 331
--sr3-search, 331
--start-reading-at, 332
--subset-embedded-fonts, 330
--tags, 334
--timestamp, 334
--title, 334
--title-sort, 334
--toc-filter, 333
--toc-threshold, 333
--toc-title, 328
--transform-css-rules, 330
--transform-html-rules, 330
--unsmarten-punctuation, 330
--use-auto-toc, 333
--verbose, 334
--version, 326
-d, 334
-h, 326
-m, 334
-v, 334
ebook-edit opção de linha de comando
--detach, 334
--help, 334
--select-text, 334
--version, 335
-h, 334
ebook-meta opção de linha de comando
--author-sort, 335
--authors, 335
--book-producer, 335
--category, 335
--comments, 335
--cover, 335
--date, 335
--disallow-rendered-cover, 335
--from-opf, 335
--get-cover, 335
--help, 336
--identifier, 336
--index, 336
--isbn, 336
--language, 336
--lrf-bookid, 336
--publisher, 336
--rating, 336
--series, 336
--tags, 336
--title, 336
--title-sort, 336
--to-opf, 336
--version, 336
-a, 335
-c, 335
-d, 335
-h, 336
-i, 336
-k, 335
-l, 336
-p, 336
-r, 336
-s, 336
-t, 336
ebook-polish opção de linha de comando
-H, 337
-U, 338
--add-soft-hyphens, 337
--compress-images, 337
--cover, 337
--download-external-resources, 337
--embed-fonts, 337
--help, 337
--jacket, 337
--opf, 337
--remove-jacket, 337
--remove-soft-hyphens, 337
--remove-unused-css, 337
--smarten-punctuation, 337
--subset-fonts, 337
--upgrade-book, 338
--verbose, 338
--version, 338
-c, 337
-d, 337
-e, 337
-f, 337
-h, 337
-i, 337
-j, 337
-o, 337
-p, 337
-u, 337
ebook-viewer opção de linha de comando
--continue, 338
--detach, 338

--force-reload, 338
--full-screen, 338
--fullscreen, 338
--help, 338
--new-instance, 338
--open-at, 338
--raise-window, 339
--version, 339
-f, 338
-h, 338
edit_file() (*método calibre.gui2.tweak_book.boss.Boss*), 371
eject() (*método calibre.devices.interface.DevicePlugin*), 271
eject() (*método calibre.devices.usbms.device.Device*), 279
embed_metadata() (*método calibre.db.cache.Cache*), 354
encoding (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
exists() (*método calibre.ebooks.oeb.polish.container.Container*), 363
expire_old_trash() (*método calibre.db.cache.Cache*), 354
export_note() (*método calibre.db.cache.Cache*), 354
expressão regular, **375**
Extensão de arquivos utilizada em e-readers da Sony, **375**
extra_css (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 47
extract_readable_article() (*método calibre.web.feeds.news.BasicNewsRecipe*), 42

F

fast_field_for() (*método calibre.db.cache.Cache*), 354
feeds (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48
fetch-ebook-metadata opção de linha de comando
-I, 339
--allowed-plugin, 339
--authors, 339
--cover, 339
--help, 339
--identifier, 339
--isbn, 339
--opf, 339
--timeout, 339
--title, 339
--verbose, 339
--version, 339
-a, 339
-c, 339
-d, 339
-h, 339
-i, 339
-o, 339
-p, 339
-t, 339
-v, 339
field_for() (*método calibre.db.cache.Cache*), 355
field_ids_for() (*método calibre.db.cache.Cache*), 355
field_supports_notes() (*método calibre.db.cache.Cache*), 355
file_type (*atributo calibre.customize.conversion.OutputFormatPlugin*), 267
file_types (*atributo calibre.customize.CatalogPlugin*), 261
file_types (*atributo calibre.customize.conversion.InputFormatPlugin*), 265
file_types (*atributo calibre.customize.FileTypePlugin*), 258
file_types (*atributo calibre.customize.MetadataReaderPlugin*), 260
file_types (*atributo calibre.customize.MetadataWriterPlugin*), 260
filename_callback() (*método calibre.devices.usbms.device.Device*), 279
filesize() (*método calibre.ebooks.oeb.polish.container.Container*), 363
FileTypePlugin (*classe em calibre.customize*), 258
filter_css() (*no módulo calibre.ebooks.oeb.polish.css*), 368
filter_regexps (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48
find_identical_books() (*método calibre.db.cache.Cache*), 355
fix_all_html() (*no módulo calibre.ebooks.oeb.polish.pretty*), 366
fix_html() (*no módulo calibre.ebooks.oeb.polish.pretty*), 366
Folhas de estilo em cascata., **375**
for_viewer (*atributo calibre.customize.conversion.InputFormatPlugin*), 266
format() (*método calibre.db.cache.Cache*), 355
format_abspath() (*método calibre.db.cache.Cache*), 355
format_added (*atributo calibre.db.cache.Cache.EventType*), 351
format_field() (*método calibre.ebooks.metadata.book.base.Metadata*), 213

