



Manuale utente di calibre

Release 6.27.0

Kovid Goyal

settembre 22, 2023

1	L'interfaccia grafica	3
2	Aggiunta del tuo sito di notizie preferito	29
3	Il Lettore E-book	47
4	Conversione E-book	53
5	Modifica degli e-book	71
6	Il Server dei Contenuti di Calibre	105
7	Comparazione degli e-book	113
8	Modifica dei metadati degli e-book	117
9	Domande frequenti	121
10	Guide	145
11	Lo schema degli URL calibre://	243
12	Personalizzare calibre	247
13	Interfaccia a riga di comando	295
14	Creazione di un ambiente di sviluppo di calibre	337
15	Gestione dei Diritti Digitali (DRM)	365
16	Glossario	369
	Indice del modulo Python	371
	Indice	373

calibre è un gestore di raccolte di libri in formato digitale (e-book). Può visualizzarli, convertirli e catalogarli in quasi tutti i principali formati disponibili. Può anche comunicare con svariati lettori di ebook e connettersi ad Internet per ottenere i metadati dei libri. Tra le sue funzionalità, c'è anche la possibilità di scaricare quotidiani e convertirli in ebook per una lettura più comoda. Funziona su Linux, su Windows e OS X.

Hai appena avviato calibre. Quali sono i primi passi? Prima che calibre possa gestire i tuoi libri, è necessario che inizi a conoscerli. Trascina e rilascia qualche file di ebook in calibre, o fai clic sul pulsante «Aggiungi libri» e cerca gli ebook che vuoi caricare. Una volta aggiunti i libri, saranno visibili nella pagina principale con un risultato simile a questo:

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	★★★

Una volta ammirata a tuo piacimento la lista dei libri che hai appena aggiunto, probabilmente vorrai leggerne uno. Per fare ciò dovrai convertire il libro in un formato che il tuo lettore comprenda. Quando avvii calibre per la prima volta, il *Procedura guidata di benvenuto* si avvia e ti permette di impostare calibre per il tuo dispositivo di lettura. La conversione è una passeggiata. Seleziona il libro che vuoi convertire e fai clic sul pulsante «Converti libri». Ignora tutte le opzioni per il momento e fai clic su «OK». La piccola icona nell'angolo in basso a destra inizierà a girare. Una volta che ha finito di girare, il tuo libro convertito è pronto. Fai clic sul pulsante «Leggi» per leggere il libro.

Se vuoi leggere il libro sul tuo lettore, connetti il lettore al computer, aspetta che calibre lo riconosca (10-20 secondi) e poi fai clic sul pulsante «Invia al dispositivo». Una volta che l'icona smetterà di ruotare, disconnetti il lettore e inizia a leggere! Se non lo hai fatto precedentemente, calibre convertirà automaticamente il libro in un formato che il tuo lettore possa leggere.

Per avvicinarti a un utilizzo più avanzato, dai un'occhiata all'*L'interfaccia grafica* (pagina 3). Per ancora più potenza e versatilità, studia l'*Interfaccia a riga di comando* (pagina 295). Troverai utile anche l'elenco delle *Domande frequenti* (pagina 121).

Se hai ulteriori domande, o desideri discutere di calibre con altri utenti o chiedere aiuto su cose specifiche, sono disponibili [forum e altre risorse di aiuto](https://calibre-ebook.com/it/help)¹.

Sezioni

¹ <https://calibre-ebook.com/it/help>

L'interfaccia grafica

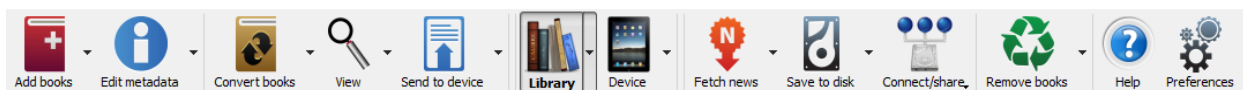
L'Interfaccia Grafica (*GUI*) dà accesso a tutte le funzionalità di gestione della biblioteca e di conversione degli e-book nei vari formati. Il primo passo per utilizzare calibre prevede l'aggiunta di libri alla biblioteca dal tuo disco rigido. calibre cercherà di reperire automaticamente i metadati dai libri e di aggiungerli al proprio database interno. Una volta nel database potrai effettuare alcune *Azioni* (pagina 4) su di essi, tra cui la conversione da un formato all'altro, il trasferimento su un dispositivo di lettura, la visualizzazione sul tuo computer e la modifica dei metadati. Quest'ultima azione consente, tra gli altri, la modifica di copertina, descrizione e tag. Nota che calibre crea una copia dei file che vi sono aggiunti. I file originali non vengono modificati.

L'interfaccia è divisa in diverse sezioni:

- *Azioni* (pagina 4)
- *Preferenze* (pagina 10)
- *Cataloghi* (pagina 11)
- *Cerca e riordina* (pagina 12)
- *L'interfaccia di ricerca* (pagina 12)
- *Salvare le ricerche* (pagina 18)
- *Ricerca nei contenuti di tutti i libri* (pagina 18)
- *Biblioteche virtuali* (pagina 19)
- *Marcatatura temporanea dei libri* (pagina 19)
- *Estrazione dei metadati dai nomi dei file* (pagina 19)
- *Dettagli del libro* (pagina 20)
- *Navigatore dei Tag* (pagina 22)
- *Griglia delle copertine* (pagina 24)
- *Navigatore delle copertine* (pagina 25)

- [Quickview](#) (pagina 25)
- [Lavori](#) (pagina 26)
- [Scorciatoie da tastiera](#) (pagina 26)

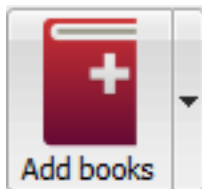
1.1 Azioni



La barra delle azioni fornisce comode scorciatoie alle funzioni più usate. Se fai clic con il tasto destro sui pulsanti, puoi eseguire varianti dell'azione predefinita. Tieni presente che la barra delle azioni apparirà leggermente diversa se hai un lettore di e-book collegato al tuo computer.

- [Aggiungi libri](#) (pagina 4)
- [Modifica metadati](#) (pagina 5)
- [Converti libri](#) (pagina 6)
- [Leggi](#) (pagina 6)
- [Invia al dispositivo](#) (pagina 6)
- [Scarica notizie](#) (pagina 7)
- [Biblioteca](#) (pagina 7)
- [Dispositivo](#) (pagina 8)
- [Salva su disco](#) (pagina 9)
- [Connetti/condividi](#) (pagina 9)
- [Rimuovi libri](#) (pagina 10)

1.1.1 Aggiungi libri



L'azione *Aggiungi libri* ha sette varianti accessibili facendo clic con il tasto destro sul pulsante.

1. **Add books from a single folder:** Opens a file chooser dialog and allows you to specify which books in a folder should be added. This action is *context sensitive*, i.e. it depends on which [catalog](#) (pagina 11) you have selected. If you have selected the *Library*, books will be added to the library. If you have selected the e-book reader device, the books will be uploaded to the device, and so on.
2. **Add books from folders and sub-folders:** Allows you to choose a folder. The folder and all its sub-folders are scanned recursively, and any e-books found are added to the library. You can choose whether to have calibre add all files present in a single folder to a single book record or multiple book records. calibre assumes that each folder contains a single book. All e-book files in a folder are assumed to be the same book in different formats. This action

is the inverse of the [Save to disk](#) (pagina 9) action, i.e. you can *Save to disk*, delete the books and re-add them in single book per folder mode, with no lost information except for the date (this assumes you have not changed any of the setting for the Save to disk action).

3. **Aggiungere libri multipli da un archivio (ZIP/RAR):** Permette di aggiungere molteplici e-book che sono immagazzinati all'interno del file ZIP o RAR selezionato. E' una comoda scorciatoia che consente di evitare di decomprimere dapprima l'archivio e poi di aggiungere i libri mediante una delle due opzioni di cui sopra.
4. **Aggiungi un libro vuoto (iscrizione di un libro senza formati):** Permette di creare un record vuoto di un libro. Questo può essere utilizzato per inserire manualmente successivamente le informazioni relative a un libro che potreste non avere ancora nella vostra raccolta.
5. **Aggiungi con ISBN:** Permette di aggiungere uno o più libri inserendo il loro codice ISBN.
6. **Aggiungi file ai record del libro selezionati:** Permette di aggiungere o di aggiornare i file associati con un libro esistente nella vostra libreria.
7. **Add data files to selected book records:** Allows you to add any number of extra files that will be stored in a data sub-directory in the book directory. See [Aggiunta di file di dati a un libro](#) (pagina 120) for details.
8. **Aggiungi un file vuoto ai record del libro selezionati:** Permette di aggiungere un file vuoto del formato specificato ai record del libro selezionati.

L'azione *Aggiungere libri* può leggere i metadati da una ampia varietà di formati di ebook. Inoltre, prova ad indovinare i metadati dal nome del file, vedere la sezione [Estrazione dei metadati dai nomi dei file](#) (pagina 19) per apprendere come configurare questa opzione.

Per aggiungere un formato ad un libro già esistente, è possibile fare una delle seguenti tre cose:

1. Trascina il file sul pannello dei Dettagli del Libro, sulla parte destra della finestra principale
2. Fai clic con il tasto destro sul pulsante *Aggiungi libri* e scegli *Aggiungi file ai libri selezionati*.
3. Fai clic sul pulsante *Aggiungi libri* nella zona in alto a destra della finestra *Modifica metadati*, accessibile con l'azione [Modifica metadati](#) (pagina 5).

1.1.2 Modifica metadati

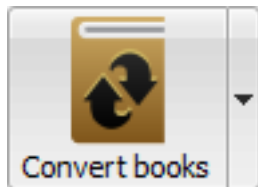


L'azione *Modifica metadati* ha quattro varianti accessibili facendo clic con il tasto destro sul pulsante.

1. **Modifica i metadati individualmente:** Permette di modificare i metadati dei libri uno per uno con l'opzione di scaricare i metadati, incluse le copertine, da Internet. Vi consente inoltre di aggiungere o rimuovere formati di ebook particolari da un libro.
2. **Modifica i metadati in massa:** Permette di modificare i campi comuni dei metadati per un grande numero di libri simultaneamente. Opera su tutti i libri che avete selezionato nella [Library view](#) (pagina 12).
3. **Scarica metadati e copertine:** Scarica metadati e copertine (se disponibili) per i libri che sono selezionati nella lista dei libri.
4. **Unisci i record dei libri:** Vi dà la capacità di unire i metadati ed i formati di due o più record di libri. potete scegliere di eliminare o mantenere i record che non sono stati cliccati prima.
5. **Gestisci i file di dati:** Gestisce i file di dati aggiuntivi associati ai libri selezionati.

Per maggiori dettagli vedi metadato.

1.1.3 Converti libri



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

Per la maggior parte delle persone, la conversione dovrebbe essere un semplice affare da un click. Se vuoi apprendere di più riguardo al processo di conversione, vedi *conversione*.

L'azione *Converti libri* ha tre varianti, accessibili facendo clic con il tasto destro sul pulsante.

1. **Converti individualmente:** Permette di specificare le opzioni per personalizzare la conversione di ognuno degli e-book selezionati.
2. **Converti in gruppo:** Permette di specificare le opzioni una volta sola per convertire un certo numero di e-book in gruppo.
3. **Creare un catalogo dei libri nella vostra libreria di calibre:** Permette di generare un elenco completo dei libri nella vostra libreria, inclusi tutti i metadati, in diversi formati quali XML, CSV, BiBTeX, EPUB e MOBI. Il catalogo conterrà tutti i libri al momento presenti nella vista della libreria. Questo vi consente di utilizzare le funzionalità di ricerca per limitare la catalogazione dei libri. In aggiunta, se selezionate più libri utilizzando il mouse, solo quei libri saranno aggiunti al catalogo. Se generate il catalogo in un formato ebook quale EPUB, MOBI o AZW3, la prossima volta che connettete il vostro lettore di ebook il catalogo sarà automaticamente inviato al dispositivo. Per maggiori informazioni sul funzionamento del catalogo, leggete il *Creazione di cataloghi AZW3 • EPUB • MOBI* (pagina 234).

1.1.4 Leggi

L'azione *Leggi* visualizza il libro in un programma di lettura degli ebook. calibre ha un lettore incorporato per diversi formati di ebook. Per altri formati utilizza l'applicazione predefinita del sistema operativo. Potete configurare quali formati devono essere aperti con il lettore interno mediante *Preferenze* → *Interfaccia* → *Comportamento*. Se un libro ha più di un formato, potete leggere un particolare formato effettuando un click destro sul pulsante.

1.1.5 Invia al dispositivo



L'azione *Invia al dispositivo* ha otto varianti, accessibili facendo clic con il tasto destro sul pulsante.

1. **Invia alla memoria principale:** I libri selezionati sono trasferiti alla memoria principale del lettore di ebook.
2. **Invia alla scheda (A):** I libri selezionati sono trasferiti alla scheda di memoria (A) sul lettore di ebook.

3. **Invia alla scheda (B):** I libri selezionati sono trasferiti alla scheda di memoria (B) sul lettore di ebook.
4. **Send specific format to:** The selected books are transferred to the selected storage location on the device, in the format that you specify.
5. **Espelli dispositivo:** Scollega il dispositivo da calibre.
6. **Set default send to device action:** Allows you to specify which of the options, 1 through 5 above or 7 below, will be the default action when you click the main button.
7. **Invia ed elimina dalla libreria:** I libri selezionati sono trasferiti alla posizione di archiviazione selezionata sul dispositivo e poi **eliminati** dalla Libreria.
8. **Scarica Annotazioni (sperimentale):** Trasferisce le annotazioni che potreste avere fatto su un ebook sul vostro dispositivo ai metadati dei commenti del libro nella libreria di Calibre.

You can control the file name and folder structure of files sent to the device by setting up a template in *Preferences* → *Import/export* → *Sending books to devices*. Also see *Il linguaggio dei modelli di calibre* (pagina 153).

1.1.6 Scarica notizie



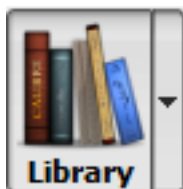
The *Fetch news* action downloads news from various websites and converts it into an e-book that can be read on your e-book reader. Normally, the newly created e-book is added to your e-book library, but if an e-book reader is connected at the time the download finishes, the news is also uploaded to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

The *Fetch news* action uses simple recipes (10-15 lines of code) for each news site. To learn how to create recipes for your own news sources, see *Aggiunta del tuo sito di notizie preferito* (pagina 29).

L'azione *Scarica novità* ha tre variazioni a cui si accede con un click destro sul pulsante.

1. **Schedule news download:** Allows you to schedule the download of your selected news sources from a list of hundreds available. Scheduling can be set individually for each news source you select and the scheduling is flexible allowing you to select specific days of the week or a frequency of days between downloads.
2. **Add a custom news source:** Allows you to create a simple recipe for downloading news from a custom news site that you wish to access. Creating the recipe can be as simple as specifying an RSS news feed URL, or you can be more prescriptive by creating Python-based code for the task. For more information, see *Aggiunta del tuo sito di notizie preferito* (pagina 29).
3. **Download all scheduled news sources:** Causes calibre to immediately begin downloading all news sources that you have scheduled.

1.1.7 Biblioteca



L'azione *Libreria* permette di creare, commutare tra, rinominare o rimuovere una libreria. calibre permette di creare quante librerie si vogliono. Potete, per esempio, creare una libreria di narrativa, una libreria di saggistica, una libreria in lingua straniera, una libreria di progetti, o qualsiasi altra struttura che si adatti alle vostre esigenze. Le

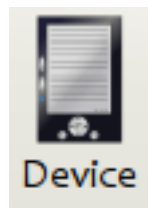
librerie sono la massima struttura organizzativa all'interno di calibre. Ogni libreria ha il proprio insieme di libri, etichette, categorie e posizioni di archiviazione di base.

1. **Commuta/crea libreria:** permette di; a) connettersi a una pre-esistente libreria di calibre in una diversa posizione, b) creare una libreria vuota in una nuova posizione oppure, c) spostare la libreria corrente in una nuova specifica posizione.
2. **Commutazione veloce:** Permette di commutare tra le librerie che sono state registrate o create in calibre.
3. **Rinomina libreria:** Vi consente di rinominare una Libreria.
4. **Scegli un libro in maniera casuale:** Sceglie per voi un libro in maniera casuale nella Libreria.
5. **Rimuovi libreria:** Vi consente di annullare la registrazione di una libreria da Calibre.
6. **Esporta/importa tutti i dati di Calibre:** Vi consente di esportare i dati di calibre per la migrazione ad un nuovo computer oppure di importare i dati precedentemente esportati.
7. **<library name>:** Actions 7, 8 etc... give you immediate switch access between multiple libraries that you have created or attached to. This list contains only the 5 most frequently used libraries. For the complete list, use the Quick Switch menu.
8. **Library maintenance:** Allows you to check the current library for data consistency issues and restore the current library's database from backups.

Nota: I metadati relativi ai vostri ebook, per esempio il titolo, l'autore e le etichette, sono immagazzinati in un singolo file nella vostra cartella della libreria di calibre chiamata metadata.db. Se questo file si corrompe (un evento molto raro), potreste perdere i metadati. Per fortuna, calibre fa automaticamente il backup dei metadati per ogni singolo libro nella cartella dei libri come un file OPF. Utilizzando l'azione Ripristina Database sotto Manutenzione Libreria descritta sopra, potete ottenere che calibre ricostruisca il file metadata.db dal rispettivo file OPF per voi.

Potete copiare o spostare libri tra differenti librerie (una volta che avrete impostato più di una libreria) facendo click destro sul libro e selezionando l'azione *Copia nella libreria*.

1.1.8 Dispositivo



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

1.1.9 Salva su disco



L'azione *Salva su disco* ha cinque varianti, accessibili facendo clic con il tasto destro sul pulsante.

1. **Salva sul disco:** Salva i libri selezionati sul disco organizzati in cartelle. La struttura della cartella ha questo aspetto:

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

Potete controllare il nome del file e la struttura della cartella dei file salvati su disco impostando un modello in *Preferenze* → *Importare/esportare* → *Salvare libri sul disco*. Vedere anche *Il linguaggio dei modelli di calibre* (pagina 153).

2. **Salva su disco in una singola cartella:** Salva i libri selezionati sul disco in una singola cartella.

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

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

1.1.10 Connetti/condividi



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

L'azione *Connetti/condividi* ha quattro varianti, alle quali si accede facendo un click destro sul pulsante.

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

3. **Setup email based sharing of books:** Allows sharing of books and news feeds by email. After setting up email addresses for this option, calibre will send news updates and book updates to the entered email addresses. You can configure how calibre sends email by setting preferences at *Preferences* → *Sharing* → *Sharing books by email*. Once you have set up one or more email addresses, this menu entry will be replaced by menu entries to send books to the configured email addresses.

1.1.11 Rimuovi libri

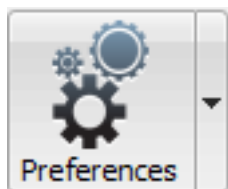


The *Remove books* action **deletes books permanently**, so use it with care. It is *context sensitive*, i.e. it depends on which *catalog* (pagina 11) you have selected. If you have selected the *Library*, books will be removed from the library. If you have selected the e-book reader device, books will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

1. **Rimuovi libri selezionati:** Vi consente di rimuovere **permanentemente** tutti i libri che sono selezionati nell'elenco dei libri.
2. °°Rimuovere i file di uno specifico formato dai libri selezionati...°°: Permette di rimuovere **permanentemente** file ebook di uno specifico formato dai libri che sono selezionati nella lista dei libri.
3. **Remove all formats from selected books, except...:** Allows you to **permanently** remove e-book files of any format except a specified format from books that are selected in the book list.
4. **Remove all formats from selected books:** Allows you to **permanently** remove all e-book files from books that are selected in the book list. Only the metadata will remain.
5. **Rimuovi le copertine dai libri selezionati:** Vi consente di rimuovere **permanentemente** i file di immagine delle copertine dai libri selezionati nella lista dei libri.
6. **Remove matching books from device:** Allows you to remove e-book files from a connected device that match the books that are selected in the book list.
7. **Ripristina eliminati di recente:** Vi consente di annullare la rimozione di libri o formati.

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

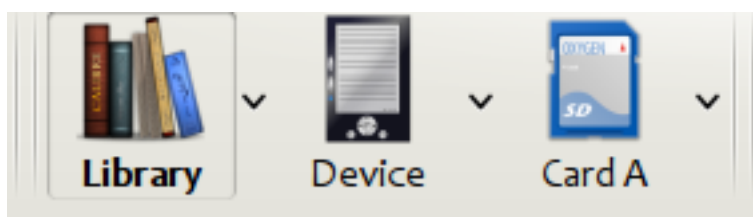
1.2 Preferenze



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

1. **Preferenze:** Vi consente di modificare il modo in cui vari aspetti di Calibre funzionano. Anche cliccando il pulsante si esegue questa azione.
2. Esegui il **Wizard di benvenuto:** Ti permette di avviare il *Wizard di benvenuto*, apparso la prima volta che hai avviato calibre.
3. **Ottieni plugin per migliorare calibre:** Apre una nuova finestra che mostra i plugin per calibre. Questi plugin sono sviluppati da terze parti per ampliare le funzionalità di calibre.
4. **Restart in debug mode:** Allows you to enable a debugging mode that can assist the calibre developers in solving problems you encounter with the program. For most users this should remain disabled unless instructed by a developer to enable it.

1.3 Cataloghi



Un *catalogo* è una raccolta di libri. Calibre può gestire due differenti tipi di cataloghi:

1. **Libreria:** Questa è una raccolta di libri immagazzinata nella vostra libreria di Calibre sul vostro computer.
2. **Dispositivo:** Questa è una raccolta di libri immagazzinata nel vostro lettore di ebook. Sarà resa disponibile quando connettete il lettore al vostro computer.

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

1.4 Cerca e riordina



	Title	Author(s)	Size (MB)	Date	Rating	Publisher	Tags	Series
1	The Complete Works of William Shakespeare	William Shakespeare	2.4	02 Jan 2007	★★★★★	manybooks.net		
2	Stalky and Co.	Rudyard Kipling	0.2	19 Jan 2007	★★★★★	manybooks.net		
3	The Comedies of William Shakespeare	William Shakespeare	2.1	15 Mar 2007	★★★★★			
4	The Histories of William Shakespeare	William Shakespeare	1.5	15 Mar 2007	★★★★★		england, historical fiction	
5	The Tragedies of William Shakespeare	William Shakespeare	1.6	15 Mar 2007	★★★★★			
6	War and Peace	Leo Tolstoy	3.1	22 Aug 2007	★★★★★	gutenberg.org	classic	
7	Anna Karenina	Leo Tolstoy	1.9	22 Aug 2007	★★★★★	gutenberg.org	classic	
8	Guns, germs, and steel: the fates of human societies	Jared Diamond	0.4	29 Nov 2007	★★★★★	New York : W.W. Norton, c1997.		
9	A Game of Thrones	George R. R. Martin	1.3	23 Jan 2007	★★★★★		fantasy	
10	A Clash of Kings	George R. R. Martin	1.4	25 Jan 2007	★★★★★		fantasy	
11	A Storm of Swords	George R. R. Martin	1.9	27 Jan 2007	★★★★★		fantasy	
12	A Feast for Crows	George R. R. Martin	1.7	29 Jan 2007	★★★★★		fantasy	Song of Ice and Fire [4]
13	Dune	Frank Herbert	0.9	00 May 2007	★★★★★		fantasy	The Dune Series [1]

La sezione Cerca e Riordina vi consente di compiere diverse poderose azioni sulla vostra collezione di libri.

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

1.5 L'interfaccia di ricerca

Potete ricercare tutti i metadati del libro inserendo i termini della ricerca nella Barra di ricerca. Ad esempio:

```
Asimov Foundation format:lrf
```

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

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


Ricerca tipi

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

“Contains” searches

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

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

then these searches find:

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

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

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

“Equality” searches

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

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

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

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

- `tags: ". .WWII" : True`. The hierarchy contains the value WWII.
- `tags: ". .Military.WWII" : False`. The `. .` search looks for single values.

“Regular expression” searches

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

“Character variant” searches

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

- letter case is always ignored.
- character variants always match each other.
- punteggiatura e spazi vuoti sono sempre significativi.

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

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

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

then these character variant searches find:

- `title: "^g"` corrisponde ad entrambe (“e” corrisponde ad entrambe “é” ed “e”)
- `title: "^g"` corrisponde a entrambi
- `title: "^g "` corrisponde a #2 perchè lo spazio è significativo
- `title: "^g, "` corrisponde a #1 perchè la virgola è significativa
- `title: "^gb"` non corrisponde a niente perché lo spazio e la virgola sono significativi
- `title: "^g b"` corrisponde a #2 perché la virgola è significativa
- `title: "^db"` non corrisponde a nulla
- `title: "^, "` corrisponde a #1 (invece di tutti i libri) perché la virgola è significativa

Sintassi per le Espressioni di Ricerca

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

- `not tag:foo` ricerca tutti i libri che non contengono l’etichetta `foo`
- `not (author:Asimov or author:Weber)` ricerca tutti i libri non scritti da Asimov o Weber.

³ <https://docs.python.org/library/re.html>

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

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

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

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

Più informazioni

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

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

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



on the left of the search box.

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

Date

La sintassi per ricercare per date è:

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

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

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

Per evitare potenziali problemi con le stringhe tradotte quando si utilizza una versione di calibre non in lingua inglese, le stringhe `_today`, `_yesterday`, `_thismonth`, e `_daysago` sono sempre disponibili. Esse non sono tradotte.

Ricercando date e valori numerici con confronti relazionali

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

Potete ricercare per libri che hanno un formato di una certa dimensione come questo:

- `size:>1.1M` troverà libri di un formato più grande di 1.1 MB
- `size:<=1K` troverà libri con un formato inferiore o uguale a 1 KB

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

Indici di serie

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

Ricerche salvate

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

Librerie virtuali

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

Se un campo ha un valore

Potete ricercare per l'assenza o la presenza di un valore per un campo usando «vero» e «falso». Per esempio:

- `cover:false` trova tutti i libri senza una copertina
- `series:true` trova tutti i libri che sono in una serie
- `series:false` trova tutti i libri che non sono in una serie
- `comments:false` trova tutti i libri con una nota vuota
- `formats:false` finds all books with no book files (empty records)

Colonne personalizzate si/no

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

Identificatori

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

- `identifiers:true` troverà libri con qualsiasi identificatore
- `identifiers:false` troverà libri senza identificatori
- `identifiers:123` ricercheranno libri di qualsiasi genere avente un valore contenente *123*.
- `identifiers:=123456789` will search for books with any type having a value equal to *123456789*.

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

Categories visible in the Tag browser

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

ricerca utilizzando modelli


You can search using a template in *Il linguaggio dei modelli di calibre* (pagina 153) instead of a metadata field. To do so you enter a template, a search type, and the value to search for. The syntax is:

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

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

Esempi:

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

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

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

Finestra di ricerca avanzata

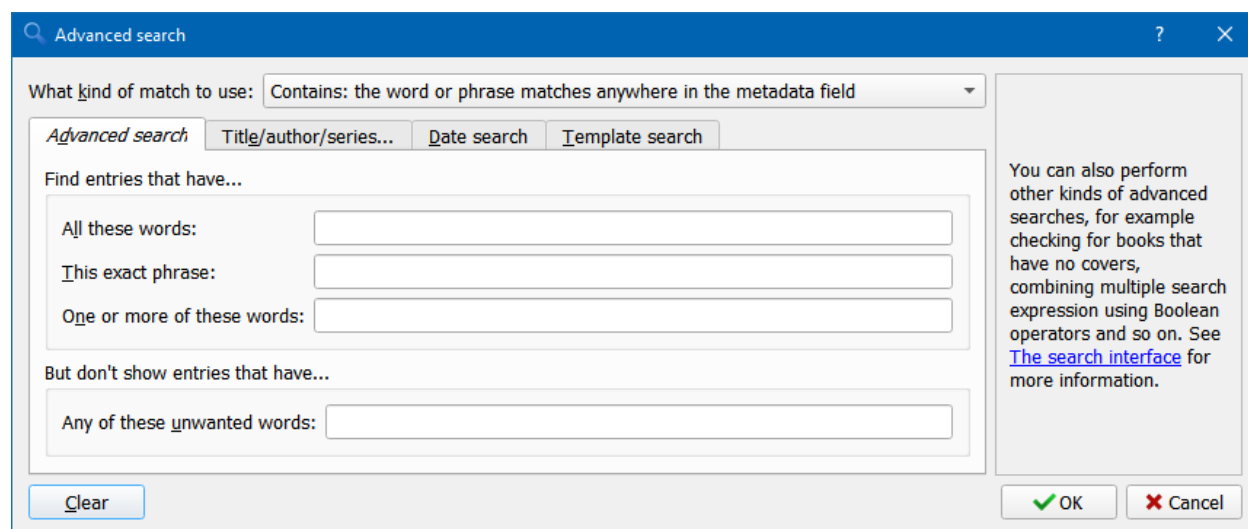


Fig. 1: Finestra di ricerca avanzata

1.6 Salvare le ricerche

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

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

1.7 Ricerca nei contenuti di tutti i libri



calibre può *facoltativamente* indicizzare i contenuti dei libri della biblioteca, in modo permettere la ricerca istantanea di parole dentro ciascuno dei libri. Per usare questa funzionalità clicca sul pulsante *FT* (Full Text) al margine sinistro della barra di ricerca.

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

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

1.8 Biblioteche virtuali

A *Virtual library* is a way to pretend that your calibre library has only a few books instead of its full collection. This is an excellent way to partition your large collection of books into smaller, manageable chunks. To learn how to create and use Virtual libraries, see the tutorial: *Biblioteche Virtuali* (pagina 239).

1.9 Marcatura temporanea dei libri

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

Potete contrassegnare i libri con una specifica etichetta di testo effettuando un click destro sul pulsante *Contrassegna libri* e selezionando *Contrassegna libri con etichetta di testo*. I libri contrassegnati con un'etichetta di testo possono essere trovati successivamente utilizzando la ricerca `marked:"=the-text-you-entered"`.

1.10 Estrazione dei metadati dai nomi dei file

Normalmente, calibre legge i metadati dall'interno del file del libro. Tuttavia, può essere invece configurato per leggere i metadati dal nome del file, per mezzo di *Preferenze* → *Importazione/esportazione* → *Aggiungere libri* → *Leggi i metadati dal contenuto del file*.

You can also control how metadata is read from the filename using regular expressions (see *Tutto sull'utilizzo delle espressioni regolari in calibre* (pagina 205)). In the *Adding books* section of the configuration dialog, you can specify a regular expression that calibre will use to try and guess metadata from the names of e-book files that you add to the library. The default regular expression is:

```
title - author
```

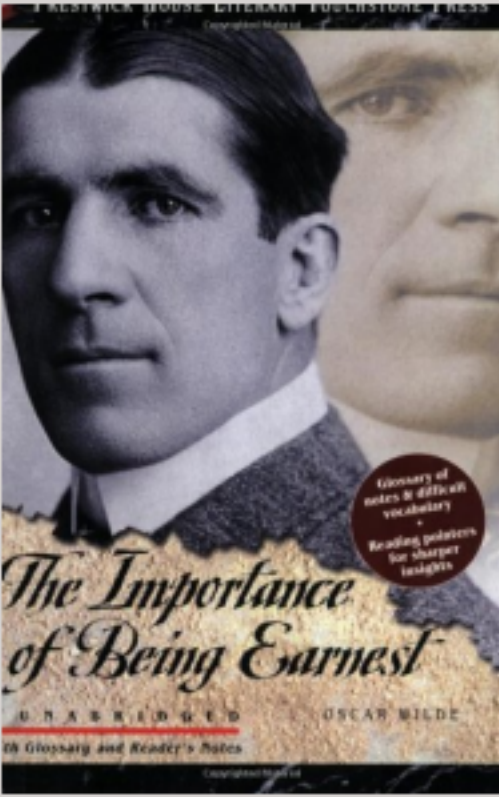
that is, it assumes that all characters up to the first `-` are the title of the book and subsequent characters are the author of the book. For example, the filename:

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

Sarà interpretato avere il titolo: Fondazione e Terra e autore: Isaac Asimov

Suggerimento: Se il nome del file non contiene il trattino, l'espressione regolare di cui sopra fallirà.

1.11 Dettagli del libro



Authors: [Oscar Wilde](#)

Formats: [EPUB](#)

Ids: [9781580495806](#)

Tags: [lit 101 homework](#)

Path: [Click to open](#)

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

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

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

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

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

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

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

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

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

1.12 Navigatore dei Tag



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

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

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

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

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

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

It is also possible to create hierarchies inside some of the text categories such as tags, series, and custom columns. These hierarchies show with the small triangle, permitting the sub-items to be hidden. To use hierarchies of items in a category, you must first go to *Preferences* → *Interface* → *Look & feel* and enter the category name(s) into the «Categories with hierarchical items» field. Once this is done, items in that category that contain periods will be shown using the small triangle. For example, assume you create a custom column called «Genre» and indicate that it contains hierarchical items. Once done, items such as *Mystery.Thriller* and *Mystery.English* will display as *Mystery* with the small triangle next to it. Clicking on the triangle will show *Thriller* and *English* as sub-items. See *Managing subgroups of books, for example «genre»* (pagina 145) for more information.

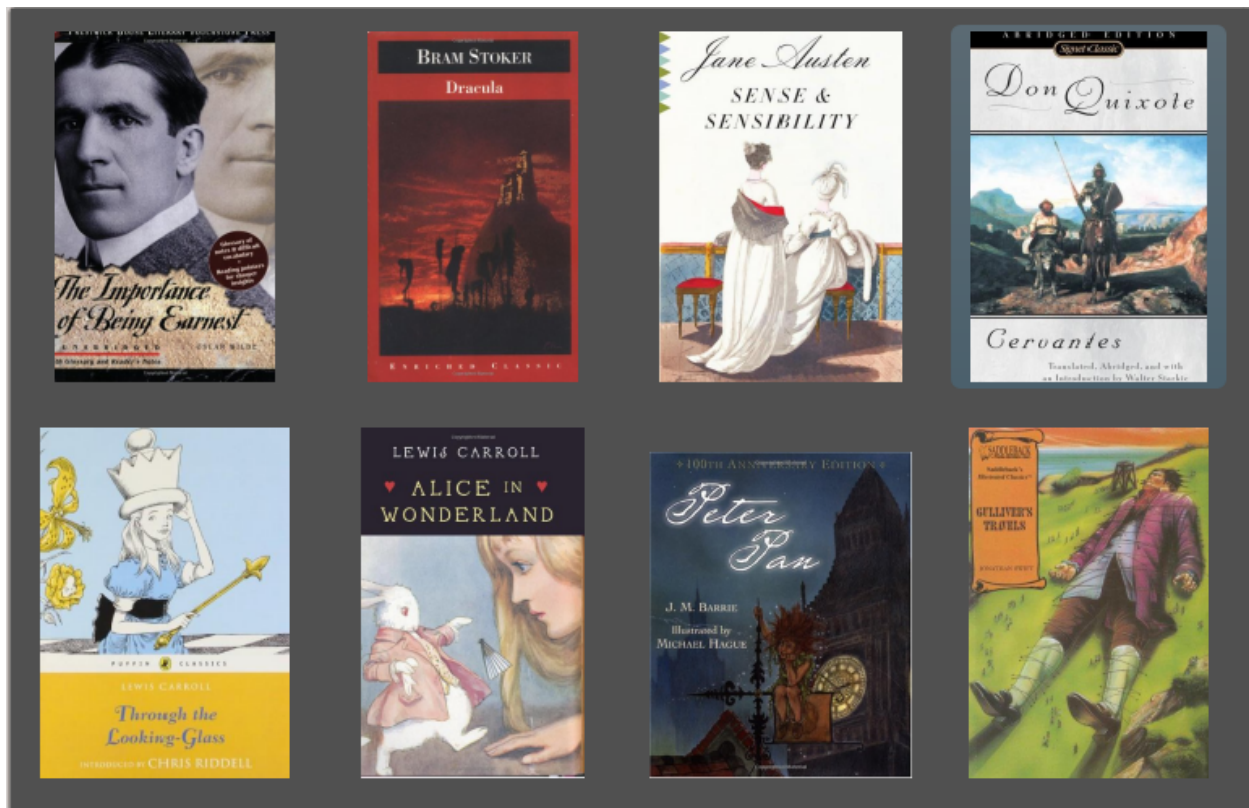
Hierarchical items (items with children) use the same four “click-on” searches as User categories. Items that do not have children use two of the searches: «everything matching» and «everything not matching».

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

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

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

1.13 Griglia delle copertine



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

1.14 Navigatore delle copertine



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

1.15 Quickview

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

Some example Quickview usages: quickly seeing what other books:

- have some tag(s) applied to the currently selected book,
- sono nelle stesse serie come il libro corrente
- have the same values in a custom column as the current book
- sono scritti da uno degli stessi autori del libro corrente
- share values in a custom column

There are two choices for where the Quickview information appears:

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

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

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

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

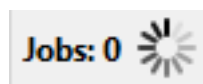
Shift- or Ctrl- double-click on a book in the Quickview window to open the edit metadata dialog on that book in the calibre window. The edited book will be Quickview'ed when you close the edit metadata dialog.

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

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

- Respect (or not) the current Virtual library. If checked then Quickview shows only books in the current Virtual library. Default: respect Virtual libraries
- Change the Quickview window contents when the column is changed on the book list using the cursor keys. Default: don't follow changes made with cursor keys
- Change the column being «quickview'ed» when a cell in the Quickview window is double-clicked. Otherwise the book is changed but the column being examined is not. Default: change the column
- Change the column being «quickview'ed» to the current column when the return key is pressed in the Quickview panel. Otherwise the book is changed but the column being examined is not. Default: change the column
- Scegliere quali colonne vengono visualizzate nella finestra/pannello Quickview.

1.16 Lavori



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

1.17 Scorciatoie da tastiera

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

Nota: Nota: le scorciatoie da tastiera di calibre non richiedono un tasto modificatore (Command, Option, Control, etc.), a meno che non sia specificamente indicato. Sarà necessario premere solo il tasto della lettera, ad esempio E per modificare.

Tabella 1: Keyboard shortcuts for the main calibre program

Scorciatoia da tastiera	Azione
F2 (In- vio in ma- cOS)	Edit the metadata of the currently selected field in the book list.
A	Aggiungi libri
Shift+.	Aggiungere formati ai libri selezionati
C	Convertire i libri selezionati
D	Invia al dispositivo
Del	Rimuovi i libri selezionati
E	Modifica i metadati dei libri selezionati
G	Ottieni libri
I	Mostra i dettagli del libro
K	Modifica indice
M	Unisci i record selezionati
Alt+M	Unisci i record selezionati, mantenendo gli originali
O	Apri cartella contenente
P	Pulisci libri
S	Salva su disco
T	Modifica libro
V	Leggi
Shift+^	Visualizza l'ultimo libro letto
Alt+V/ Cmd+V in macOS	Leggi un formato specifico
Alt+Sh	Mostra/nascondi l'elenco dei lavori
Alt+Sh	Mostra/nascondi il Navigatore delle copertine
Alt+Sh	Mostra/nascondi il pannello Dettagli libro
Alt+Sh	Mostra/nascondi il Navigatore dei Tag
Alt+Sh	Mostra/nascondi la Griglia delle copertine
Alt+A	Mostra libri dello stesso autore del libro corrente
Alt+T	Mostra libri con gli stessi tag del libro corrente
Alt+P	Mostra libri dello stesso editore del libro corrente
Alt+Sh	Mostra libri della stessa serie del libro corrente
/, Ctrl+F	Posiziona il cursore nella Barra di Ricerca
Shift+.	Apri la finestra di Ricerca Avanzata
Shift+.	Mostra/nascondi la Barra di Ricerca
Esc	Cancella la ricerca corrente
Shift+.	Attiva l'elenco dei libri
Ctrl+E	Abbandona la Biblioteca virtuale
Alt+Es	Cancella la restrizione aggiuntiva
Ctrl+*	Crea una Biblioteca Virtuale temporanea basata sulla ricerca corrente
Ctrl+R	Seleziona la scheda di Biblioteca Virtuale successiva
Ctrl+L	Seleziona la scheda di Biblioteca Virtuale precedente

continues on next page

Tabella 1 – continua dalla pagina precedente

Scor- ciatoia da ta- stiera	Azione
N op- pure F3	Trova il libro successivo che corrisponde alla ricerca corrente (funziona solo se l'evidenziazione dei risultati è attivata nelle preferenze di ricerca)
Shift+ op- pure Shift+	Trova il libro precedente che corrisponde alla ricerca corrente (funziona solo se l'evidenziazione dei risultati è attivata nelle preferenze di ricerca)
Ctrl+D	Scarica metadati e copertine
Ctrl+R	Riavvia calibre
Ctrl+S	Riavvia calibre in modalità debug
Shift+	Aggiungi libri vuoti a calibre
Ctrl+M	Marca/rimuovi marcatura dai libri selezionati
Ctrl+/ o Ctrl+A	Apri la finestra di ricerca nei contenuti di tutti i libri della biblioteca
Q	Apri il pannello Quickview per visualizzare i libri con serie/tag/ecc. collegati
Shift+	Sposta il focus sul pannello Quickview
Shift+	Esegui una ricerca nel pannello Quickview
F5	Riapplica l'ordinamento corrente
Ctrl+Q	Esci da calibre
X	Attiva/disattiva lo scorrimento automatico dell'elenco dei libri
Ctrl+A	Restrict the displayed books to only those books that are in a category currently displayed in the <i>Tag browser</i>

Aggiunta del tuo sito di notizie preferito

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.