format_hash() (*método calibre.db.cache.Cache*), 355
format_metadata() (*método calibre.db.cache.Cache*), 355
FORMATS (*atributo calibre.devices.interface.DevicePlugin*), 268
FORMATS (*atributo calibre.devices.usbms.driver.USBMS*), 280
formats() (*método calibre.db.cache.Cache*), 356
formats_removed (*atributo calibre.db.cache.Cache.EventType*), 351
free_space() (*método calibre.devices.interface.DevicePlugin*), 272
free_space() (*método calibre.devices.usbms.device.Device*), 278
from_files() (*no módulo calibre.ebooks.oeb.polish.toc*), 369
from_links() (*no módulo calibre.ebooks.oeb.polish.toc*), 369
from_xpaths() (*no módulo calibre.ebooks.oeb.polish.toc*), 369

G

generate_item() (*método calibre.ebooks.oeb.polish.container.Container*), 363
genesis() (*método calibre.gui2.actions.InterfaceAction*), 284
genesis() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 286
get_all_items_that_have_notes() (*método calibre.db.cache.Cache*), 356
get_all_link_maps_for_book() (*método calibre.db.cache.Cache*), 356
get_all_standard_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
get_all_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
get_annotations() (*método calibre.devices.usbms.device.Device*), 279
get_article_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 43
get_author_tokens() (*método calibre.ebooks.metadata.sources.base.Source*), 263
get_book_url() (*método calibre.ebooks.metadata.sources.base.Source*), 263
get_book_url_name() (*método calibre.ebooks.metadata.sources.base.Source*), 263
get_book_urls() (*método calibre.ebooks.metadata.sources.base.Source*), 263
get_browser() (*método calibre.web.feeds.news.BasicNewsRecipe*), 43
get_cached_cover_url() (*método calibre.ebooks.metadata.sources.base.Source*), 264
get_categories() (*método calibre.db.cache.Cache*), 356
get_collections() (*método calibre.devices.interface.BookList*), 276
get_cover_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 43
get_custom_book_data() (*método calibre.db.cache.Cache*), 356
get_device_information() (*método calibre.devices.interface.DevicePlugin*), 271
get_device_information() (*método calibre.devices.usbms.driver.USBMS*), 280
get_device_uid() (*método calibre.devices.interface.DevicePlugin*), 274
get_driveinfo() (*método calibre.devices.interface.DevicePlugin*), 271
get_extra_css() (*método calibre.web.feeds.news.BasicNewsRecipe*), 43
get_feeds() (*método calibre.web.feeds.news.BasicNewsRecipe*), 43
get_file() (*método calibre.devices.interface.DevicePlugin*), 273
get_file_path_for_processing() (*método calibre.ebooks.oeb.polish.container.Container*), 363
get_id_map() (*método calibre.db.cache.Cache*), 356
get_identifiers() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
get_ids_for_custom_book_data() (*método calibre.db.cache.Cache*), 356
get_images() (*método calibre.customize.conversion.InputFormatPlugin*), 266
get_item_id() (*método calibre.db.cache.Cache*), 356
get_item_ids() (*método calibre.db.cache.Cache*), 356
get_item_name() (*método calibre.db.cache.Cache*), 356
get_item_name_map() (*método calibre.db.cache.Cache*), 356
get_link_map() (*método calibre.db.cache.Cache*), 356
get_masthead_title() (*método calibre.web.feeds.news.BasicNewsRecipe*), 43
get_masthead_url() (*método calibre.web.feeds.news.BasicNewsRecipe*), 43
get_metadata() (*método calibre.customize.MetadataReaderPlugin*), 260