To gain an understanding of how to use the framework, follow the examples in the order listed below:

- *Recupero completamente automatico* (pagina 30)
 - *Il blog di calibre* (pagina 30)
 - *bbc.co.uk* (pagina 31)
- *Personalizzazione del processo di recupero* (pagina 31)
 - *Using the print version of bbc.co.uk* (pagina 31)
 - *Sostituire gli stili dell'articolo* (pagina 33)
 - *Slicing and dicing* (pagina 33)
 - *Esempio di vita reale* (pagina 34)
- *Suggerimenti per lo sviluppo di nuove ricette* (pagina 36)
- *Ulteriori approfondimenti* (pagina 37)
- *Documentazione API* (pagina 37)

2.1 Recupero completamente automatico

If your news source is simple enough, calibre may well be able to fetch it completely automatically, all you need to do is provide the URL. calibre gathers all the information needed to download a news source into a *recipe*. In order to tell calibre about a news source, you have to create a *recipe* for it. Let's see some examples:

2.1.1 Il blog di 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.




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

Recipe title:

Oldest article: 

Max. number of articles per feed: 


Feeds in recipe






Add feed to recipe


Feed title:

Feed URL:

 Add feed

 Save

Switch to Advanced mode

 Cancel

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

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

2.2 Personalizzazione del processo di recupero

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

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

2.2.1 Using the print version of bbc.co.uk

The first step is to look at the e-book we downloaded previously from *bbc.co.uk* (pagina 31). 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 articolo

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

URL versione di stampa

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

So it looks like to get the print version, we need to prefix every article URL with:

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

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

```
Recipe source code (python)

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

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

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/newsonlin

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

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

Now, click the *Add/update recipe* button and your changes will be saved. Re-download the e-book. You should have a much improved e-book. One of the problems with the new version is that the fonts on the print version webpage are too small. This is automatically fixed when converting to an e-book, but even after the fixing process, the font size of the

⁴ <https://www.python.org>

⁵ <https://docs.python.org/tutorial/>

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 Sostituire gli stili dell'articolo

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

```
no_stylesheets = True
```

The recipe now looks like:

```
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/
```

The new version looks pretty good. If you're a perfectionist, you'll want to read the next section, which deals with actually modifying the downloaded content.

2.2.3 Slicing and dicing

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* (pagina 33) recipe again. Looking at the source code (*HTML*) of a couple of articles (print version), we see that they have a footer that contains no useful information, contained in

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

Può essere rimosso aggiungendo:

```
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 Esempio di vita reale

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

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

class NYTimes(BasicNewsRecipe):

    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)
```

(continues on next page)

(continua dalla pagina precedente)

```

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

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

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

```

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* (pagina 46).

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* (pagina 45), *remove_tags_before* (pagina 45), *remove_tags_after* (pagina 45).

The next interesting feature is:

```

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()` (pagina 38) actually does the login into the NYT website. Once logged in, calibre will use the same, logged in, browser instance to fetch all content. See [mechanize](#)⁶ to understand the code in `get_browser`.

The next new feature is the `calibre.web.feeds.news.BasicNewsRecipe.parse_index()` (pagina 40) method. Its job is to go to <https://www.nytimes.com/pages/todayspaper/index.html> and fetch the list of articles that appear in *today's* paper. While more complex than simply using [RSS](#), the recipe creates an e-book that corresponds very closely to the days paper. `parse_index` makes heavy use of [BeautifulSoup](#)⁷ to parse the daily paper webpage. You can also use other, more modern parsers if you dislike BeautifulSoup. calibre comes with [lxml](#)⁸ and [html5lib](#)⁹, which are the recommended parsers. To use them, replace the call to `index_to_soup()` with the following:

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

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

2.3 Suggerimenti per lo sviluppo di nuove ricette

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` (pagina 327) option.

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

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

If you're satisfied with your recipe, and you feel there is enough demand to justify its inclusion into the set of built-in recipes, post your recipe in the [calibre recipes forum](#)¹⁰ to share it with other calibre users.

⁶ <https://mechanize.readthedocs.io/en/latest/>

⁷ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

⁸ <https://lxml.de/>

⁹ <https://github.com/html5lib/html5lib-python>

¹⁰ <https://www.mobileread.com/forums/forumdisplay.php?f=228>

Nota: In macOS, gli strumenti da linea di comando si trovano dentro il bundle di calibre, per esempio, se hai installato calibre in `/Applications` gli strumenti si trovano in `/Applications/calibre.app/Contents/MacOS/`.

Vedi anche:

generated/en/ebook-convert

The command line interface for all e-book conversion.

2.4 Ulteriori approfondimenti

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

API documentation (pagina 37)

Documentation of the `BasicNewsRecipe` class and all its important methods and fields.

BasicNewsRecipe¹¹

The source code of `BasicNewsRecipe`

Built-in recipes¹²

The source code for the built-in recipes that come with calibre

The calibre recipes forum¹³

Lots of knowledgeable calibre recipe writers hang out here.

2.5 Documentazione API

2.5.1 Documentazione API per le ricette

L'API per scrivere ricette è definita dalla classe `BasicNewsRecipe`

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

Base class that contains logic needed in all recipes. By overriding progressively more of the functionality in this class, you can make progressively more customized/powerful recipes. For a tutorial introduction to creating recipes, see *Aggiunta del tuo sito di notizie preferito* (pagina 29).

abort_article (*msg=None*)

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

abort_recipe_processing (*msg*)

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

add_toc_thumbnail (*article, src*)

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

¹¹ <https://github.com/kovidgoyal/calibre/blob/master/src/calibre/web/feeds/news.py>

¹² <https://github.com/kovidgoyal/calibre/tree/master/recipes>

¹³ <https://www.mobileread.com/forums/forumdisplay.php?f=228>

classmethod `adeify_images (soup)`

If your recipe when converted to EPUB has problems with images when viewed in Adobe Digital Editions, call this method from within `postprocess_html ()` (pagina 40).

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

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

Parametri

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

cleanup ()

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

clone_browser (*br*)

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

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

default_cover (*cover_file*)

Crea una copertina generica per le ricette che non hanno una copertina

download ()

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

extract_readable_article (*html*, *url*)

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

get_article_url (*article*)

Override in a subclass to customize extraction of the [URL](#) that points to the content for each article. Return the article URL. It is called with *article*, an object representing a parsed article from a feed. See [feedparser](#)¹⁴. By default it looks for the original link (for feeds syndicated via a service like FeedBurner or Pheedo) and if found, returns that or else returns [article.link](#)¹⁵.

get_browser (*args, **kwargs)

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

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

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

(continues on next page)

(continua dalla pagina precedente)

```
br.submit()
return br
```

get_cover_url()

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

get_extra_css()

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

get_feeds()

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

get_masthead_title()

Override in subclass to use something other than the recipe title

get_masthead_url()

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

get_obfuscated_article(url)

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

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

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.

Ritorna

A floating point number, the delay in seconds.

classmethod image_url_processor(baseurl, url)

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

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

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

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

is_link_wanted(url, tag)

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

Parametri

- **url** – L'URL da seguire

- **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()` (pagina 39). 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()` (pagina 40).

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

Each list of articles must contain dictionaries of the form:

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

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

If you want to abort processing for some reason and have calibre show the user a simple message instead of an error, call `abort_recipe_processing()` (pagina 37).

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).

Parametri

- **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.

Parametri

- **oeb** – Un oggetto OEBBook
- **opts** – Opzioni di conversione

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.

Parametri

- **soup** – A BeautifulSoup¹⁸ instance containing the downloaded *HTML*.
- **first_fetch** – Vero se questa è la prima pagina di un articolo.

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 BeautifulSoup¹⁹ instance containing the downloaded *HTML*.

preprocess_image (*img_data*, *image_url*)

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

preprocess_raw_html (*raw_html*, *url*)

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

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

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

classmethod print_version (*url*)

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

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

publication_date ()

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

skip_ad_pages (*soup*)

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

soup: A BeautifulSoup²⁰ instance containing the downloaded *HTML*.

sort_index_by (*index*, *weights*)

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

index: A list of titles.

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

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

Convenience method to take a BeautifulSoup²¹ *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: BeautifulSoup²² *Tag*

articles_are_obfuscated = False

Set to True and implement `get_obfuscated_article()` (pagina 39) to handle websites that try to make it difficult to scrape content.

auto_cleanup = False

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

auto_cleanup_keep = None

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

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

center_navbar = True

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

compress_news_images = False

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

compress_news_images_auto_size = 16

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

compress_news_images_max_size = None

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

conversion_options = {}

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

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

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

By default, the cover image returned by `get_cover_url()` will be used as the cover for the periodical. Overriding

this in your recipe instructs calibre to render the downloaded cover into a frame whose width and height are expressed as a percentage of the downloaded cover. `cover_margins = (10, 15, "#ffffff")` pads the cover with a white margin 10px on the left and right, 15px on the top and bottom. Color names are defined [here](#)²³. 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()` (pagina 39) 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()` (pagina 39) method instead. For example:

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

feeds = None

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

filter_regexps = []

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

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

rimuoverà tutti gli URL che hanno al loro interno `ads.doubleclick.net`.

Only one of `BasicNewsRecipe.match_regexps` (pagina 44) or `BasicNewsRecipe.filter_regexps` (pagina 43) should be defined.

handle_gzip = True

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

ignore_duplicate_articles = None

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

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

To use URLs instead, set it to:

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

To match on title or URL, set it to:

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

keep_only_tags = []

Keep only the specified tags and their children. For the format for specifying a tag see [BasicNewsRecipe.remove_tags](#) (pagina 45). 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](#) (pagina 44) or [BasicNewsRecipe.filter_regexps](#) (pagina 43) 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](#) (pagina 44)

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

L'articolo più datato da scaricare da questa fonte di notizie. In giorni.

preprocess_regexps = []

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

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

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

publication_type = 'unknown'

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

recipe_disabled = None

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

recursions = 0

Number of levels of links to follow on article webpages

remove_attributes = []

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

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

remove_empty_feeds = False

If True empty feeds are removed from the output. This option has no effect if `parse_index` is overridden in the sub class. It is meant only for recipes that return a list of feeds using `feeds` or `get_feeds()` (pagina 39). It is also used if you use the `ignore_duplicate_articles` option.

remove_javascript = True

Convenient flag to strip all JavaScript tags from the downloaded HTML

remove_tags = []

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

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

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

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

This will remove all `<div class=»advert»>` tags and all their children from the downloaded *HTML*.

remove_tags_after = None

Remove all tags that occur after the specified tag. For the format for specifying a tag see [BasicNewsRecipe.remove_tags](#) (pagina 45). For example:

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

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

remove_tags_before = None

Remove all tags that occur before the specified tag. For the format for specifying a tag see [BasicNewsRecipe.remove_tags](#) (pagina 45). For example:

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

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

requires_version = (0, 6, 0)

Versione minima di calibre per utilizzare questa ricetta

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()` (pagina 38) to work with the URL scheme of your particular website.

reverse_article_order = False

Inverte l'ordine degli articoli in ogni fonte

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` (pagina 43) > 0

summary_length = 500

Numero massimo di caratteri nella descrizione breve

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

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

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

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

timeout = 120.0

Timeout for fetching files from server in seconds

title = 'Sorgente di notizie sconosciuta'

The title to use for the e-book

use_embedded_content = None

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

¹⁴ <https://pythonhosted.org/feedparser/>

¹⁵ <https://pythonhosted.org/feedparser/reference-entry-link.html>

¹⁶ <https://mechanize.readthedocs.io/en/latest/>

¹⁷ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

¹⁸ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

¹⁹ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

²⁰ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

²¹ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

²² <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

²³ <https://www.imagemagick.org/script/color.php>

²⁴ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/#searching-the-tree>

CAPITOLO 3

Il Lettore E-book

calibre include un lettore di e-book che può visualizzare tutti i principali formati di libro digitale. Il Lettore E-book è altamente personalizzabile e ha molte funzionalità avanzate.

- *Avviare il Lettore E-book* (pagina 48)
- *Navigazione dentro un e-book* (pagina 48)
- *Evidenziazione del testo* (pagina 49)
- *Lettura ad alta voce* (pagina 49)
- *Cercare il testo* (pagina 50)
- *Seguire i collegamenti usando solo la tastiera* (pagina 50)
- *Personalizzare la grafica della tua esperienza di lettura* (pagina 50)
- *Consultazione del dizionario* (pagina 51)
- *Copiare testo e immagini* (pagina 51)
- *Ingrandimento delle immagini* (pagina 51)
- *Contenuto non adattabile* (pagina 51)
- *Progettare il proprio libro in modo che funzioni bene con il lettore di calibre* (pagina 51)

3.1 Avviare il Lettore E-book

Puoi visualizzare ognuno dei libri nella tua biblioteca di calibre selezionando il libro e premendo il pulsante *Leggi*. Ciò farà aprire il libro nel Lettore E-book. Puoi anche avviare il Lettore E-book senza passare da calibre usando il Menu Start su Windows. Su macOS, puoi agganciarlo alla dock e avviarlo da lì. Su Linux puoi usare il suo lanciatore nei menu della scrivania o eseguire il comando **ebook-viewer**.

3.2 Navigazione dentro un e-book

Puoi «girare pagina» in un libro in questi modi:

- Il clic sul margine sinistro o su quello destro della pagina con il mouse
- Premendo i tasti `spacebar`, `page up`, `page down` o le frecce
- Su uno schermo tattile toccando il testo o scorrendo a destra e sinistra

Puoi accedere ai controlli del lettore in questi modi:

- Facendo clic con il tasto destro sul testo
- Premendo i tasti `Esc` o `Menu`
- Su uno schermo tattile toccando la parte (1/3) superiore dello schermo

Il lettore ha due modalità, «pagine» e «flusso». In modalità Pagine il contenuto del libro è presentato in pagine, in modo simile a un libro cartaceo. In modalità Flusso il testo è presentato in modo continuo, come in un browser web. Puoi cambiare modalità usando le *Preferenze* del lettore sotto *Impaginazione* o premendo il tasto `Ctrl+M`.

3.2.1 Segnalibri

Quando sei nel bel mezzo di un libro e chiudi il Lettore E-book, si ricorderà dove hai smesso di leggere e ci ritornerà la prossima volta che aprirai il libro. Puoi creare segnalibri anche usando il pulsante *Segnalibri* nei controlli del Lettore E-book o premendo `Ctrl+B`. Quando visualizzi libri in formato EPUB, questi segnalibri sono salvati proprio dentro il file EPUB stesso. Puoi aggiungere segnalibri e poi inviare il file a un amico. Quando aprirà il file, sarà in grado di vedere i tuoi segnalibri. Puoi disattivare questo comportamento nella sezione *Varie* delle impostazioni del lettore.

3.2.2 Indice

Se il libro che stai leggendo include un Indice, puoi accedervi premendo il pulsante *Indice*. Questo aprirà una lista di sezioni del libro. Puoi fare clic su ciascuna di esse per saltare a quella porzione del libro.

3.2.3 Navigare in base alla collocazione

Gli e-book, a differenza dei libri cartacei, non hanno un concetto di pagina. Puoi riferirti a collocazioni precise dentro agli e-book usando la funzionalità *Vai a* → *Collocazione* nei controlli del lettore.

Puoi usare queste informazioni di collocazione per riferirti senza ambiguità a parti di libri quando ne parli con amici o li citi in altri lavori. Puoi inserire queste collocazioni in *Vai a* → *Collocazione* nei controlli del lettore.

C'è un URL che puoi copiare negli appunti e incollare in altri programmi o documenti. Il clic su questo URL aprirà il libro nel lettore di e-book di calibre alla collocazione corrente.

Se clicchi su collegamenti dentro all'e-book che portano a parti differenti del libro, come a una nota di chiusura, puoi usare i pulsanti *Avanti* e *Indietro* nell'angolo in alto a sinistra dei controlli del lettore. Questi pulsanti si comportano proprio come in un browser web.

3.2.4 Modalità riferimento

calibre ha anche una molto utile *Modalità riferimento*. Puoi attivarla facendo clic sul pulsante *Modalità riferimento* nei controlli del lettore. Una volta fatto, ogni paragrafo avrà un numero univoco mostrato al suo inizio, composto dai numeri di sezione e di paragrafo.

Puoi usare questo numero per riferirti senza ambiguità a parti di libri quando ne parli con amici o li citi in altri lavori. Puoi inserire questi numeri nella *funzione Vai a* per navigare alla posizione di un particolare riferimento.

3.3 Evidenziazione del testo

Quando selezioni del testo nel lettore, compare una piccola barra di fianco alla selezione. Puoi fare clic sul pulsante *Evidenzia* in quella barra per creare un'evidenziazione. Puoi aggiungere note e cambiare il colore dell'evidenziazione. Su uno schermo tattile, tieni premuto a lungo su una parola per selezionarla e mostrare la barra. Una volta in modalità evidenziazione puoi modificare la selezione usando cursori funzionali anche per il tocco. Trascina i cursori ai margini superiori o inferiori per scorrere durante la selezione. Puoi anche usare `Maiusc+clic` o `clic destro` per estendere la selezione, utile in particolare per selezioni multi pagina.

Puoi usare il pulsante *Evidenziazioni* nei controlli del lettore per mostrare un pannello separato con un elenco di tutte le evidenziazioni del libro, ordinate per capitolo.

Puoi navigare *tutte le evidenziazioni* della tua biblioteca di calibre facendo clic con il tasto destro sul pulsante *Visualizza* e scegliendo *Naviga annotazioni*.

Infine, se usi il server dei contenuti di calibre nel lettore del browser, puoi fare in modo che il lettore sincronizzi le sue annotazioni con il lettore del browser andando in *Preferenze* → *Varie* nelle preferenze del lettore e inserendo il nome utente del lettore del server dei contenuti con cui sincronizzare. Usa il carattere speciale `*` per sincronizzare con utenti anonimi.

3.4 Lettura ad alta voce

Il lettore può leggere il testo dei libri ad alta voce. Per usare questa funzione devi semplicemente fare clic sul pulsante *Leggi ad alta voce* nei controlli del lettore per cominciare a leggere il testo del libro ad alta voce. La parola che viene letta in un dato momento è evidenziata. La voce è sintetizzata dal testo usando i servizi text-to-speech (TTS) del tuo sistema operativo. Puoi cambiare la voce usata facendo clic sull'icona della rotellina nella barra che viene mostrata quando *Leggi ad alta voce* è attivo.

Puoi anche leggere ad alta voce i passaggi evidenziati aggiungendo il pulsante *Leggi ad alta voce* alla barra di selezione nelle preferenze del lettore sotto *Selezione*.