get_metadata() (*método calibre.db.cache.Cache*), 357
get_next_series_num_for() (*método calibre.db.cache.Cache*), 357
get_notes_resource() (*método calibre.db.cache.Cache*), 357
get_obfuscated_article() (*método calibre.web.feeds.news.BasicNewsRecipe*), 44
get_open_popup_message() (*método de classe calibre.devices.interface.DevicePlugin*), 270
get_option() (*método calibre.devices.interface.DevicePlugin*), 275
get_proxy_metadata() (*método calibre.db.cache.Cache*), 357
get_recommended_folders() (*no módulo calibre.ebooks.oeb.polish.replace*), 366
get_standard_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
get_title_tokens() (*método calibre.ebooks.metadata.sources.base.Source*), 263
get_url_specific_delay() (*método calibre.web.feeds.news.BasicNewsRecipe*), 44
get_usage_count_by_id() (*método calibre.db.cache.Cache*), 357
get_user_blacklisted_devices() (*método calibre.devices.interface.DevicePlugin*), 274
get_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
gui (*propriedade calibre.gui2.tweak_book.plugin.Tool*), 369
gui_category (*atributo calibre.customize.PreferencesPlugin*), 285
gui_configuration_widget() (*método calibre.customize.conversion.InputFormatPlugin*), 267
gui_configuration_widget() (*método calibre.customize.conversion.OutputFormatPlugin*), 268
gui_layout_complete() (*método calibre.gui2.actions.InterfaceAction*), 284
gui_name (*atributo calibre.customize.PreferencesPlugin*), 285
guide_type_map (*propriedade calibre.ebooks.oeb.polish.container.Container*), 363

H

handle_gzip (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48
has_book() (*método calibre.db.cache.Cache*), 357
has_format() (*método calibre.db.cache.Cache*), 357
has_html_comments (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
has_id() (*método calibre.db.cache.Cache*), 357
has_name() (*método calibre.ebooks.oeb.polish.container.Container*), 363
href_to_name() (*método calibre.ebooks.oeb.polish.container.Container*), 363

I

icon (*atributo calibre.customize.PreferencesPlugin*), 285
icon (*atributo calibre.devices.interface.DevicePlugin*), 269
id_from_url() (*método calibre.ebooks.metadata.sources.base.Source*), 264
identify() (*método calibre.ebooks.metadata.sources.base.Source*), 264
identify_results_keygen() (*método calibre.ebooks.metadata.sources.base.Source*), 264
ignore_connected_device() (*método calibre.devices.interface.DevicePlugin*), 274
ignore_duplicate_articles (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48
ignore_ssl_errors (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
image_url_processor() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 41
import_note() (*método calibre.db.cache.Cache*), 357
index_to_soup() (*método calibre.web.feeds.news.BasicNewsRecipe*), 44
indexing_progress_changed (*atributo calibre.db.cache.Cache.EventType*), 352
init() (*método calibre.db.cache.Cache*), 357
initial_tab_changed() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 286
initialization_complete() (*método calibre.gui2.actions.InterfaceAction*), 284
initialize() (*método calibre.customize.CatalogPlugin*), 261
initialize() (*método calibre.customize.Plugin*), 257
initialize() (*método calibre.gui2.preferences.ConfigWidgetBase*), 287
initialize() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 286
InputFormatPlugin (*classe em calibre.customize.conversion*), 265

insert_into_xml() (*método calibre.ebooks.oeb.polish.container.Container*), 363
installation_type (*atributo calibre.customize.Plugin*), 257
Interface de programação de aplicativos., **375**
InterfaceAction (*classe em calibre.gui2.actions*), 281
InterfaceActionBase (*classe em calibre.customize*), 284
InternalMetadataCompareKeyGen (*classe em calibre.ebooks.metadata.sources.base*), 265
is_configured() (*método calibre.ebooks.metadata.sources.base.Source*), 262
is_dir (*atributo calibre.ebooks.oeb.polish.container.Container*), 363
is_dynamically_controllable() (*método calibre.devices.interface.DevicePlugin*), 274
is_image_collection (*atributo calibre.customize.conversion.InputFormatPlugin*), 265
is_link_wanted() (*método calibre.web.feeds.news.BasicNewsRecipe*), 44
is_null() (*método calibre.ebooks.metadata.book.base.Metadata*), 211
is_running() (*método calibre.devices.interface.DevicePlugin*), 275
is_usb_connected() (*método calibre.devices.interface.DevicePlugin*), 270
items_removed (*atributo calibre.db.cache.Cache.EventType*), 352
items_renamed (*atributo calibre.db.cache.Cache.EventType*), 352
items_with_notes_in_book() (*método calibre.db.cache.Cache*), 357
iterlinks() (*método calibre.ebooks.oeb.polish.container.Container*), 363