Nota: Il supporto a TTS nei browser è molto carente e pieno di problemi quindi il buon funzionamento della *Letture ad alta voce* nel lettore del browser dipenderà da quanto bene il browser supporta il TTS. In particolare, l'evidenziazione della parola corrente non funziona, e cambiare la velocità o la voce narrante farà ripartire la lettura dall'inizio.

Nota: Su Linux *Leggi ad alta voce* richiede che [Speech Dispatcher](https://freebsoft.org/speechd)²⁵ sia installato e funzionante.

²⁵ <https://freebsoft.org/speechd>

Nota: Su Windows non tutte le voci installate potrebbero essere visibili al sottosistema SAPI usato per il TTS. Sono disponibili [istruzioni per rendere tutte le voci visibili](#)²⁶.

3.5 Cercado il testo

Il lettore ha delle potenti capacità di ricerca. Premi `Ctrl+F` o accedi ai controlli del lettore e clicca cerca. La più semplice forma di ricerca è quella di cercare per qualsiasi testo che inserisci nella casella testo. I diversi moduli di ricerca sono scelti nel riquadro modalità di ricerca sottostante la casella di ricerca. Le modalità disponibili sono:

1. *Contains* - La modalità predefinita più semplice. Il testo inserito nella casella di ricerca è cercato ovunque. Tutta la punteggiatura, accenti e spazi sono ignorati. Per esempio, la ricerca `Pena` restituirà tutti i termini seguenti: `penal`, `pen a`, `pen.a` and `Peña`. Se selezioni la casella *Case sensitive*, allora accenti, spazi e punteggiatura non sono più ignorati.
2. *guilabel:Whole words* - Cerca tra tutte le parole. Per esempio, la ricerca `pena` restituirà la parola `Peña` ma non la parola `Penal`. Come per *Contains*, con le ricerche sopracitate, accenti e punteggiatura sono ignorate a meno che la *Case sensitive* sia attivata.
3. *Nearby words* - Cerca intere parole che sono vicine tra di loro. Quindi, per esempio la ricerca `calibre cool` restituirà risultati dove le parole `calibre` e `cool` si trovano all'interno di sessanta caratteri tra di loro. Per cambiare il numero di caratteri aggiungi il nuovo numero alla fine della lista di parole. Per esempio, `calibre cool awesome 120` restituirà risultati dove le tre parole si trovano all'interno di 120 caratteri tra loro. Nota che punteggiatura e accenti sono *not* ignorate da queste ricerche.
4. *Regex* - Interpreta il testo di ricerca come una *espressione regolare*. Per saperne di più su espressioni regolari, vedi, `:doc: the tutorial ``.

3.6 Seguire i collegamenti usando solo la tastiera

Il lettore di e-book ha una *Modalità suggerimenti* che permette di fare clic sui collegamenti nel testo senza usare il mouse. Premi il tasto `Alt+F` e tutti i collegamenti nella schermata attuale saranno evidenziati con un numero o una lettera soprastanti. Premi la lettera sulla tua tastiera per fare clic sul collegamento. Premere il tasto `Esc` terminerà la *Modalità suggerimenti* senza selezionare alcun collegamento.

Se ci sono più di 35 collegamenti nella schermata, alcuni di essi avranno più di una lettera, nel qual caso dovrai premere la prima e la seconda lettera, o la prima seguita da `Invio` per attivare. Puoi anche usare il tasto `Backspace` per annullare un errore di digitazione.

3.7 Personalizzare la grafica della tua esperienza di lettura

Puoi cambiare al volo la dimensione dei caratteri usando *Dimensione carattere* nei controlli del lettore oppure `Ctrl++` e `Ctrl+-`, oppure tenendo premuto il tasto `Ctrl` e usando la rotellina del mouse.

I colori possono essere cambiati nella sezione *Colori* delle preferenze del lettore.

Puoi cambiare il numero di pagine mostrate sullo schermo, come anche i margini di pagina in *Impaginazione* nelle preferenze del lettore.

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

Puoi mostrare intestazioni e piè di pagina personalizzati come il tempo di lettura rimanente, il titolo del capitolo corrente, la posizione nel libro, ecc. attraverso la sezione *Intestazioni e piè di pagina* delle preferenze del lettore.

Personalizzazioni oii avanzate possono essere ottenute attraverso le impostazioni degli *Stili*. Qui puoi specificare un'immagine di sfondo da mostrare sotto al testo, e anche un foglio di stile da impostare che sarà applicato ad ogni libro. Usandolo puoi fare cose come cambiare gli stili di paragrafo, la giustificazione del testo, ecc. Per esempi di fogli di stile personalizzati usati dagli utenti di calibre, visita [i forum](#)²⁷.

3.8 Consultazione del dizionario

Puoi ricercare il significato delle parole nel libro attuale facendo doppio clic o tenendo premuto sulla parola che vuoi cercare, e facendo clic poi sul pulsante di consultazione che assomiglia a una biblioteca.

3.9 Copiare testo e immagini

Puoi selezionare testo e immagini trascinando il contenuto con il mouse e poi facendo clic con il tasto destro e selezionando *Copia* per copiare negli appunti. Il materiale copiato può essere incollato in un'altra applicaziobe come testo e immagini semplici.

3.10 Ingrandimento delle immagini

Puoi ingrandire un immagine per mostrarla a dimensione intera in una finestra separata o facendo doppio clic, o toccando a lungo su di essa. Puoi anche fare clic con il tasto destro del mouse e scegliere *Visualizza immagine*.

3.11 Contenuto non adattabile

Alcuni libri hanno del contenuto molto largo che non può essere spezzato ai confini della pagina, come le tabelle o i tag `<pre>`. In casi del genere, dovresti usare il lettore in *modalità flusso* premendo `Ctrl+M` per leggere questo contenuto. Alternativamente, puoi anche aggiungere il seguente CSS alla sezione *Stili* delle preferenze del lettore, per forzare il lettore a spezzare le linee di testo nei tag `<pre>` tags:

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

3.12 Progettare il proprio libro in modo che funzioni bene con il lettore di calibre

Il lettore di calibre imposterà la classe `is-calibre-viewer` nell'elemento root. Quindi puoi scrivere regole CSS che si applichino solo ad esso. In più, il lettore imposterà le classi seguenti all'elemento `body`:

`body.calibre-viewer-dark-colors`

Impostata quando un tema scuro è utilizzato

`body.calibre-viewer-light-colors`

Impostata quando un tema chiaro è utilizzato

²⁷ <https://www.mobileread.com/forums/showthread.php?t=51500>

body.calibre-viewer-paginated

Impostata quando il lettore è in modalità Pagine

body.calibre-viewer-scrolling

Impostata quando il lettore è in modalità Flusso (non impaginato)

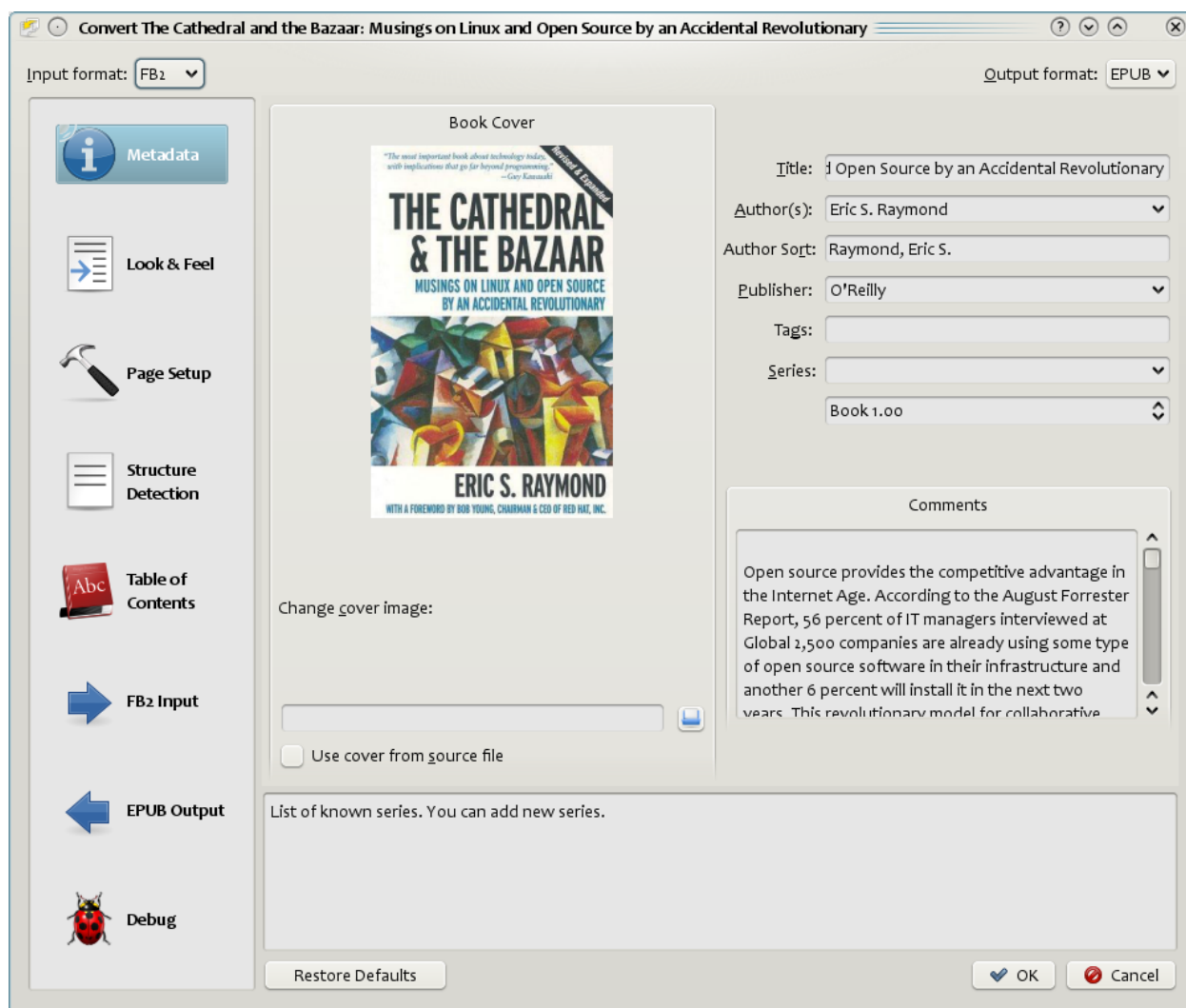
Infine, puoi usare i colori dello schema colori di calibre per mezzo delle [variabili CSS](#)²⁸. Il lettore di calibre definisce le seguenti variabili: `--calibre-viewer-background-color`, `--calibre-viewer-foreground-color` e opzionalmente `--calibre-viewer-link-color` nei temi che definiscono un colore per i collegamenti.

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

Conversione E-book

Calibre ha un sistema di conversione progettato per essere molto semplice, generalmente basta aggiungere un e-book ed il sistema farà di tutto per generare un risultato più fedele possibile all'originale. Calibre accetta vari formati in ingresso ma alcuni sono più adatti di altri alla conversione, se hai questo tipo di file e desideri un maggior controllo sul risultato finale, Calibre dispone di molte opzioni per ottimizzare il processo. Ricorda che il sistema di conversione non può sostituire un editor di e-book dedicato. Per creare un e-book ti suggerisco di convertirlo prima in EPUB o AZW3 e quindi usare la funzione :guilabel:"Modifica Libro" per ottenere l'aspetto migliore, utilizza il file così generato per ottenere altri formati in Calibre.

Questo documento farà riferimento alle impostazioni che si trovano nella finestra di dialogo «Converti Libro» come nella foto qui sotto. Tutte le impostazioni sono disponibili anche da riga di comando come descritto su "ebook-convert". In Calibre puoi ottenere aiuto tenendo il puntatore del mouse sopra ogni singola voce, si aprirà una finestra che descrive l'impostazione.

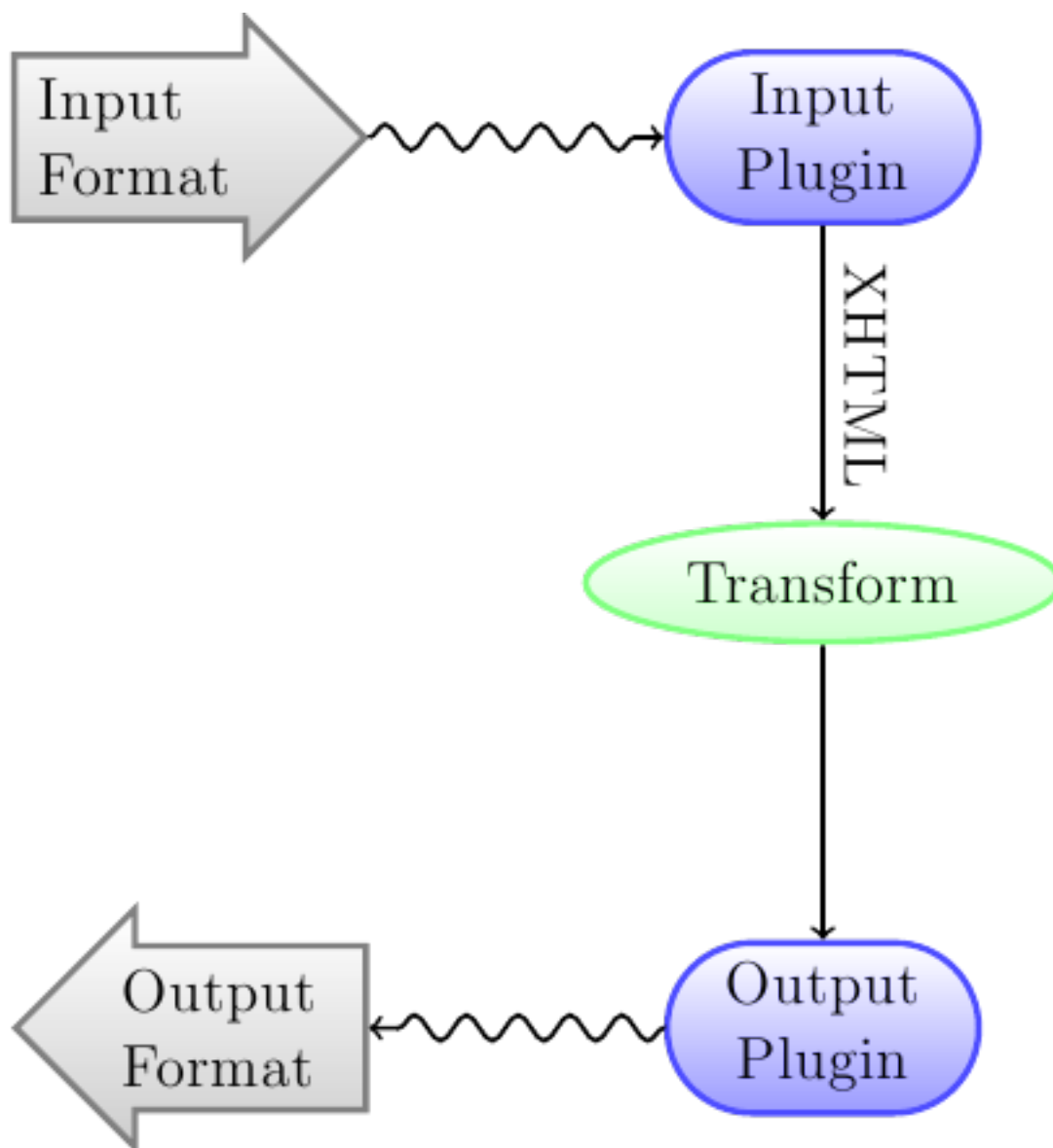


Contenuti

- *Introduzione* (pagina 55)
- *Aspetto* (pagina 56)
- *Impostazioni di pagina* (pagina 59)
- *Elaborazione euristica* (pagina 59)
- *Cerca e sostituisci* (pagina 61)
- *Rilevazione struttura* (pagina 61)
- *Indice* (pagina 62)
- *Usare immagini come titolo di capitolo nella conversione di documenti sorgente HTML* (pagina 64)
- *Usare gli attributi dei tag per fornire il testo delle voci dell'Indice* (pagina 64)
- *Come vengono impostate e salvate le opzioni di conversione* (pagina 64)
- *Suggerimenti per formati specifici* (pagina 65)

4.1 Introduzione

La prima cosa da comprendere è che la conversione è concepita come un sistema a cascata che schematicamente si presenta così:



Il formato in ingresso viene inizialmente convertito in XHTML dall' *Input plugin* appropriato. Questo XHTML è quindi *trasformato*. Nell'ultima fase l'XHTML elaborato viene convertito nel formato di uscita desiderato dall' *Output plugin* specifico. Alcuni formati in ingresso si convertono meglio di altri e questo può incidere notevolmente sui risultati, una lista dei migliori formati di input è disponibile [qui](#) (pagina 122).

Tutto il lavoro si svolge nella fase di modifica dell' XHTML in uscita. Ci sono varie operazioni da fare, ad esempio si possono inserire i metadata del libro come pagina iniziale, riconoscere l'intestazione dei capitoli e creare automaticamente una tavola dei contenuti «TOC», regolare la grandezza del font in modo proporzionale, ecc... È importante ricordare che tutte le operazioni di modifica vengono effettuate dall' *Input plugin* sull' XHTML e non sul file d'origine. Per fare un esempio: se chiedi a Calibre di convertire un RTF in EPUB, questi convertirà prima l'RTF in XHTML internamente e quindi, dopo aver applicato le modifiche desiderate, verrà creato l'EPUB tramite l' *Output plugin* generando automaticamente i metadata, la TOC ecc..

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



. Just specify the path to a folder for the

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

Tabella 1: Fasi del processo di conversione

Car-tella	Descrizione
input	Contiene l'HTML in uscita prodotto dal Plugin di Input. Usala per il debug del Plugin di Input.
par-sed	Il risultato di pre-elaborazione e conversione in XHTML del risultato del Plugin di Input. Usalo per il debug del rilevamento struttura.
struc-ture	Dopo il rilevamento della struttura ma prima dell'appiattimento del CSS e della conversione della dimensione carattere. Usala per il debug di dimensione carattere e trasformazioni CSS.
pro-ces-sed	Appena prima che l'e-book sia passato al Plugin di Output. Usala per il debug di quest'ultimo passo.

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

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

4.2 Aspetto

Contenuti

- *Caratteri* (pagina 57)
- *Testo* (pagina 58)
- *Disposizione* (pagina 58)
- *Stile* (pagina 58)
- *Trasforma stili* (pagina 59)
- *Trasforma HTML* (pagina 59)

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

4.2.1 Caratteri

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

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

Normally, calibre will automatically choose a base font size appropriate to the output profile you have chosen (see [Impostazioni di pagina](#) (pagina 59)). However, you can override this here in case the default is not suitable for you.

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

Illustriamo con un esempio. Supponiamo che il documento di partenza che stiamo convertendo sia stato prodotto da una persona con una vista eccellente e che abbia una dimensione base dei caratteri di 8pt. Ciò significa che la maggior parte del testo del documento ha una dimensione di 8pt, mentre le intestazioni sono leggermente più grandi (ad esempio 10 e 12pt) e le note a piè di pagina sono leggermente più piccole (6pt). Ora, se utilizziamo le seguenti impostazioni:

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

Il documento di output avrà una dimensione di base di 12pt, titoli di 14 e 16pt e note a piè di pagina di 8pt. Supponiamo ora di voler far risaltare maggiormente il titolo più grande e di rendere un po' più grandi anche le note a piè di pagina. Per ottenere questo risultato, l'opzione del font deve essere cambiata in:

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

I titoli più grandi diventeranno ora di 18pt, mentre le note a piè di pagina diventeranno di 9pt. È possibile giocare con queste impostazioni per cercare di capire quale sia quella ottimale per voi utilizzando la procedura guidata di ridimensionamento dei caratteri, a cui si accede facendo clic sul piccolo pulsante accanto all'impostazione *Tasto dimensione carattere*.

È inoltre possibile disattivare tutti i ridimensionamenti delle dimensioni dei caratteri durante la conversione, se si desidera conservare le dimensioni dei caratteri nel documento di input.