K

keep_only_tags (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48

L

language (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48
library_about_to_change() (*método calibre.gui2.actions.InterfaceAction*), 284
library_changed() (*método calibre.gui2.actions.InterfaceAction*), 284
Linguagem de marcação de hipertexto., **375**
link_for() (*método calibre.db.cache.Cache*), 357
links_changed (*atributo calibre.db.cache.Cache.EventType*), 352
list_extra_files() (*método calibre.db.cache.Cache*), 357
load_actual_plugin() (*método calibre.customize.InterfaceActionBase*), 285
load_resources() (*método calibre.customize.Plugin*), 257
load_resources() (*método calibre.gui2.actions.InterfaceAction*), 283
Localizador padrão de recursos., **375**
location_selected() (*método calibre.gui2.actions.InterfaceAction*), 284
lrf2lrs opção de linha de comando
  --dont-output-resources, 340
  --help, 340
  --output, 340
  --verbose, 340
  --version, 340
  -h, 340
  -o, 340
lrfviewer opção de linha de comando
  --disable-hyphenation, 340
  --help, 340
  --profile, 340
  --verbose, 340
  --version, 340
  --visual-debug, 340
  --white-background, 340
  -h, 340
lrs2lrf opção de linha de comando
  --help, 341
  --lrs, 341
  --output, 341
  --verbose, 341
  --version, 341
  -h, 341
  -o, 341

M

make_name_unique() (*método calibre.ebooks.oeb.polish.container.Container*), 363
MANAGES_DEVICE_PRESENCE (*atributo calibre.devices.interface.DevicePlugin*), 269
manifest_has_name() (*método calibre.ebooks.oeb.polish.container.Container*), 363
manifest_id_map (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
manifest_items_of_type() (*método calibre.ebooks.oeb.polish.container.Container*), 364

manifest_items_with_property() (*método calibre.ebooks.oeb.polish.container.Container*), 364
manifest_type_map (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
mark_as_cover() (*no módulo calibre.ebooks.oeb.polish.cover*), 368
mark_as_titlepage() (*no módulo calibre.ebooks.oeb.polish.cover*), 368
masthead_url (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48
match_regexps (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 48
max_articles_per_feed (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49
MAX_PATH_LEN (*atributo calibre.devices.usbms.device.Device*), 277
merge() (*no módulo calibre.ebooks.oeb.polish.split*), 367
merge_annotations_for_book() (*método calibre.db.cache.Cache*), 358
merge_extra_files() (*método calibre.db.cache.Cache*), 358
Metadata (*classe em calibre.ebooks.metadata.book.base*), 211
metadata_changed (*atributo calibre.db.cache.Cache.EventType*), 352
metadata_for_field() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
MetadataReaderPlugin (*classe em calibre.customize*), 260
MetadataWriterPlugin (*classe em calibre.customize*), 260
mi (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
minimum_calibre_version (*atributo calibre.customize.Plugin*), 257
model_metadata() (*método de classe calibre.devices.interface.DevicePlugin*), 270
model_metadata() (*método de classe calibre.devices.usbms.device.Device*), 277
module
    calibre.customize, 256
    calibre.customize.conversion, 265
    calibre.db.cache, 351
    calibre.devices.interface, 268
    calibre.ebooks.metadata.book.base, 211
    calibre.ebooks.metadata.sources.base, 262
    calibre.ebooks.oeb.polish.container, 361
    calibre.ebooks.oeb.polish.cover, 367
    calibre.ebooks.oeb.polish.css, 368
    calibre.ebooks.oeb.polish.jacket, 367
    calibre.ebooks.oeb.polish.pretty, 366
    calibre.ebooks.oeb.polish.replace, 366
    calibre.ebooks.oeb.polish.split, 367
    calibre.ebooks.oeb.polish.toc, 369
    calibre.gui2.tweak_book.boss, 370
    calibre.utils.formatter_functions, 184
    calibre.web.feeds.news, 41
move_book_from_trash() (*método calibre.db.cache.Cache*), 358
move_format_from_trash() (*método calibre.db.cache.Cache*), 358
multisort() (*método calibre.db.cache.Cache*), 358
multisplit() (*no módulo calibre.ebooks.oeb.polish.split*), 367

N

name (*atributo calibre.customize.Plugin*), 256
name (*atributo calibre.gui2.actions.InterfaceAction*), 282
name (*atributo calibre.gui2.tweak_book.plugin.Tool*), 369
name(), **160**
name_order (*atributo calibre.customize.PreferencesPlugin*), 285
name_to_abspath() (*método calibre.ebooks.oeb.polish.container.Container*), 364
name_to_href() (*método calibre.ebooks.oeb.polish.container.Container*), 364
names_that_must_not_be_changed (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
names_that_must_not_be_removed (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
names_that_need_not_be_manifested (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
needs_subscription (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49
NEWS_IN_FOLDER (*atributo calibre.devices.usbms.device.Device*), 277
no_stylesheets (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49
normalize_path() (*método de classe calibre.devices.usbms.driver.USBMS*), 281
notes_changed (*atributo calibre.db.cache.Cache.EventType*), 352
notes_data_for() (*método calibre.db.cache.Cache*), 358
notes_for() (*método calibre.db.cache.Cache*), 358
notes_resources_used_by() (*método calibre.db.cache.Cache*), 358

NUKE_COMMENTS (*atributo calibre.devices.interface.DevicePlugin*), 269

O

oldest_article (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49
on_import (*atributo calibre.customize.FileTypePlugin*), 258
on_postconvert (*atributo calibre.customize.FileTypePlugin*), 258
on_postdelete (*atributo calibre.customize.FileTypePlugin*), 258
on_postimport (*atributo calibre.customize.FileTypePlugin*), 258
on_postprocess (*atributo calibre.customize.FileTypePlugin*), 259
on_preprocess (*atributo calibre.customize.FileTypePlugin*), 259
open() (*método calibre.devices.interface.DevicePlugin*), 271
open() (*método calibre.devices.usbms.device.Device*), 278
open() (*método calibre.ebooks.oeb.polish.container.Container*), 364
open_book() (*método calibre.gui2.tweak_book.boss.Boss*), 371
OPEN_FEEDBACK_MESSAGE (*atributo calibre.devices.interface.DevicePlugin*), 269
opf (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
opf_get_or_create() (*método calibre.ebooks.oeb.polish.container.Container*), 364
opf_version (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
opf_version_parsed (*propriedade calibre.ebooks.oeb.polish.container.Container*), 364
opf_xpath() (*método calibre.ebooks.oeb.polish.container.Container*), 364
options (*atributo calibre.customize.conversion.InputFormatPlugin*), 266
options (*atributo calibre.customize.conversion.OutputFormatPlugin*), 267
options (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
opção de linha de comando
  --help, 313
  --library-path, 313
  --password, 313
  --timeout, 313
  --username, 313
  --version, 313
  --with-library, 313
  -h, 313
OSX_MAIN_MEM_VOL_PAT (*atributo calibre.devices.usbms.device.Device*), 277
output_encoding (*atributo calibre.customize.conversion.InputFormatPlugin*), 266
OutputFormatPlugin (*classe em calibre.customize.conversion*), 267

P

parse_feeds() (*método calibre.web.feeds.news.BasicNewsRecipe*), 44
parse_index() (*método calibre.web.feeds.news.BasicNewsRecipe*), 44
parsed() (*método calibre.ebooks.oeb.polish.container.Container*), 365
path_sep (*atributo calibre.devices.interface.DevicePlugin*), 269
Plugin (*classe em calibre.customize*), 256
populate_article_metadata() (*método calibre.web.feeds.news.BasicNewsRecipe*), 45
popup_type (*atributo calibre.gui2.actions.InterfaceAction*), 282
post_yank_cleanup() (*método calibre.devices.interface.DevicePlugin*), 271
post_yank_cleanup() (*método calibre.devices.usbms.device.Device*), 279
postadd() (*método calibre.customize.FileTypePlugin*), 260
postconvert() (*método calibre.customize.FileTypePlugin*), 259
postdelete() (*método calibre.customize.FileTypePlugin*), 259
postimport() (*método calibre.customize.FileTypePlugin*), 259
postprocess_book() (*método calibre.customize.conversion.InputFormatPlugin*), 266
postprocess_book() (*método calibre.web.feeds.news.BasicNewsRecipe*), 45
postprocess_html() (*método calibre.web.feeds.news.BasicNewsRecipe*), 45
pref() (*método calibre.db.cache.Cache*), 358
prefer_results_with_isbn (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
PreferencesPlugin (*classe em calibre.customize*), 285
prepare_addable_books() (*método calibre.devices.interface.DevicePlugin*), 274

preprocess_html() (*método calibre.web.feeds.news.BasicNewsRecipe*), 45
preprocess_image() (*método calibre.web.feeds.news.BasicNewsRecipe*), 45
preprocess_raw_html() (*método calibre.web.feeds.news.BasicNewsRecipe*), 45
preprocess_regexps (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49
pretty_all() (*no módulo calibre.ebooks.oeb.polish.pretty*), 367
pretty_css() (*no módulo calibre.ebooks.oeb.polish.pretty*), 366
pretty_html() (*no módulo calibre.ebooks.oeb.polish.pretty*), 366
pretty_xml() (*no módulo calibre.ebooks.oeb.polish.pretty*), 366
print_version() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 41
priority (*atributo calibre.customize.Plugin*), 257
priority (*atributo calibre.gui2.actions.InterfaceAction*), 282
PRODUCT_ID (*atributo calibre.devices.interface.DevicePlugin*), 268
PRODUCT_ID (*atributo calibre.devices.usbms.device.Device*), 277
publication_date() (*método calibre.web.feeds.news.BasicNewsRecipe*), 46
publication_type (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49