A related setting is *Line height*. Line height controls the vertical height of lines. By default, (a line height of 0), no manipulation of line heights is performed. If you specify a non-default value, line heights will be set in all locations that don't specify their own line heights. However, this is something of a blunt weapon and should be used sparingly. If you want to adjust the line heights for some section of the input, it's better to use the [Extra CSS](#) (pagina 58).

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

4.2.2 Testo

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

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

Finally, there is *Input character encoding*. Older documents sometimes don't specify their character encoding. When converted, this can result in non-English characters or special characters like smart quotes being corrupted. calibre tries to auto-detect the character encoding of the source document, but it does not always succeed. You can force it to assume a particular character encoding by using this setting. *cp1252* is a common encoding for documents produced using Windows software. You should also read *Come posso convertire il mio file contenente caratteri non inglesi o virgolette intelligenti?* (pagina 122) for more on encoding issues.

4.2.3 Disposizione

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

If you want to remove the spacing between all paragraphs, except a select few, don't use these options. Instead add the following CSS code to *Extra CSS* (pagina 58):

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

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

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

4.2.4 Stile

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

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

o se si vuole cambiare l'indentazione di tutti i paragrafi:

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

Extra CSS è un'opzione molto potente, ma è necessario conoscere il funzionamento dei CSS per poterla utilizzare al meglio. Si può usare l'opzione pipeline di debug descritta sopra per vedere quali CSS sono presenti nel documento di input.

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

4.2.5 Trasforma stili

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

4.2.6 Trasforma HTML

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

4.3 Impostazioni di pagina

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

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

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

4.4 Elaborazione euristica

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

Abilita l'elaborazione euristica

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

Unwrap lines

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

Line-unwrap factor

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

Rileva e qualifica titoli e sottotitoli di capitolo non formattati

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

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

Le intestazioni inserite non sono formattate, per applicare la formattazione, utilizzare l'opzione *Extra CSS* nelle impostazioni di conversione Look and Feel. Per esempio, per centrare i tag di intestazione, usare il seguente:

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

Renumber sequences of <h1> or <h2> tags

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

Elimina le righe vuote tra i paragrafi

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

Assicurati che i salti di scena siano formattati in modo coerente

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

Sostituisci i salti di scena

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

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

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

```

```

Esempio di regola orizzontale con stili:

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

Rimuovi trattini superflui

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

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

Applica il corsivo a parole e schemi comuni

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

Sostituisci i rientri forzati con rientri CSS

Alcuni documenti seguono una convenzione che costruisce i rientri usando spazi unificatori (che non vanno a capo). Quando questa opzione è abilitata calibre proverà a rilevare questa formattazione e a convertirla in un rientro CSS del 3%.

4.5 Cerca e sostituisci

Queste opzioni sono utili soprattutto per la conversione di documenti PDF o per le conversioni OCR, ma possono anche essere utilizzate per risolvere molti problemi specifici del documento. Ad esempio, alcune conversioni possono lasciare intestazioni e piè di pagina nel testo. Queste opzioni utilizzano espressioni regolari per cercare di individuare intestazioni, piè di pagina o altro testo arbitrario e rimuoverli o sostituirli. Ricordate che operano sull’XHTML intermedio prodotto dalla pipeline di conversione. È disponibile una procedura guidata che consente di personalizzare le espressioni regolari per il documento. Fare clic sulla bacchetta magica accanto alla casella di espressione e fare clic sul pulsante “Prova” dopo aver composto l’espressione di ricerca. Le corrispondenze corrette saranno evidenziate in Giallo.

The search works by using a Python regular expression. All matched text is simply removed from the document or replaced using the replacement pattern. The replacement pattern is optional, if left blank then text matching the search pattern will be deleted from the document. You can learn more about regular expressions and their syntax at [Tutto sull'utilizzo delle espressioni regolari in calibre](#) (pagina 205).

4.6 Rilevazione struttura

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

4.6.1 Capitoli e interruzioni di pagina

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

calibre uses *XPath*, a powerful language to allow the user to specify chapter boundaries/page breaks. XPath can seem a little daunting to use at first, fortunately, there is a [XPath tutorial](#) (pagina 151) in the User Manual. Remember that Structure detection operates on the intermediate XHTML produced by the conversion pipeline. Use the debug option described in the [Introduzione](#) (pagina 55) to figure out the appropriate settings for your book. There is also a button for a XPath wizard to help with the generation of simple XPath expressions.

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

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

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

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

L'impostazione predefinita per il rilevamento delle interruzioni di pagina è:

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

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

Nota: Le espressioni predefinite possono cambiare a seconda del formato di input che si sta convertendo.

4.6.2 Altro

In questa sezione sono presenti altre opzioni.

Insert metadata as page at start of book

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

Rimuovi la prima immagine

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

4.7 Indice

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

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

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

The default way that the creation of the auto generated Table of Contents works is that, calibre will first try to add any detected chapters to the generated table of contents. You can learn how to customize the detection of chapters in the [Rilevazione struttura](#) (pagina 61) section above. If you do not want to include detected chapters in the generated table of contents, check the *Do not add detected chapters* option.

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

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

```
Next|Previous
```

Le opzioni *Level 1,2,3 TOC* consentono di creare un sofisticato indice multilivello. Sono espressioni XPath che corrispondono a tag nell'XHTML intermedio prodotto dalla pipeline di conversione. Si veda [Introduzione](#) (pagina 55) per sapere come accedere a questo XHTML. Leggere anche [Guida di XPath](#) (pagina 151), per imparare a costruire le espressioni XPath. Accanto a ogni opzione c'è un pulsante che lancia una procedura guidata per aiutare a creare le espressioni XPath di base. Il semplice esempio che segue illustra l'uso di queste opzioni.

Supponiamo di avere un documento di input che risulta in XHTML come questo:

```
<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>
```

Quindi, impostiamo le opzioni come:

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

Si otterrà così un indice a due livelli, generato automaticamente, che avrà l'aspetto di:

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

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

4.8 Usare immagini come titolo di capitolo nella conversione di documenti sorgente HTML

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

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

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

4.9 Usare gli attributi dei tag per fornire il testo delle voci dell'Indice

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

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

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

4.10 Come vengono impostate e salvate le opzioni di conversione

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

You can also change settings in the conversion dialog for each book conversion. When you convert a book, calibre remembers the settings you used for that book, so that if you convert it again, the saved settings for the individual book will take precedence over the defaults set in *Preferences*. You can restore the individual settings to defaults by using the *Restore defaults* button in the individual book conversion dialog. You can remove the saved settings for a group of books

by selecting all the books and then clicking the *Edit metadata* button to bring up the bulk metadata edit dialog, near the bottom of the dialog is an option to remove stored conversion settings.

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

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

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

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

4.11 Suggerimenti per formati specifici

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

4.11.1 Conversione di documenti Microsoft Word

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

Nota: There is a [demo .docx file](https://calibre-ebook.com/downloads/demos/demo.docx)²⁹ that demonstrates the capabilities of the calibre conversion engine. Just download it and convert it to EPUB or AZW3 to see what calibre can do.

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

Vecchi file .doc

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

Un'altra alternativa è quella di usare il programma gratuito LibreOffice. Apri il tuo file .doc in LibreOffice e salvalo come .docx, che può essere convertito direttamente in calibre.

²⁹ <https://calibre-ebook.com/downloads/demos/demo.docx>

4.11.2 Conversione di documenti TXT

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

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

Paragraph style: Auto

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

Paragraph style: Block

Assumes one or more blank lines are a paragraph boundary:

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

Paragraph style: Single

Assume che ogni riga sia un paragrafo:

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

Paragraph style: Print

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

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

Paragraph style: Unformatted

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

Formatting style: Auto

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

Formatting style: Heuristic

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

Formatting style: Markdown

calibre also supports running TXT input through a transformation preprocessor known as Markdown. Markdown allows for basic formatting to be added to TXT documents, such as bold, italics, section headings, tables, lists, a Table of Contents, etc. Marking chapter headings with a leading # and setting the chapter XPath detection expression to «//h:h1» is the easiest way to have a proper table of contents generated from a TXT document. You can learn more about the Markdown syntax at [daringfireball](https://daringfireball.net/projects/markdown/syntax)³⁰.

³⁰ <https://daringfireball.net/projects/markdown/syntax>

Formatting style: None

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

4.11.3 Conversione di documenti PDF

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

Also, they often have headers and footers as part of the document that will become included with the text. Use the *Search and replace* panel to remove headers and footers to mitigate this issue. If the headers and footers are not removed from the text it can throw off the paragraph unwrapping. To learn how to use the header and footer removal options, read [Tutto sull'utilizzo delle espressioni regolari in calibre](#) (pagina 205).

Some limitations of PDF input are:

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

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

4.11.4 Collezioni di fumetti

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

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

Il file .cbc conterrà:

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

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

4.11.5 Esempio di formattazione avanzata EPUB

Various advanced formatting for EPUB files is demonstrated in this [demo file](#)³¹. The file was created from hand coded HTML using calibre and is meant to be used as a template for your own EPUB creation efforts.

The source HTML it was created from is available [demo.zip](#)³². The settings used to create the EPUB from the ZIP file are:

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

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

4.11.6 Conversione di documenti ODT

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

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

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

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

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

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

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

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

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

³² <https://calibre-ebook.com/downloads/demos/demo.zip>

4.11.7 Conversione in PDF

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

Intestazioni e piè di pagina

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- `_TOTAL_PAGES_` - total number of pages in the PDF file, useful for implementing a progress counter, for example.

- `_TOP_LEVEL_SECTION_PAGES_` - total number of pages in the current top level section
- `_TOP_LEVEL_SECTION_PAGENUM_` - the page number of the current page within the current top level section

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

Printable Table of Contents

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

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

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

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

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

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

Custom page margins for individual HTML files

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

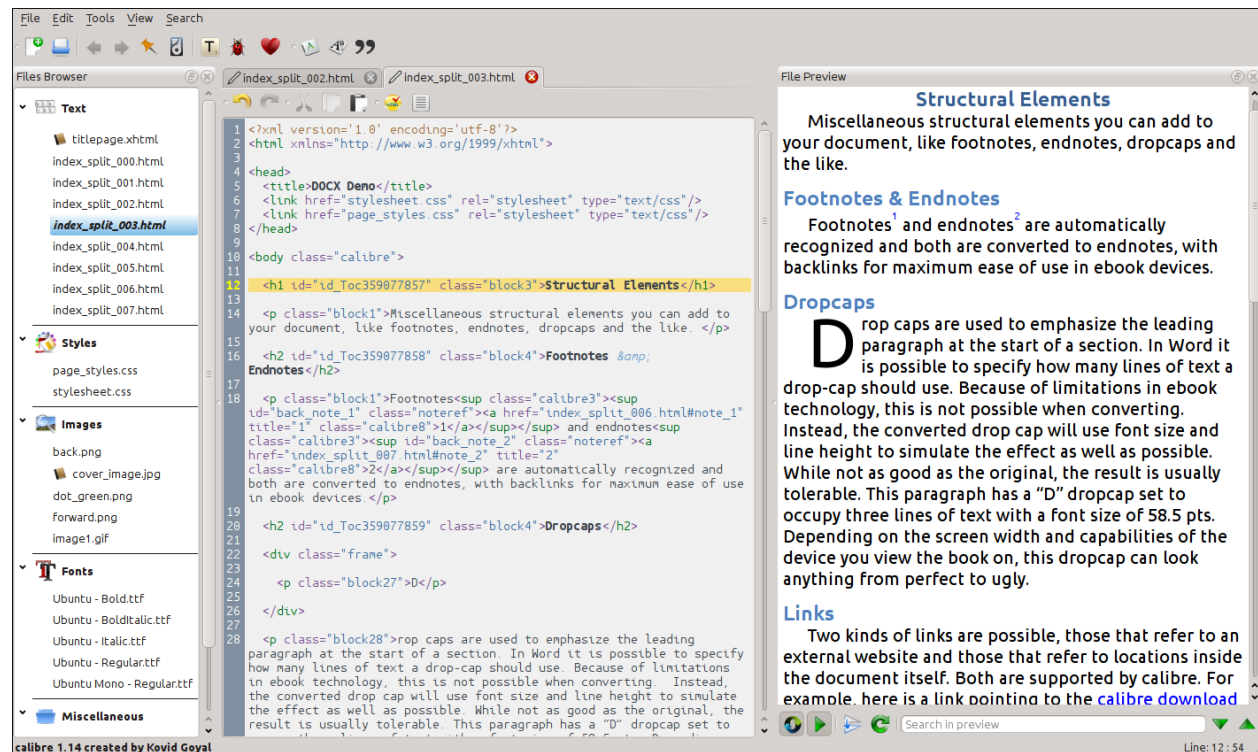
```
<style>
@page {
    margin-left: 10pt;
    margin-right: 10pt;
    margin-top: 10pt;
    margin-bottom: 10pt;
}
</style>
```

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

Modifica degli e-book

calibre contiene un editor di e-book integrato che può essere utilizzato per modificare libri nei formati EPUB e AZW3 (Kindle). L'editor mostra il codice HTML e CSS utilizzato all'interno dei file del libro, insieme ad una anteprima in tempo reale che si aggiorna mentre effettui le modifiche. Contiene inoltre diversi strumenti automatici per effettuare operazioni comuni di pulizia e riparazione.

Puoi utilizzare questo editor facendo click con il tasto destro su ogni libro contenuto in calibre e selezionando *Modifica libro*.



Contenuti

- *Flusso di lavoro di base* (pagina 73)
- *Il navigatore dei file* (pagina 75)
 - *La rinomina dei file* (pagina 76)
 - *L'unione dei file* (pagina 76)
 - *Cambio dell'ordine dei file di testo* (pagina 77)
 - *Marcare la copertina* (pagina 77)
 - *Eliminare i file* (pagina 77)
 - *Esportare i file* (pagina 77)
 - *Aggiunta di nuove immagini/caratteri/ecc. o creazione di nuovi file vuoti* (pagina 77)
 - *Sostituire i file* (pagina 78)
 - *Collegamento efficiente dei fogli di stile ai file HTML* (pagina 78)
- *Cerca e sostituisci* (pagina 78)
 - *Ricerche salvate* (pagina 78)
 - *Modalità funzione* (pagina 79)
 - *Ricerca senza considerare i tag HTML* (pagina 79)
- *Strumenti automatizzati* (pagina 79)
 - *Modifica dell'Indice* (pagina 79)
 - *Controllare il libro* (pagina 80)
 - *L'aggiunta di una copertina* (pagina 81)
 - *Integrazione dei caratteri referenziati* (pagina 81)
 - *Riduzione dei caratteri integrati* (pagina 81)
 - *Miglioramento della punteggiatura* (pagina 82)
 - *Transforming CSS properties* (pagina 82)
 - *Rimozione delle regole CSS non utilizzate* (pagina 82)
 - *Correzione del codice HTML* (pagina 82)
 - *Migliorare l'aspetto dei file* (pagina 82)
 - *Inserimento di un Indice in linea* (pagina 83)
 - *Impostare le semantiche* (pagina 83)
 - *Filtrare le informazioni di stile* (pagina 83)
 - *Aggiornamento dei contenuti interni del libro* (pagina 83)
- *Punti di controllo* (pagina 83)
- *Il pannello di anteprima in tempo reale* (pagina 85)
 - *Dividere i file HTML* (pagina 86)
- *Il pannello CSS Live* (pagina 87)

- *Strumenti vari* (pagina 88)
 - *La vista dell'Indice* (pagina 88)
 - *Controllo ortografico delle parole nel libro* (pagina 88)
 - *Inserimento di caratteri speciali* (pagina 90)
 - *La vista dell'ispettore del codice* (pagina 91)
 - *Controllo dei collegamenti esterni* (pagina 91)
 - *Scaricamento delle risorse esterne* (pagina 91)
 - *Organizzare i file in cartelle per tipo* (pagina 91)
 - *Importazione di file in altri formati e-book come EPUB* (pagina 91)
 - *Lo strumento Rapporti* (pagina 102)
- *Funzionalità speciali dell'editor del codice* (pagina 102)
 - *Evidenziazione della sintassi* (pagina 102)
 - *Aiuto legato al contesto* (pagina 103)
 - *Autocompletamento* (pagina 103)
 - *Frammenti* (pagina 103)

5.1 Flusso di lavoro di base

Nota: Un tour video dell'Editor di e-book di calibre è disponibile [qui](#)³³.

Quando apri un libro per la prima volta con lo strumento Modifica libro, comparirà una lista di file sulla sinistra. Questi sono file HTML individuali, fogli di stile, immagini, ecc. che costituiscono il contenuto del libro. Per iniziare a modificare, basta fare doppio clic su un file. Se vuoi fare cose più sofisticate di qualche piccolo ritocco, dovrai conoscere il [Tutorial HTML](#)³⁴ e il [Tutorial CSS](#)³⁵.

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

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

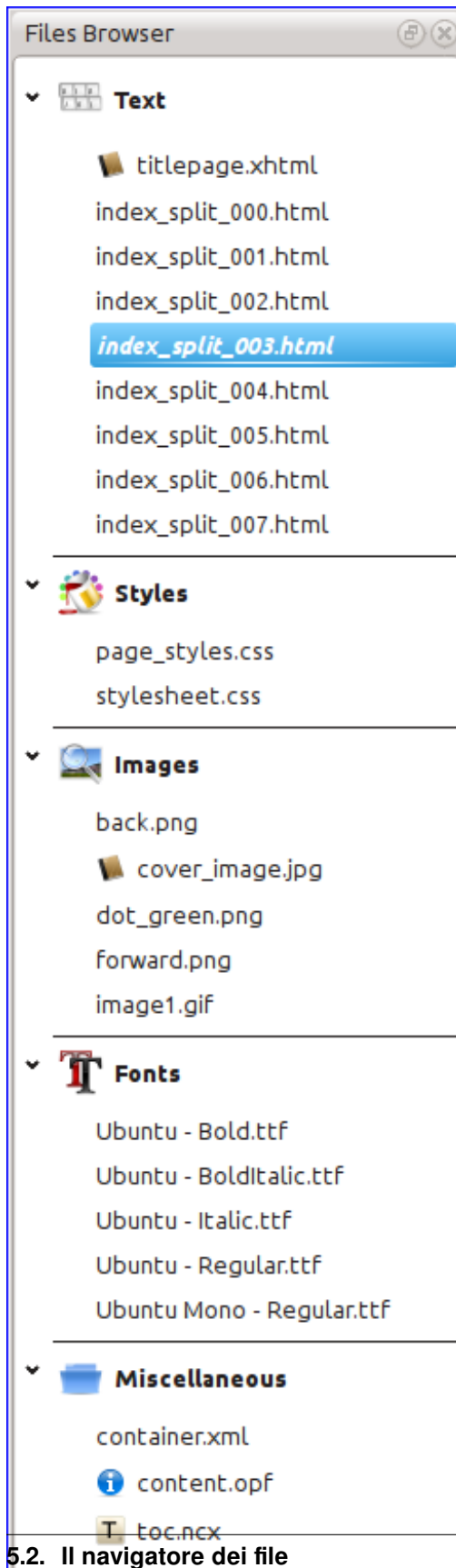
Questo è il flusso di lavoro di base per modificare libri – Apri un file, effettua le modifiche, visualizza l'anteprima e salva. Il resto di questo manuale illustrerà i vari strumenti e caratteristiche presenti per consentirti di effettuare operazioni specifiche in modo efficiente.

³³ <https://calibre-ebook.com/it/demo#tutorials>

³⁴ <http://html.net/tutorials/html/>

³⁵ <http://html.net/tutorials/css/>

5.2 Il navigatore dei file



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

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

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

È possibile svolgere diverse azioni sui singoli file, premendo con il tasto destro su di essi.