R

raw_data() (*método calibre.ebooks.oeb.polish.container.Container*), 365
re:test(), **160**
read_backup() (*método calibre.db.cache.Cache*), 358
Receita, **375**
recipe_disabled (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49
recipe_specific_options (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 49
recommendations (*atributo calibre.customize.conversion.InputFormatPlugin*), 266
recommendations (*atributo calibre.customize.conversion.OutputFormatPlugin*), 267
recursions (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
refresh_gui() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 286
register() (*método calibre.gui2.preferences.ConfigWidgetBase*), 287
register_shortcut() (*método calibre.gui2.tweak_book.plugin.Tool*), 369
relpath() (*método calibre.ebooks.oeb.polish.container.Container*), 365
remove_attributes (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_book() (*método calibre.devices.interface.BookList*), 276
remove_books() (*método calibre.db.cache.Cache*), 358
remove_books_from_metadata() (*método calibre.devices.usbms.driver.USBMS*), 281
remove_books_from_metadata() (*método de classe calibre.devices.interface.DevicePlugin*), 273
remove_empty_feeds (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_extra_files() (*método calibre.db.cache.Cache*), 358
remove_formats() (*método calibre.db.cache.Cache*), 358
remove_from_spine() (*método calibre.ebooks.oeb.polish.container.Container*), 365
remove_from_xml() (*método calibre.ebooks.oeb.polish.container.Container*), 365
remove_item() (*método calibre.ebooks.oeb.polish.container.Container*), 365
remove_items() (*método calibre.db.cache.Cache*), 359
remove_jacket() (*no módulo calibre.ebooks.oeb.polish.jacket*), 367
remove_javascript (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_stale_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
remove_tags (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_tags_after (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_tags_before (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
remove_unused_css() (*no módulo calibre.ebooks.oeb.polish.css*), 368
rename() (*método calibre.ebooks.oeb.polish.container.Container*), 365
rename_extra_files() (*método calibre.db.cache.Cache*), 359
rename_files() (*no módulo calibre.ebooks.oeb.polish.replace*), 366
rename_items() (*método calibre.db.cache.Cache*), 359
replace() (*método cali-*

bre.ebooks.oeb.polish.container.Container), 365
replace_links() (*método calibre.ebooks.oeb.polish.container.Container*), 365
replace_links() (*no módulo calibre.ebooks.oeb.polish.replace*), 366
requires_version (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
reset() (*método calibre.devices.interface.DevicePlugin*), 270
reset() (*método calibre.devices.usbms.device.Device*), 277
resolve_internal_links (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 50
restart_critical (*atributo calibre.gui2.preferences.ConfigWidgetBase*), 287
restart_critical (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 286
restore_book() (*método calibre.db.cache.Cache*), 359
restore_defaults() (*método calibre.gui2.preferences.ConfigWidgetBase*), 287
restore_defaults() (*método calibre.gui2.preferences.ConfigWidgetInterface*), 286
restore_defaults_desc (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 286
restore_original_format() (*método calibre.db.cache.Cache*), 359
reverse_article_order (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
rewind_savepoint() (*método calibre.gui2.tweak_book.boss.Boss*), 371
RSS, **375**
run() (*método calibre.customize.CatalogPlugin*), 261
run() (*método calibre.customize.FileTypePlugin*), 259

S

safe_read_lock (*propriedade calibre.db.cache.Cache*), 359
sanitize_callback() (*método calibre.devices.usbms.device.Device*), 279
sanitize_path_components() (*método calibre.devices.usbms.device.Device*), 279
save_book() (*método calibre.gui2.tweak_book.boss.Boss*), 371
save_original_format() (*método calibre.db.cache.Cache*), 359
save_settings() (*método calibre.customize.Plugin*), 257
save_settings() (*método calibre.ebooks.metadata.sources.base.Source*), 263
save_settings() (*método de classe calibre.devices.interface.DevicePlugin*), 273
scale_news_images (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
scale_news_images_to_device (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
search() (*método calibre.db.cache.Cache*), 359
search_annotations() (*método calibre.db.cache.Cache*), 360
search_notes() (*método calibre.db.cache.Cache*), 360
serialize_item() (*método calibre.ebooks.oeb.polish.container.Container*), 365
set_all_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
set_annotations_for_book() (*método calibre.db.cache.Cache*), 360
set_conversion_options() (*método calibre.db.cache.Cache*), 360
set_cover() (*método calibre.db.cache.Cache*), 360
set_cover() (*no módulo calibre.ebooks.oeb.polish.cover*), 367
set_driveinfo_name() (*método calibre.devices.interface.DevicePlugin*), 273
set_driveinfo_name() (*método calibre.devices.usbms.driver.USBMS*), 280
set_field() (*método calibre.db.cache.Cache*), 360
set_identifier() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
set_identifiers() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
set_library_info() (*método calibre.devices.interface.DevicePlugin*), 274
set_link_map() (*método calibre.db.cache.Cache*), 360
set_metadata() (*método calibre.customize.MetadataWriterPlugin*), 261
set_metadata() (*método calibre.db.cache.Cache*), 360
set_modified() (*método calibre.gui2.tweak_book.boss.Boss*), 371
set_notes_for() (*método calibre.db.cache.Cache*), 360
set_option() (*método calibre.devices.interface.DevicePlugin*), 275
set_plugboards() (*método calibre.devices.interface.DevicePlugin*), 273
set_pref() (*método calibre.db.cache.Cache*), 361
set_progress_reporter() (*método calibre.devices.interface.DevicePlugin*), 271
set_progress_reporter() (*método calibre.devices.usbms.device.Device*), 278
set_spine() (*método cali-*

bre.ebooks.oeb.polish.container.Container), 365
set_user_blacklisted_devices() (*método calibre.devices.interface.DevicePlugin*), 274
set_user_metadata() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
settings() (*método de classe calibre.devices.interface.DevicePlugin*), 273
show_current_diff() (*método calibre.gui2.tweak_book.boss.Boss*), 371
show_editor() (*método calibre.gui2.tweak_book.boss.Boss*), 371
shutdown() (*método calibre.devices.interface.DevicePlugin*), 274
shutting_down() (*método calibre.gui2.actions.InterfaceAction*), 284
simultaneous_downloads (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
skip_ad_pages() (*método calibre.web.feeds.news.BasicNewsRecipe*), 46
SLOW_DRIVEINFO (*atributo calibre.devices.interface.DevicePlugin*), 269
smart_update() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
sort_index_by() (*método calibre.web.feeds.news.BasicNewsRecipe*), 46
Source (*classe em calibre.ebooks.metadata.sources.base*), 262
specialize() (*método calibre.customize.conversion.InputFormatPlugin*), 266
specialize_css_for_output() (*método calibre.customize.conversion.OutputFormatPlugin*), 268
specialize_global_preferences() (*método calibre.devices.interface.DevicePlugin*), 274
specialize_options() (*método calibre.customize.conversion.OutputFormatPlugin*), 268
spine_items (*propriedade calibre.ebooks.oeb.polish.container.Container*), 365
spine_iter (*propriedade calibre.ebooks.oeb.polish.container.Container*), 366
spine_names (*propriedade calibre.ebooks.oeb.polish.container.Container*), 366
split() (*no módulo calibre.ebooks.oeb.polish.split*), 367
split_if_is_multiple_composite() (*método calibre.db.cache.Cache*), 361
split_jobs() (*método calibre.ebooks.metadata.sources.base.Source*), 263
standard_field_keys() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
STANDARD_METADATA_FIELDS (*no módulo calibre.ebooks.metadata.book.base*), 213
start_plugin() (*método calibre.devices.interface.DevicePlugin*), 274
startup() (*método calibre.devices.interface.DevicePlugin*), 274
stop_plugin() (*método calibre.devices.interface.DevicePlugin*), 274
summary_length (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
supported_platforms (*atributo calibre.customize.conversion.InputFormatPlugin*), 265
supported_platforms (*atributo calibre.customize.conversion.OutputFormatPlugin*), 267
supported_platforms (*atributo calibre.customize.InterfaceActionBase*), 284
supported_platforms (*atributo calibre.customize.MetadataReaderPlugin*), 260
supported_platforms (*atributo calibre.customize.MetadataWriterPlugin*), 260
supported_platforms (*atributo calibre.customize.Plugin*), 256
supported_platforms (*atributo calibre.customize.PreferencesPlugin*), 285
supported_platforms (*atributo calibre.devices.usbms.driver.USBMS*), 279
supported_platforms (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
supports_collections() (*método calibre.devices.interface.BookList*), 276
supports_gzip_transfer_encoding (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
supports_restoring_to_defaults (*atributo calibre.gui2.preferences.ConfigWidgetBase*), 287
supports_restoring_to_defaults (*atributo calibre.gui2.preferences.ConfigWidgetInterface*), 286
sync_booklists() (*método calibre.devices.interface.DevicePlugin*), 273
sync_booklists() (*método calibre.devices.usbms.driver.USBMS*), 281
sync_preview_to_editor() (*método calibre.gui2.tweak_book.boss.Boss*), 372
synchronize_with_db() (*método calibre.devices.interface.DevicePlugin*), 275