5.2.1 La rinomina dei file

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

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

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

5.2.2 L'unione dei file

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

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

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

5.2.3 Cambio dell'ordine dei file di testo

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 Marcare la copertina

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

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

5.2.5 Eliminare i file

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 Esportare i file

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 Aggiunta di nuove immagini/caratteri/ecc. o creazione di nuovi file vuoti

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

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

5.2.8 Sostituire i file

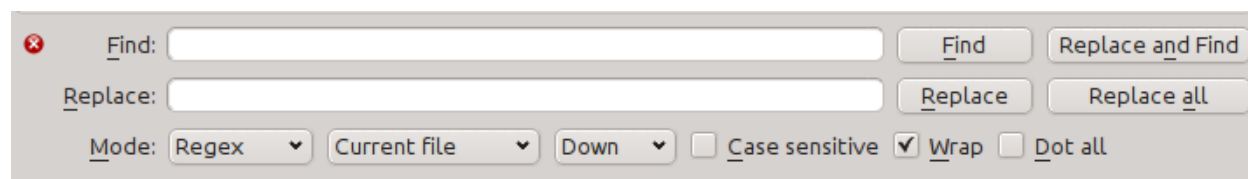
You can easily replace existing files in the book, by right clicking on the file and choosing replace. This will automatically update all links and references, in case the replacement file has a different name than the file being replaced.

5.2.9 Collegamento efficiente dei fogli di stile ai file 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 Cerca e sostituisci

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 *Tutto sull'utilizzo delle espressioni regolari in calibre* (pagina 205).



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

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

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

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

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

Nota: Remember, to harness the full power of search and replace, you will need to use regular expressions. See *Tutto sull'utilizzo delle espressioni regolari in calibre* (pagina 205).

5.3.1 Ricerche salvate

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

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

5.3.2 Modalità funzione

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 *Modalità Funzione per Cerca & sostituisci nell'Editor* (pagina 91).

5.3.3 Ricerca senza considerare i tag HTML

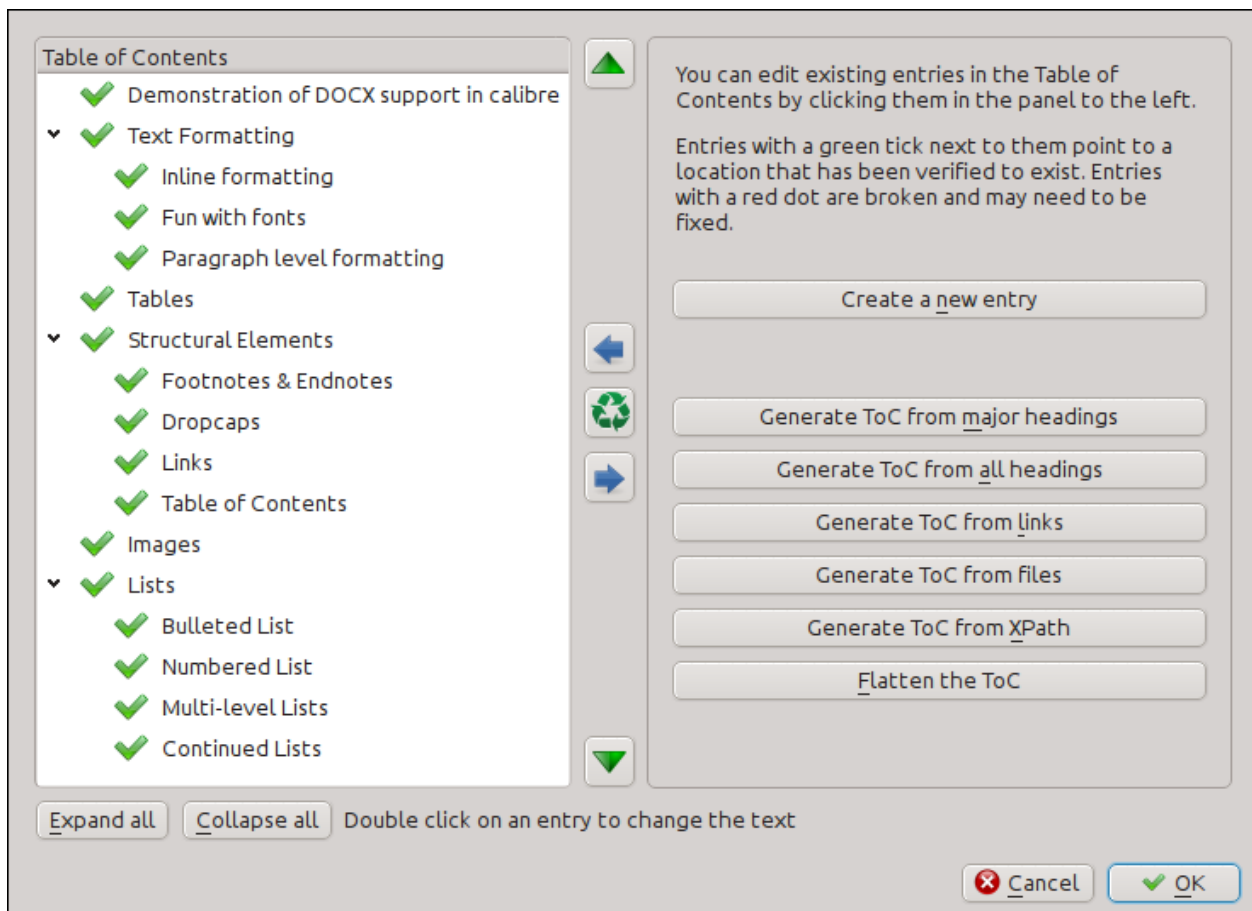
There is also a dedicated tool for searching for text, ignoring any HTML tags in between. For example, if the book has the HTML `Empahisis on a <i>word</i>`. you can search for `on a word` and it will be found even though there is an `<i>` tag in the middle. Use this tool via the *Search → Search ignoring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

5.4 Strumenti automatizzati

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

5.4.1 Modifica dell'Indice

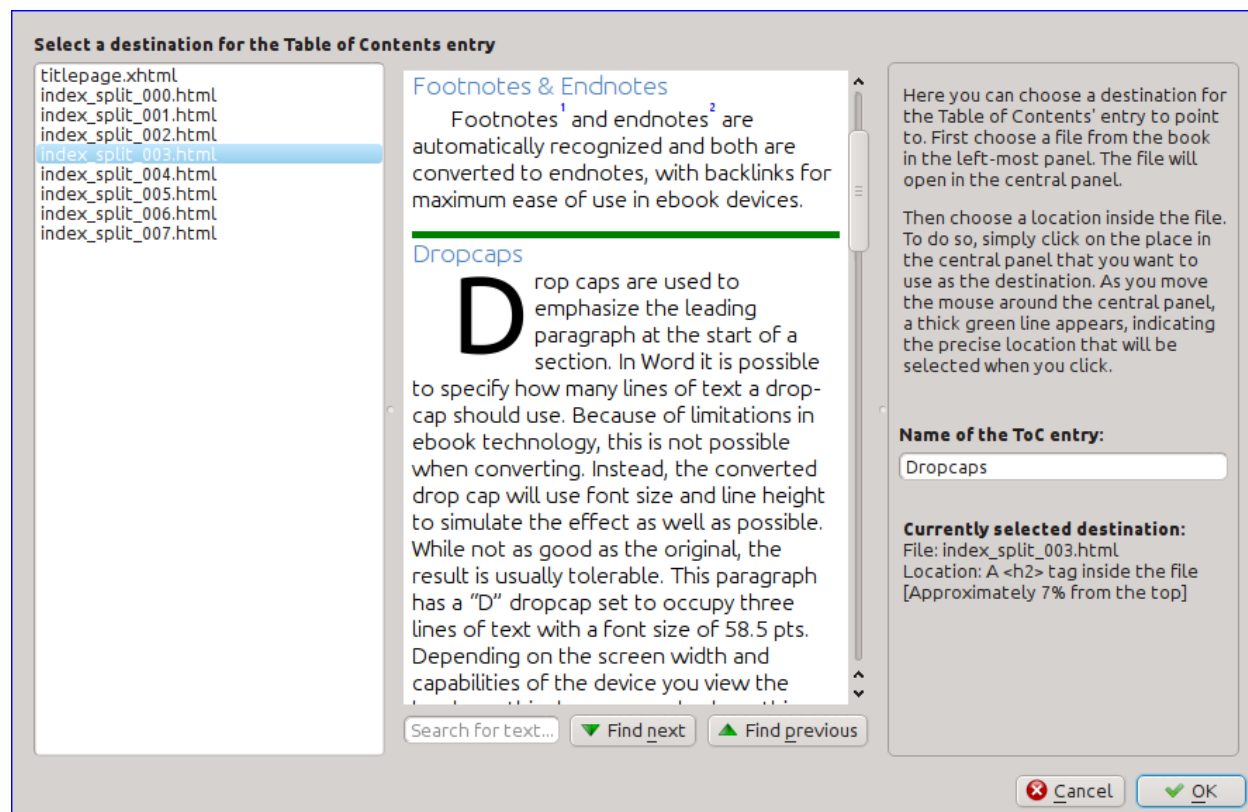
There is a dedicated tool to ease editing of the Table of Contents. Launch it with *Tools → Table of Contents → Edit Table of Contents*.



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

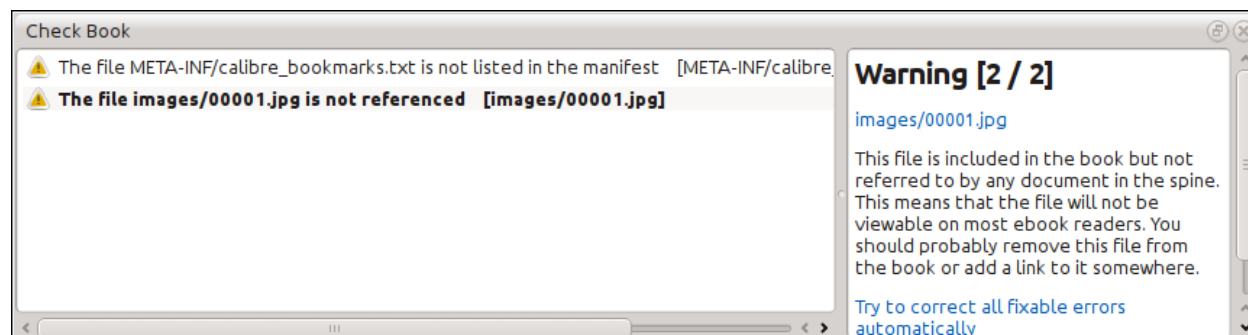
For books that do not have a pre-existing Table of Contents, the tool gives you various options to auto-generate a Table of Contents from the text. You can generate from the headings in the document, from links, from individual files and so on.

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



5.4.2 Controllare il libro

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



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

Some of the checks performed are:

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

5.4.3 L'aggiunta di una copertina

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 Integrazione dei caratteri referenziati

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 Riduzione dei caratteri integrati

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 Miglioramento della punteggiatura

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

5.4.7 Transforming CSS properties

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

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

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

Accessed via *Tools* → *Transform styles*.

5.4.8 Rimozione delle regole CSS non utilizzate

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

5.4.9 Correzione del codice HTML

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

5.4.10 Migliorare l'aspetto dei file

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 <div> tags will be beautified. This can sometimes mean that a particular file will not be affected by beautify as it has no suitable block level tags. In such cases you can try different beautification tools, that are less careful, for example: [HTML Tidy](https://infohound.net/tidy/)³⁶.

³⁶ <https://infohound.net/tidy/>

5.4.11 Inserimento di un Indice in linea

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

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

5.4.12 Impostare le semantiche

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

5.4.13 Filtrare le informazioni di stile

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 Aggiornamento dei contenuti interni del libro

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 Punti di controllo

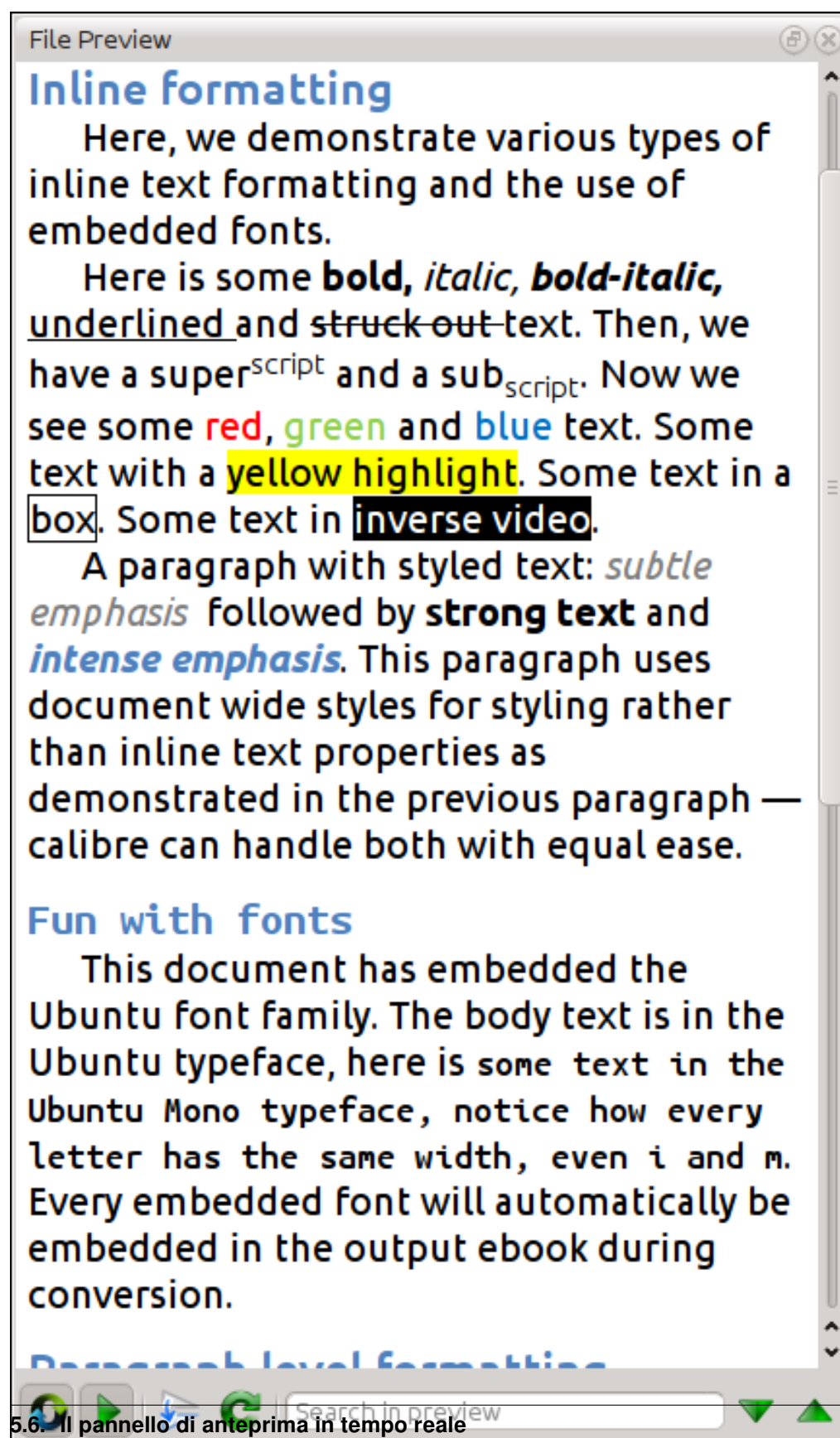
Checkpoints are a way to mark the current state of the book as «special». You can then go on to do whatever changes you want to the book and if you don't like the results, return to the checkpointed state. Checkpoints are automatically created every time you run any of the automated tools described in 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 *Confronto degli e-book* (pagina 113) 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 Il pannello di anteprima in tempo reale



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

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

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

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 Dividere i file HTML

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

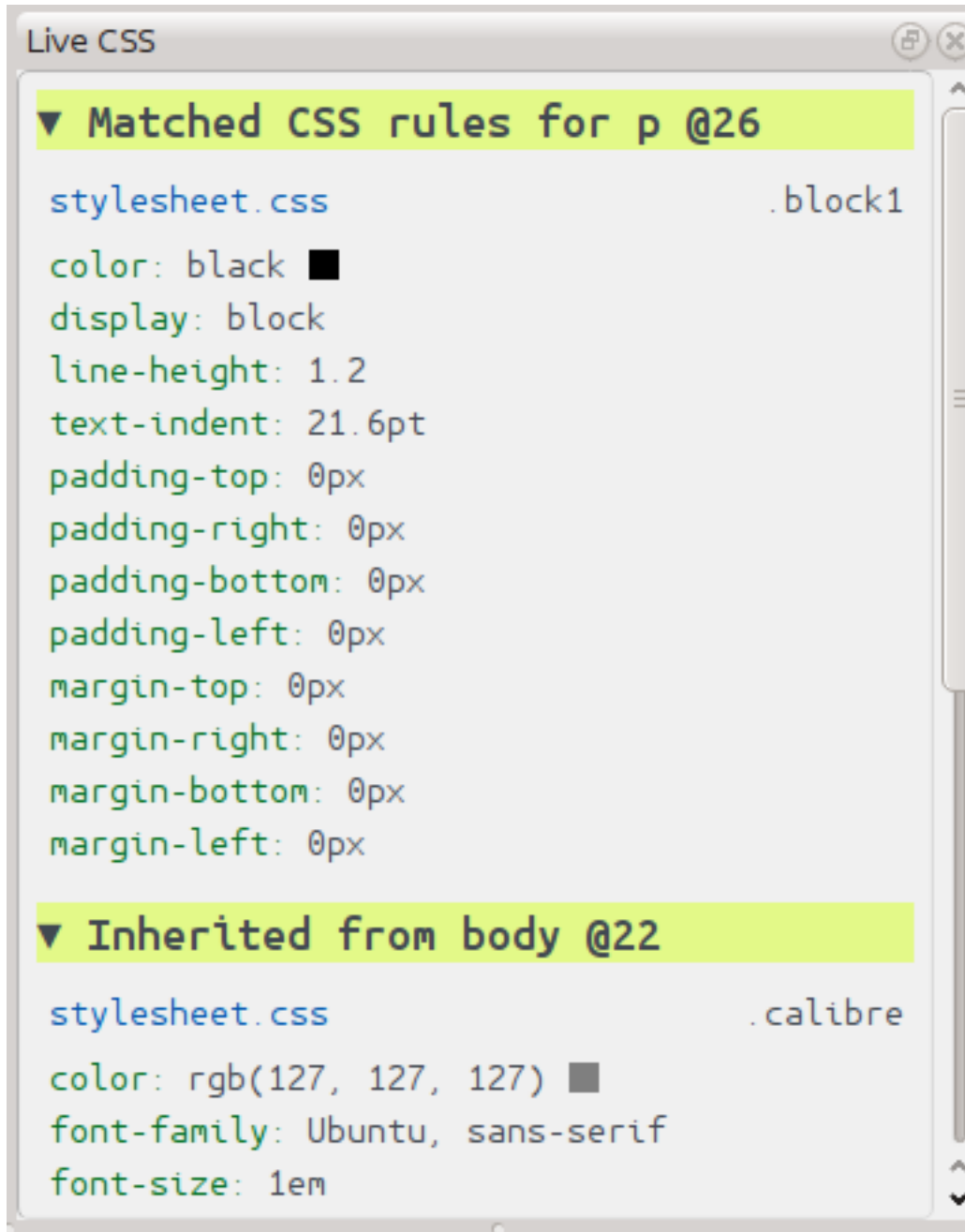


click the *Split mode* button under the preview panel. Then simply move your mouse to the place where you want to split the file and click. A thick green line will show you exactly where the split will happen as you move your mouse. Once you have found the location you want, simply click and the split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



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 Strumenti vari

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

5.8.1 La vista dell'Indice

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 Controllo ortografico delle parole nel libro

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

Filter the list of words

Word	Count	Language
DOCX	16	English
dropcap	2	English
Dropcaps	3	English
dropcaps	2	English
ebook	11	English
ebook.com	1	English
ebooks	3	English
EPUB	2	English
etc	1	English
Goyal	2	English
gray	1	English
hyperlinks	1	English
i	1	English
i.e	1	English
inline	2	English
Inline	5	English
Kovid	2	English

Ignore inline

Add to dictionary:

Default

Show next occurrence

Change selected word to:

online

- online
- incline
- in-line
- in line
- mainline
- inlier
- unlined
- newline
- inland
- on-line

Misspelled words: 30 Total words: 571 ☒ Show only misspelled words

Refresh Close

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: Per escludere un singolo file dal controllo ortografico eseguito dallo strumento omonimo, puoi usare il pulsante *Escludi file* o aggiungere il seguente commento appena sotto al tag di apertura nel file:

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

Aggiunta di nuovi dizionari

The spelling checker comes with builtin dictionaries for the English and Spanish languages. You can install your own dictionaries via *Preferences* → *Editor* → *Manage spelling dictionaries*. The spell checker can use dictionaries from the LibreOffice program (in the .oxf format). You can download these dictionaries from [The LibreOffice Extensions repository](https://extensions.libreoffice.org/?Tags%5B%5D=50)³⁷.