T

tag_browser_context_action() (*método calibre.gui2.actions.InterfaceAction*), 284
tag_to_string() (*método de classe calibre.web.feeds.news.BasicNewsRecipe*), 42
tags_older_than() (*método calibre.db.cache.Cache*), 361
template_css (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
template_to_attribute() (*método calibre.ebooks.metadata.book.base.Metadata*), 212
temporary_file() (*método calibre.customize.Plugin*), 258
test_fields() (*método calibre.ebooks.metadata.sources.base.Source*), 263
THUMBNAIL_COMPRESSION_QUALITY (*atributo calibre.devices.interface.DevicePlugin*), 268
THUMBNAIL_HEIGHT (*atributo calibre.devices.interface.DevicePlugin*), 268
timefmt (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
timeout (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
title (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
to_html() (*método calibre.ebooks.metadata.book.base.Metadata*), 213
Tool (*classe em calibre.gui2.tweak_book.plugin*), 369
toolbar_button_popup_mode (*atributo calibre.gui2.tweak_book.plugin.Tool*), 369
total_space() (*método calibre.devices.interface.DevicePlugin*), 271
total_space() (*método calibre.devices.usbms.device.Device*), 278
touched_fields (*atributo calibre.ebooks.metadata.sources.base.Source*), 262
type (*atributo calibre.customize.CatalogPlugin*), 261
type (*atributo calibre.customize.conversion.InputFormatPlugin*), 265
type (*atributo calibre.customize.conversion.OutputFormatPlugin*), 267
type (*atributo calibre.customize.FileTypePlugin*), 259
type (*atributo calibre.customize.InterfaceActionBase*), 285
type (*atributo calibre.customize.MetadataReaderPlugin*), 260
type (*atributo calibre.customize.MetadataWriterPlugin*), 261
type (*atributo calibre.customize.Plugin*), 257
type (*atributo calibre.customize.PreferencesPlugin*), 285
type (*atributo calibre.devices.interface.DevicePlugin*), 268
type (*atributo calibre.ebooks.metadata.sources.base.Source*), 262

U

unretire_note_for() (*método calibre.db.cache.Cache*), 361
update_annotations() (*método calibre.db.cache.Cache*), 361
upload_books() (*método calibre.devices.interface.DevicePlugin*), 272
upload_books() (*método calibre.devices.usbms.driver.USBMS*), 280
upload_cover() (*método calibre.devices.usbms.driver.USBMS*), 280
USBMS (*classe em calibre.devices.usbms.driver*), 279
use_embedded_content (*atributo calibre.web.feeds.news.BasicNewsRecipe*), 51
user_categories_for_books() (*método calibre.db.cache.Cache*), 361
user_feedback_after_callback (*atributo calibre.devices.interface.DevicePlugin*), 269
UserAnnotation (*atributo calibre.devices.interface.DevicePlugin*), 269

V

VENDOR_ID (*atributo calibre.devices.interface.DevicePlugin*), 268
VENDOR_ID (*atributo calibre.devices.usbms.device.Device*), 277
version (*atributo calibre.customize.MetadataReaderPlugin*), 260
version (*atributo calibre.customize.MetadataWriterPlugin*), 261
version (*atributo calibre.customize.Plugin*), 256
VIRTUAL_BOOK_EXTENSION_MESSAGE (*atributo calibre.devices.interface.DevicePlugin*), 269
VIRTUAL_BOOK_EXTENSIONS (*atributo calibre.devices.interface.DevicePlugin*), 269

W

WANTS_UPDATED_THUMBNAILS (*atributo calibre.devices.interface.DevicePlugin*), 268
web2disk opção de linha de comando
--base-dir, 341
--delay, 341
--dont-download-stylesheets, 341
--encoding, 341
--filter-regexp, 341
--help, 341
--match-regexp, 341
--max-files, 342
--max-recursions, 342
--timeout, 342
--verbose, 342

`--version`, 342
`-d`, 341
`-h`, 341
`-n`, 342
`-r`, 342
`-t`, 342
`WINDOWS_CARD_A_MEM` (*atributo calibre.devices.usbms.device.Device*), 277
`WINDOWS_CARD_B_MEM` (*atributo calibre.devices.usbms.device.Device*), 277
`WINDOWS_MAIN_MEM` (*atributo calibre.devices.usbms.device.Device*), 277
`windows_sort_drives()` (*método calibre.devices.usbms.device.Device*), 278