5.8.3 Inserimento di caratteri speciali

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

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

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

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

³⁷ <https://extensions.libreoffice.org/?Tags%5B%5D=50>

5.8.4 La vista dell'ispettore del codice

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 Controllo dei collegamenti esterni

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 Scaricamento delle risorse esterne

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 Organizzare i file in cartelle per 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 Importazione di file in altri formati e-book come 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*.

Modalità Funzione per Cerca & sostituisci nell'Editor

The *Search & replace* tool in the editor support a *function mode*. In this mode, you can combine regular expressions (see *Tutto sull'utilizzo delle espressioni regolari in calibre* (pagina 205)) with arbitrarily powerful Python functions to do all sorts of advanced text processing.

In the standard *regex* mode for search and replace, you specify both a regular expression to search for as well as a template that is used to replace all found matches. In function mode, instead of using a fixed template, you specify an arbitrary function, in the *Python programming language*³⁸. This allows you to do lots of things that are not possible with simple templates.

Le tecniche per usare la modalità funzione e la sintassi saranno descritte attraverso esempi, mostrandoti come creare funzioni per effettuare compiti sempre più complessi.

³⁸ <https://docs.python.org>

Correggere automaticamente le maiuscole delle intestazioni nel documento

Qui, useremo una delle funzioni integrate nell'editor per cambiare automaticamente la casella di tutto il testo all'interno dei tag di titolo per intitolare la casella:

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

Come funzione, scegli semplicemente la funzione integrata *Title-case text (ignore tags)*. La funzione cambierà i titoli che appaiono come: `<h1>un TITOLO</h1>` in `<h1>Un TitloLo</h1>`. Funzionerà anche se ci sono altri tag HTML all'interno dei tag di tipo titolo.

La tua prima funzione personalizzata - miglioramento dei trattini

Il vero potere della modalità funzione sta nel fatto di essere in grado di creare le tue personali funzioni per elaborare il testo in modi arbitrari. Lo strumento Punteggiatura Furba nell'editor lascia i trattini individuali da soli, in modo che tu possa usare questa funzione per sostituirli con trattini di enfasi.

Per creare una nuova funzione, fai clic semplicemente sul pulsante :guilabel:`Crea/Modifica` per creare una nuova funzione e copia il codice Python qui sotto.

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

Every *Search & replace* custom function must have a unique name and consist of a Python function named `replace`, that accepts all the arguments shown above. For the moment, we won't worry about all the different arguments to `replace()` function. Just focus on the `match` argument. It represents a match when running a search and replace. Its full documentation is available [here](https://docs.python.org/library/re.html#match-objects)³⁹. `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.

Usa questa funzione con l'espressione regolare di ricerca:

`>[^<>]+<`

And it will replace all hyphens with em-dashes, but only in actual text and not inside HTML tag definitions.

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

The power of function mode - using a spelling dictionary to fix mis-hyphenated words

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

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

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

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

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

Usa questa funzione con la stessa espressione di ricerca di prima, vale a dire:

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

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.

Numerazione automatica delle sezioni

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'
```

Usala con l'espressione di ricerca:

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

Posiziona il cursore sopra il file e fai clic su *Sostituisci tutto*.

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 *Scegliere l'ordine dei file quando si usano diversi file HTML* (pagina 97) for details.

Genera automaticamente un indice

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

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

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

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

(continues on next page)

(continua dalla pagina precedente)

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

E usala con l'espressione di ricerca:

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

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

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

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

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

L'API per la modalità funzione

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

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

Quando si usa la funzione cerca/sostituisci, per ogni riscontro, viene richiamata la funzione `replace()`, deve restituire la stringa sostitutiva. Se non avviene nessuna sostituzione, deve restituire `match.group()` che è la stringa originale. I vari argomenti della funzione `replace()` sono descritti qui sotto.

L'argomento "corrispondenza"

The `match` argument represents the currently found match. It is a [Python Match object](#)⁴⁰. 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.

L'argomento `number`

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.

⁴⁰ <https://docs.python.org/library/re.html#match-objects>

L'argomento `nome_file`

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.

L'argomento `metadata`

This represents the metadata of the current book, such as title, authors, language, etc. It is an object of class `calibre.ebooks.metadata.book.base.Metadata` (pagina 201). Useful attributes include, `title`, `authors` (a list of authors) and `language` (the language code).

L'argomento `dizionari`

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.

L'argomento `dati`

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.

L'argomento `funzioni`

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):
    ...
```

Poi, in un'altra funzione, puoi accedere alla funzione `utility()` in questo modo:

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

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

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

(continues on next page)

(continua dalla pagina precedente)

```
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']
    ...
```

Debug delle proprie funzioni

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.

Scegliere l'ordine dei file quando si usano diversi file HTML

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

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

replace.file_order = 'spine'
```

`file_order` 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
```

Più esempi

More useful examples, contributed by calibre users, can be found in the [calibre E-book editor forum](https://www.mobileread.com/forums/showthread.php?t=237181)⁴¹.

Frammenti

L'editor di e-book di calibre supporta l'utilizzo dei «frammenti». Un frammento è un segmento di testo che è riutilizzato spesso o contiene parecchio testo ridondante. L'editor ti permette di inserire frammenti premendo solamente alcuni tasti. Per esempio, supponi di dover inserire molte volte dei tag link quando modifichi file HTML, puoi semplicemente scrivere `<a` nell'editor e premere `Control+J`. L'editor espanderà il testo in:

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

Non solo, la parola `filename` sarà selezionata e il cursore posizionato su di essa, in modo che tu possa facilmente digitare il nome effettivo del file usando l'ingegnosa funzionalità di *Autocompletamento* (pagina 103). E una volta che hai finito di digitare il nome del file, premi di nuovo `Control+J` e il cursore salterà alla posizione tra i tag `<a>`, in modo che tu possa facilmente inserire il testo per il link.

Il sistema dei frammenti dell'editor è molto sofisticato; ci sono alcuni frammenti predefiniti e puoi crearne di nuovi che si adattino al tuo stile di modifica.

La seguente discussione sui frammenti predefiniti dovrebbe aiutare a illustrare le potenzialità del sistema dei frammenti.

Nota: Puoi anche usare i frammenti nelle caselle di testo del pannello *Cerca e sostituisci*, tuttavia i placeholder (l'uso di `Control+J` per i salti di cursore) non funzioneranno.

⁴¹ <https://www.mobileread.com/forums/showthread.php?t=237181>

I frammenti predefiniti

I frammenti predefiniti sono descritti di seguito. Tieni presente che puoi sovrascriverli creando i tuoi frammenti con lo stesso testo di attivazione.

Inserimento di un testo riempitivo [Lorem]

Il primo e più semplice frammento predefinito è utilizzato per inserire un testo riempitivo all'interno di un documento. Il testo riempitivo è preso da *De finibus bonorum et malorum*⁴² un trattato filosofico di Cicerone (tradotto in inglese). Per utilizzarlo digita semplicemente «Lorem» in un file HTML e premi `Control+J`. Il frammento sarà automaticamente sostituito da due paragrafi di riempitivo.

La definizione di questo frammento è molto semplice, il testo di attivazione è definito come «Lorem» e lo schema è definito semplicemente come il testo che deve essere inserito. Puoi facilmente personalizzarlo perché usi il testo riempitivo che preferisci.

Inserimento di un tag HTML autochiudente [<>]

Ora diamo un'occhiata a un semplice esempio del potente concetto dei *placeholder*. Supponiamo di voler inserire il tag autochiudente `<hr/>`: basta digitare `<>` e premere `Control+J`, l'editor espanderà il frammento in:

```
<|/>
```

Qui il simbolo `|` rappresenta la posizione corrente del cursore. Basta digitare `hr` e premere `Control+J` per muovere il cursore dopo la chiusura del tag. Questo frammento è definito come:

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

I placeholder sono semplicemente il simbolo del dollaro (\$) seguito da un numero. Quando il frammento viene espanso premendo `Control+J` il cursore è posizionato sul primo placeholder (cioè il placeholder con il numero più basso). Quando `Control+J` viene premuto nuovamente, il cursore salta al placeholder successivo (cioè il placeholder con il numero successivo più alto).

Inserimento di un tag HTML link [<a]

Tutti i tag HTML link condividono la stessa struttura. Hanno un attributo `href` e del testo tra i tag di apertura e di chiusura. Un frammento per inserirli in modo più efficiente ci introdurrà a nuove funzionalità dei placeholder. Per usare questo frammento digita semplicemente «<a» e premi `Control+J`. L'editor lo espanderà a:

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

Non solo, la parola `filename` sarà selezionata e il cursore posizionato su di essa, in modo che tu possa facilmente digitare il nome effettivo del file usando l'ingegnosa funzionalità di *Autocompletamento* (pagina 103). E una volta che hai finito di digitare il nome del file, premi di nuovo `Control+J` e il cursore salterà alla posizione tra i tag `<a>`, in modo che tu possa facilmente inserire il testo per il link. Dopo aver finito di inserire il testo, premi `Control+J` ancora una volta per muovere il cursore dopo il tag di chiusura. Questo frammento è definito come:

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

⁴² https://en.wikipedia.org/wiki/De_finibus_bonorum_et_malorum

Ci sono un paio di novità qui. Per prima cosa il placeholder `$1` è diventato più complesso: include ora del *testo predefinito* (la parola `filename`). Se un placeholder contiene del testo predefinito, il testo predefinito è sostituito per il placeholder quando il frammento viene espanso. Per di più quando salti ad un placeholder con del testo predefinito usando `Control+J`, il testo predefinito viene selezionato. In questo modo puoi usare il testo predefinito come promemoria per ricordarti di riempire parti importanti dello schema. Puoi specificare il testo predefinito per un placeholder usando la sintassi: `${:testo predefinito}`.

L'altra nuova caratteristica è che il secondo placeholder ha un asterisco a seguire (`${2*}`). Questo vuol dire che il testo (eventualmente) selezionato prima dell'espansione dello schema è sostituito per il placeholder. Per vedere questa caratteristica in azione, seleziona del testo nell'editor, premi `Control+J`, digita `<a` e premi nuovamente `Control+J`, lo schema sarà espanso a:

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

Inserimento di un tag HTML immagine [<i]

Questo è molto simile all'inserimento di un link HTML, come abbiamo appena visto. Ti permette di inserire rapidamente un tag `` e di saltare tra gli attributi `src` e `alt`:

```
Trigger: <i
Template: $3
```

Inserimento di un tag HTML arbitrario [<<]

Questo ti permette di inserire un tag HTML arbitrario e completo (o di racchiudere del testo selezionato in precedenza nel tag). Per usarlo digita semplicemente `<<` e premi `Control+J`. L'editor lo espanderà a:

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

Digita il nome del tag, per esempio `span` e premi `Control+J`, questo porterà a:

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

Puoi notare che il tag di chiusura contiene già al suo interno `span`. Questo è possibile grazie a un'altra caratteristica dei placeholder, il *rispecchiamento*. Rispecchiamento significa semplicemente che se specifichi il placeholder di esempio più volte nello stesso schema, al premere di `Control+J` la seconda posizione e tutte quelle successive saranno automaticamente riempite con qualunque testo tu abbia digitato nella prima posizione. La definizione per questo frammento è:

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

Come puoi vedere, il primo placeholder (`$1`) è stato specificato due volte, la seconda nel tag di chiusura, che semplicemente copierà qualunque testo tu scriva nel tag di apertura.

Inserimento di un tag HTML arbitrario con un attributo di classe [<c]

Questo è molto simile all'inserimento di tag arbitrari visto sopra, eccetto per il fatto che si assume che tu voglia specificare una classe per il tag:

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

Questo ti permetterà di inserire prima il nome del tag, `Control+J`, inserire il nome della classe, premere `Control+J`, inserire i contenuti del tag e premere `Control+J` un'ultima volta per saltare al di fuori del tag. Il tag di chiusura sarà riempito automaticamente.

Creazione di frammenti personalizzati

I frammenti sono così interessanti perché puoi crearne di tuoi per adattarli al tuo stile di modifica. Per creare frammenti personalizzati entra nell'editor e vai in *Modifica* → *Preferenze* → *Impostazioni dell'editor* → *Gestisci i frammenti*. Apparirà una finestra che ti aiuterà a creare facilmente i tuoi frammenti personalizzati. Basta fare clic sul pulsante *Aggiungi frammento* per aprire una finestra come questa:

Create a snippet

For help with snippets, see the [User Manual](#)

Name: The name of this snippet

Trigger: The text used to trigger this snippet

Template:

File types: ☒ All ☐ css ☐ html ☐ javascript ☐ text ☐ xml

Test:

OK Cancel

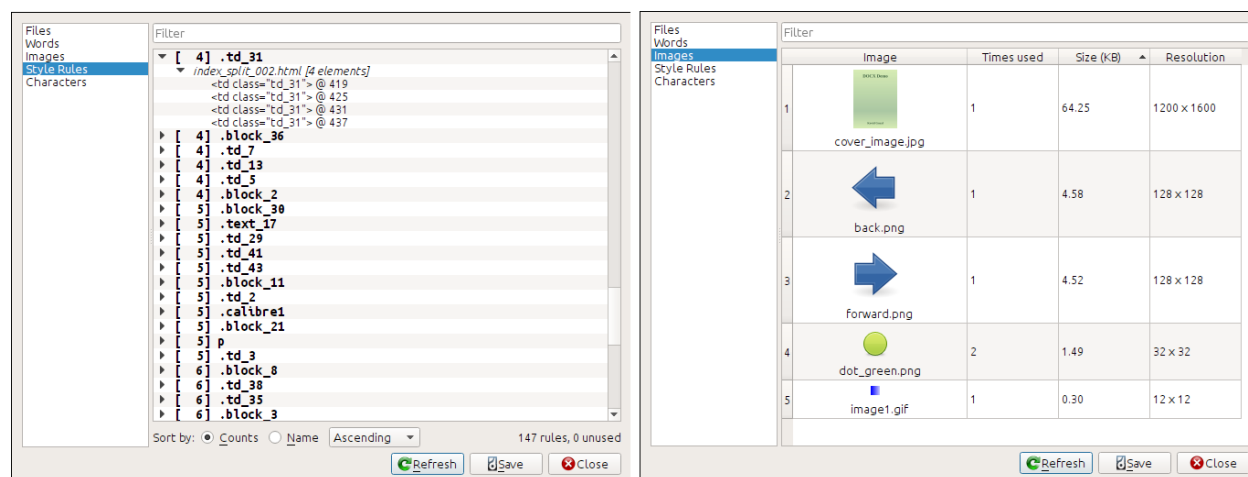
Per prima cosa dai un nome al tuo frammento, qualcosa di descrittivo che aiuti a identificare il frammento in futuro. Dopodiché specifica il testo di attivazione: è semplicemente il testo che devi digitare nell'editor prima di premere `Control+J` per far espandere il frammento.

Ora specifica lo schema per il frammento. Dovresti cominciare con uno degli esempi visti finora e modificarlo secondo le tue necessità. Infine specifica per quali tipi di file vuoi che il frammento sia attivo. In questo modo potrai avere frammenti diversi con lo stesso testo di attivazione che diano risultati differenti a seconda del tipo di file.

Il passo successivo è quello di provare i tuoi nuovi frammenti. Usa la casella *Prova* in fondo alla finestra. Digita il testo di attivazione e premi **Control+J** per far espandere il frammento e saltare tra i placeholder.

5.8.9 Lo strumento Rapporti

The editor includes a nice *Reports* tool (via *Tools* → *Reports*) that shows summaries of the files, images, links, words, characters and styles used in the book. Every line in the report is hot-linked. Double clicking a line jumps to the place in the book where that item is used or defined (as appropriate). For example, in the *Links* view, you can double click entries in the *Source* column to jump to where the link is defined and entries in the *Target* column to jump to where the link points.



5.9 Funzionalità speciali dell'editor del codice

The calibre HTML editor is very powerful. It has many features that make editing of HTML (and CSS) easier.

5.9.1 Evidenziazione della sintassi

The HTML editor has very sophisticated syntax highlighting. Features include:

- The text inside bold, italic and heading tags is made bold/italic
- As you move your cursor through the HTML, the matching HTML tags are highlighted, and you can jump to the opening or closing tag with the keyboard shortcuts **Ctrl+{** and **Ctrl+}**. Similarly, you can select the contents of a tag with **Ctrl+Alt+T** or **Ctrl+Shift+T**.
- Invalid HTML is highlighted with a red underline
- Spelling errors in the text inside HTML tags and attributes such as title are highlighted. The spell checking is language aware, based on the value of the lang attribute of the current tag and the overall book language.
- CSS embedded inside `<style>` tags is highlighted
- Special characters that can be hard to distinguish such as non-breaking spaces, different types of hyphens, etc. are highlighted.

- Links to other files in `<a>` tags, `` and `<link>` tags all have the filenames highlighted. If the filename they point to does not exist, the filename is marked with a red underline.

5.9.2 Aiuto legato al contesto

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 Autocompletamento

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 Frammenti

The calibre E-book editor supports *snippets*. A snippet is a piece of text that is either re-used often or contains a lot of redundant text. The editor allows you to insert a snippet with only a few key strokes. The snippets are very powerful, with many features, such as placeholders you can jump between, automatic mirroring of repeated text and so on. For more information, see *Frammenti* (pagina 98).

Il Server dei Contenuti di Calibre

The calibre *Content server* allows you to access your calibre libraries and read books directly in a browser on your favorite mobile phone or tablet device. As a result, you do not need to install any dedicated book reading/management apps on your phone. Just use the browser. The server downloads and stores the book you are reading in an off-line cache so that you can read it even when there is no internet connection.

Contenuti

- *Accesso al server dei contenuti da altri dispositivi* (pagina 106)
 - *Accesso al server da dispositivi nella rete di casa* (pagina 106)
 - *Accesso al server da qualsiasi luogo via internet* (pagina 107)
- *L'interfaccia del server* (pagina 107)
 - *L'elenco dei libri* (pagina 108)
 - *Il lettore di libri* (pagina 108)
- *Supporto dei browser* (pagina 108)
- *Abilitare il supporto per l'uso offline* (pagina 108)
- *Gestione degli utenti usando solamente la linea di comando* (pagina 109)
- *Integrazione del server dei contenuti di calibre dentro altri server* (pagina 109)
 - *Using a full virtual host* (pagina 109)
 - *Using a URL prefix* (pagina 110)
- *Creazione di un servizio per il server di calibre su un sistema Linux moderno* (pagina 111)

To start the server, click the *Connect/share* button and choose *Start Content server*. You might get a message from your computer's firewall or anti-virus program asking if it is OK to allow access to `calibre.exe`. Click the *Allow* or *OK* button. Then open a browser (preferably Chrome or Firefox) in your computer and type in the following address:

`http://127.0.0.1:8080`

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

Nota: The address used above `http://127.0.0.1:8080` will only work on the computer that is running calibre. To access the server from other computers/phones/tablets/etc. you will need to do a little more work, as described in the next section.

6.1 Accesso al server dei contenuti da altri dispositivi

There are two types of remote device access that you will typically need. The first, simpler kind is from within your home network. If you are running calibre on a computer on your home network and you have also connected your other devices to the same home network, then you should be easily able to access the server on those devices.

6.1.1 Accesso al server da dispositivi nella rete di casa

After starting the server in calibre as described above, click the *Connect/share* button again. Instead of the *Start Content server* action, you should see a *Stop Content server* action instead. To the right of this action will be listed an IP address and port number. These look like a bunch of numbers separated by periods. For example:

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

These numbers tell you what address to use to connect to the server in your devices. Following the example above, the address becomes:

```
http://192.168.1.5:8080
```

The first part of the address is always `http://` the next part is the IP address, which is the numbers before the comma and finally we have the port number which must be added to the IP address with a colon (:). If you are lucky, that should be all you need and you will be looking at the calibre libraries on your device. If not, read on.

Trouble-shooting the home network connection

If you are unable to access the server from your device, try the following steps:

1. Check that the server is running by opening the address `http://127.0.0.1:8080` in a browser running on the same computer as the server.
2. Check that your firewall/anti-virus is allowing connections to your computer on the port 8080 and to the calibre program. The easiest way to eliminate the firewall/anti-virus as the source of problems is to temporarily turn them both off and then try connecting. You should first disconnect from the internet, before turning off the firewall, to keep your computer safe.
3. Check that your device and computer are on the same network. This means they should both be connected to the same wireless router. In particular neither should be using a cellular or ISP provided direct-WiFi connection.
4. If you have non-standard networking setup, it might be that the IP address shown on the *Connect/share* menu is incorrect. In such a case you will have to figure out what the correct IP address to use is, yourself. Unfortunately, given the infinite diversity of network configurations possible, it is not possible to give you a roadmap for doing so.

5. If you have setup a username and password, first try it without that to see if it is causing issues. Some e-ink devices have browsers that do not handle authentication. You can sometimes workaround this by including the username and password in the URL, for example: `http://username:password@192.168.1.2:8080`.
6. If you are stuck, you can always ask for help in the [calibre user forums](#)⁴³.

6.1.2 Accesso al server da qualsiasi luogo via internet

Avvertimento: 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](#)⁴⁴ 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](#)⁴⁵ 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 L'interfaccia del server

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.

⁴³ <https://www.mobileread.com/forums/forumdisplay.php?f=166>

⁴⁴ <https://www.whatismyip.com/>

⁴⁵ <https://www.noip.com/free>

6.2.1 L'elenco dei libri

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 Il lettore di libri

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

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

6.3 Supporto dei browser

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 Abilitare il supporto per l'uso offline

Browser makers have been trying to force people to use SSL by disabling advanced features in their browsers for plain HTTP connections. One such casualty is offline support. So you may need to enable HTTPS on the server to get offline support working. In addition, in Firefox on Android, you will need to type `about:config` and search for `browser.tabs.useCache` and toggle it to `true`.

6.5 Gestione degli utenti usando solamente la linea di comando

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 Integrazione del server dei contenuti di calibre dentro altri server

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

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

6.6.1 Using a full virtual host

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

```
calibre-server
```

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

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

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

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

Or, for Apache:

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

(continues on next page)

(continua dalla pagina precedente)

```
<VirtualHost *:80>
    ServerName myserver.example.com
    AllowEncodedSlashes On
    ProxyPreserveHost On
    ProxyPass "/" "http://localhost:8080/"
</VirtualHost>
```

6.6.2 Using a URL prefix

If you do not want to dedicate a full virtual host to calibre, you can have it use a URL prefix. Start the calibre server as:

```
calibre-server --url-prefix /calibre --port 8080
```

The key parameter here is `--url-prefix /calibre`. This causes the Content server to serve all URLs prefixed by `/calibre`. To see this in action, visit `http://localhost:8080/calibre` in your browser. You should see the normal Content server website, but now it will run under `/calibre`.

With nginx, the required configuration is:

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

proxy_set_header X-Forwarded-For $remote_addr;
location /calibre/ {
    proxy_buffering off;
    proxy_pass http://127.0.0.1:8080$request_uri;
}
location /calibre {
    # we need a trailing slash for the Application Cache to work
    rewrite /calibre /calibre/ permanent;
}
```

For Apache, first enable the proxy modules in Apache, by adding the following to `httpd.conf`:

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

The exact technique for enabling the proxy modules will vary depending on your Apache installation. Once you have the proxy modules enabled, add the following rules to `httpd.conf` (or if you are using virtual hosts to the conf file for the virtual host in question):

```
AllowEncodedSlashes On
RewriteEngine on
RewriteRule ^/calibre/(.*) http://127.0.0.1:8080/calibre/$1 [proxy]
RedirectMatch permanent ^/calibre$ /calibre/
```

That's all, you will now be able to access the calibre Content server under the `/calibre` URL in your main server. The above rules pass all requests under `/calibre` to the calibre server running on port 8080 and thanks to the `--url-prefix` option above, the calibre server handles them transparently.

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 Creazione di un servizio per il server di calibre su un sistema Linux moderno

You can easily create a service to run calibre at boot on a modern ([systemd](https://www.freedesktop.org/wiki/Software/systemd/)⁴⁶) 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.

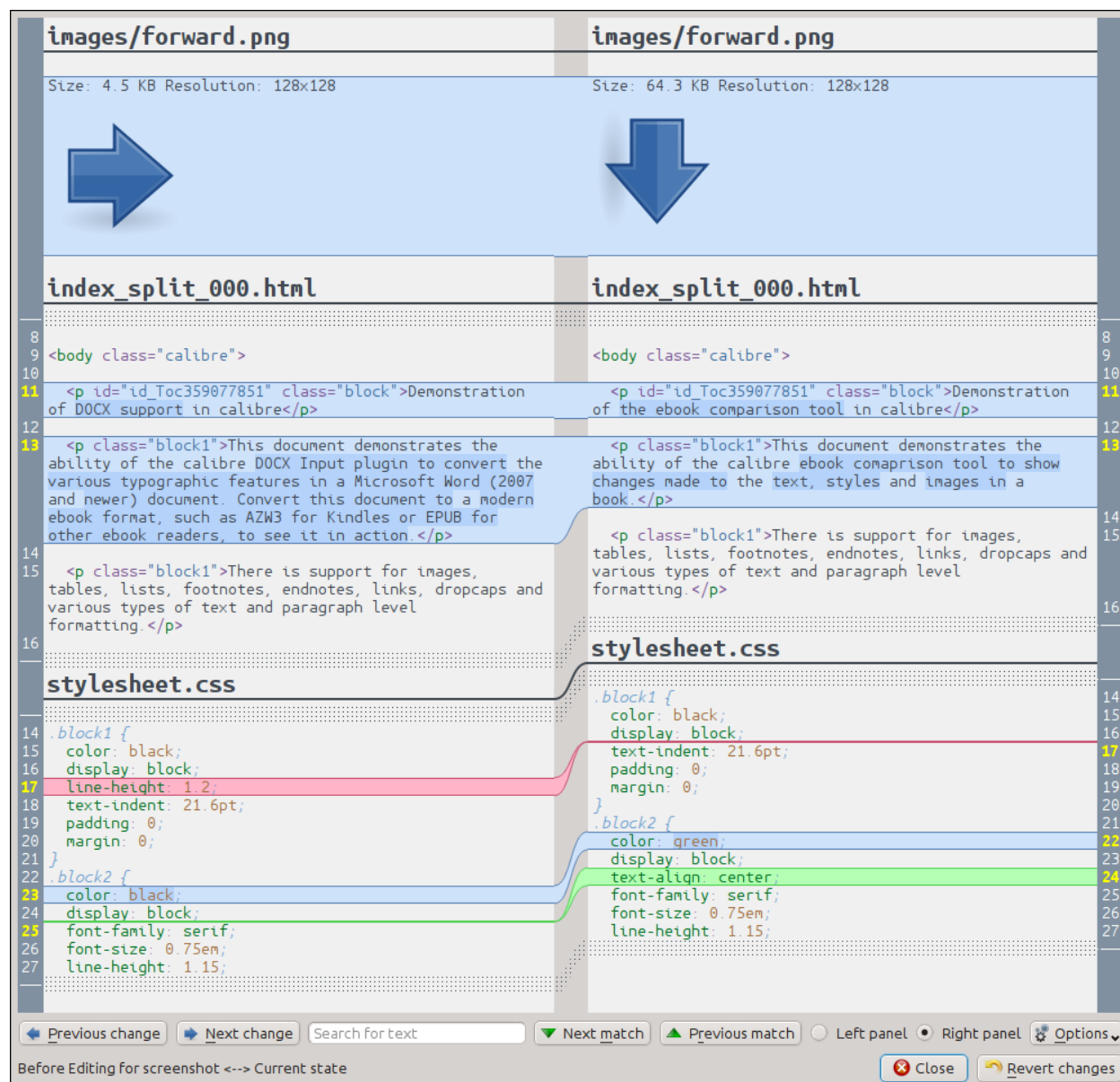
⁴⁶ <https://www.freedesktop.org/wiki/Software/systemd/>

Comparazione degli e-book

calibre include uno strumento di comparazione integrato di e-book che può essere usato per visualizzare le modifiche di un e-book dopo l'editing o la conversione. Può comparare i libri nei formati EPUB e AZW3.

Per usarlo, devi o aprire il libro in *Modifica degli e-book* (pagina 71) e poi fare clic su *File* → *Compara con un altro libro* o usare il pannello *Dettagli del libro* (pagina 20). Se effettui una conversione da EPUB a EPUB, il file EPUB originale verrà salvato come ORIGINAL_EPUB. Fai semplicemente clic con il tasto destro sulla voce ORIGINAL_EPUB nel pannello Dettagli Libro e scegli *Compara con il formato EPUB*.

Lo strumento di comparazione si presenterà come nell'immagine sottostante. Mostra le differenze in testo, stili e immagini dei libri selezionati.



7.1 Comprendere la vista di comparazione

Come si può vedere dagli screenshot sopra, la modalità di comparazione, mostra le differenze tra due libri fianco a fianco. Vengono mostrate solo le differenze, con alcune righe di contesto attorno. Ciò permette di vedere ad occhio solo le differenze all'interno di documenti con molto contenuto come un libro.

Il testo aggiunto è indicato con uno sfondo verde, il testo rimosso con uno sfondo rosso e il testo modificato con un sfondo blu.

Il numero di linea di tutti i testi modificati viene mostrato ai lati, rendendo semplice la modifica tramite l'editor. Quando si apre lo strumento di comparazione dall'interno dell'editor, è possibile fare doppio clic su una riga del pannello di destra per andare direttamente a quella riga nell'editor.

Una tecnica utile quando si comparano i libri è dire allo strumento di comparazione di migliorare l'aspetto dei file di testo e di stile prima di calcolare le differenze. Ciò può spesso portare a differenze più pulite e più facili da seguire. Per fare ciò,

fai clic sul pulsante :guilabel:”Opzioni” in basso a destra e scegli :guilabel:”Migliora l’aspetto dei file prima di compararli”. Nota che a volte l’abbellimento può avere effetti indesiderati, in quanto può causare l’alterazione del markup non valido per renderlo valido. Puoi anche cambiare il numero di linee di contesto mostrate intorno alle differenze tramite il pulsante :guilabel:”Opzioni”.

E’ possibile cercare qualsiasi testo nelle differenze tramite la barra di ricerca in basso. Bisogna specificare in quale pannello cercare, quello a *sinistra* o quello a *destra*.

7.2 Avvio dello strumento di comparazione

Lo strumento di comparazione è molto utile quando si hanno due versioni dello stesso libro e si vogliono vedere le differenze. Ci sono molti modi per avviare questo strumento.

7.2.1 Comparazione di due e-book

Apri il primo file nello strumento *Modifica degli e-book* (pagina 71). Ora fai clic su *File* → *Compara con un altro libro* e scegli il secondo file (deve essere nello stesso formato del primo). Si aprirà la vista di comparazione, con il file che viene modificato a destra e il secondo file a sinistra.

7.2.2 Comparazione di ORIGINAL_FMT e FMT

Quando si esegue una conversione in calibre da un FMT a sé stesso, il file originale viene salvato come ORIGINAL_FMT. Puoi vedere cosa è stato modificato nella conversione facendo clic con il tasto destro sulla voce ORIGINAL_FMT nel pannello *Dettagli del libro* (pagina 20) della finestra principale di calibre e selezionando *Compara con FMT*. La vista di comparazione si aprirà con ORIGINAL_FMT sulla sinistra e FMT sulla destra.

7.2.3 Compara un punto di controllo con lo stato attuale del libro mentre si edita

Lo strumento *Modifica degli e-book* (pagina 71) ha una funzione molto utile, chiamata *Punti di controllo* (pagina 83). Ciò consente di salvare lo stato corrente del libro come un *Punto di controllo* con nome, al quale è possibile ritornare se non ti piacciono le modifiche apportate dopo la sua creazione. I punti di controllo vengono inoltre creati automaticamente all’eseguirsi di varie azioni automatiche nell’editor. Puoi vedere la lista dei punti di controllo andando in *Visualizza* → *Punti di controllo* e poi usare il pulsante *Compara* per comparare lo stato del libro nel punto di controllo selezionato con lo stato corrente. Lo strumento di comparazione mostrerà il punto di controllo a sinistra e lo stato corrente a destra.

Modifica dei metadati degli e-book

Contenuti

- *Modifica dei metadati di un libro per volta* (pagina 117)
 - *Scaricamento dei metadati* (pagina 118)
 - *Gestione dei formati dei libri* (pagina 118)
 - *Tutto sulle copertine* (pagina 118)
- *Modifica dei metadati di più libri per volta* (pagina 118)
 - *Cerca e sostituisci* (pagina 119)
 - *Scaricamento in blocco dei metadati* (pagina 120)
- *Aggiunta di file di dati a un libro* (pagina 120)

Esistono e-book di vario tipo e dimensione, e il più delle volte i loro metadati (cose come titolo/autore/serie/editore) sono incompleti o errati. Il modo più semplice per modificare i metadati in calibre è semplicemente fare doppio clic su una voce e inserire il valore corretto. Per una modifica più avanzata usa gli strumenti di modifica dei metadati discussi di seguito.

8.1 Modifica dei metadati di un libro per volta

Fai clic sul libro che vuoi modificare, poi fai clic sul pulsante *Modifica metadati* o premi il tasto E. Si aprirà una finestra che ti permetterà di modificare tutti gli aspetti dei metadati. Essa possiede varie funzionalità per rendere la modifica più veloce ed efficiente. Una lista dei consigli più usati:

- Puoi fare clic sul pulsante che si trova tra titolo e autori per scambiarli automaticamente.
- Puoi fare clic sul pulsante a fianco dell'ordinamento autore per fare in modo che calibre lo compili automaticamente usando i valori di ordinamento conservati per ciascun autore. Usa la finestra *Gestisci autori* per vedere e modificare

i valori di ordinamento degli autori. Questa finestra può essere aperta facendo clic e tenendo premuto sul pulsante a fianco dell'ordinamento autore.

- Puoi fare clic sul pulsante a fianco dei tag per usare l'*Editor dei Tag* e gestire i tag associati al libro.
- La casella degli «Id» può essere usata per inserire un ISBN (e molti altri tipi di id), avrà uno sfondo rosso se inserisci un ISBN non valido. Sarà verde se l'ISBN è valido.
- La casella di ordinamento autore sarà rossa se il valore di ordinamento autore è diverso da quello che calibre pensa che dovrebbe essere.

8.1.1 Scaricamento dei metadati

La più carina delle funzionalità della finestra di modifica dei metadati è la sua abilità di compilare automaticamente molti campi di metadati prendendo i metadati da vari siti web. Al momento calibre utilizza Google Libri e Amazon. Lo scaricamento dei metadati può compilare per te titolo, autore, serie, tag, valutazione, descrizione e ISBN.

Per utilizzare lo scaricamento, compila i campi titolo e autore e fai clic sul pulsante *Scarica metadati*. calibre ti mostrerà una lista di libri che più corrispondono a titolo e autore. Se inserisci prima l'ISBN, sarà usato preferenzialmente ad autore e titolo. Se nessuna corrispondenza è trovata, prova a rendere la tua ricerca un pochino meno specifica includendo solo alcune parole chiave nel titolo e solo il cognome dell'autore.

8.1.2 Gestione dei formati dei libri

In calibre una singola voce di libro può avere molti «formati» diversi associati ad essa. Per esempio potresti aver ottenuto le Opere Complete di Shakespeare in formato EPUB e averle in seguito convertite in MOBI per leggerle sul tuo Kindle. calibre gestisce automaticamente i formati multipli al posto tuo. Nella sezione *Formati disponibili* della finestra Modifica metadati puoi gestire questi formati. Puoi aggiungere un nuovo formato, cancellarne uno esistente e persino chiedere a calibre di prendere i metadati e la copertina per la voce del libro dai metadati in uno dei formati.

8.1.3 Tutto sulle copertine

Puoi chiedere a calibre di scaricare le copertine dei libri per te, purché il libro abbia un ISBN noto. In alternativa, puoi indicare un file sul tuo computer da usare come copertina. calibre può persino generare una copertina predefinita contenente i metadati di base. Puoi trascinare immagini sulla copertina per cambiarla e puoi anche fare clic con il tasto destro per copiare/incollare immagini di copertina.

In aggiunta c'è un pulsante che ritaglia automaticamente i bordi della copertina, nel caso in cui la tua copertina abbia dei bordi sgradevoli.

8.2 Modifica dei metadati di più libri per volta

Per prima cosa seleziona i libri che vuoi modificare tenendo premuto **Ctrl** o **Maiusc** e facendo clic su di esse. Se selezioni più di un libro, il clic sul pulsante *Modifica metadati* farà aprire la finestra di modifica dei metadati *in gruppo*. Usando questa finestra puoi impostare velocemente autore/editore/valutazione/tag/serie ecc. di un gruppo di libri allo stesso valore. Questo è particolarmente utile se hai appena importato parecchi libri con dei metadati in comune. Questa finestra è molto potente, per esempio ha una scheda *Cerca e sostituisci* che puoi usare per eseguire operazioni di gruppo sui metadati e persino copiare metadati da una colonna a un'altra.

La normale finestra di modifica metadati ha anche i pulsanti *Successivo* e *Precedente*, che puoi usare per modificare i metadati di più libri uno dopo l'altro.

8.2.1 Cerca e sostituisci

La finestra *Modifica metadati per più libri* ti permette di eseguire operazioni arbitrariamente potenti di ricerca e sostituzione sui libri selezionati. Per impostazione predefinita usa un semplice trova e sostituisci sul testo, ma supporta anche le *espressioni regolari*. Per più informazioni sulle espressioni regolari, vedi *Tutto sull'utilizzo delle espressioni regolari in calibre* (pagina 205).

Come accennato sopra, ci sono due modalità di ricerca e sostituzione: corrispondenza dei caratteri ed espressione regolare. La corrispondenza dei caratteri cercherà nel *Campo di ricerca* che sceglierai i caratteri inseriti nella casella *cerca* e sostituirà quei caratteri con quelli inseriti nella casella *sostituisci*. Ogni occorrenza dei caratteri ricercati nel campo verrà sostituita. Per esempio, immagina che il campo in cui viene effettuata la ricerca contenga *vado a casa*. Se cerchi *a* per sostituirlo con *VERSO*, il risultato sarà *vVERSOdo VERSO cVERSOsVERSO*.

Se il campo in cui stai cercando è un campo *molteplice* come per i tag, ogni tag è trattato separatamente. Per esempio, se i tuoi tag contengono *Horror*, *Paura*, l'espressione di ricerca *r*, non restituirà risultati perché l'espressione sarà prima applicata ad *Horror* e poi a *Paura*.

Se vuoi che la ricerca ignori le differenze maiuscola/minuscola, togli la spunta dalla casella *Distingui le maiuscole*.

Puoi fare in modo che calibre cambi le maiuscole del risultato (l'informazione a sostituzione avvenuta) scegliendo una delle funzioni dalla casella *Applica funzione dopo la sostituzione*. Le operazioni disponibili sono:

- *Minuscolo* – cambia tutti i caratteri nel campo in caratteri minuscoli
- *Maiuscolo* – cambia tutti i caratteri nel campo in caratteri maiuscoli
- *Iniziali maiuscole* – Rende maiuscola la prima lettera di ogni parola.

La casella *La tua prova* è fornita per farti inserire del testo per controllare che il cerca/sostituisci stia facendo ciò che vuoi. Nella maggior parte dei casi le caselle di prova dei libri saranno sufficienti, ma è possibile che ci sia un caso che vuoi controllare che non è mostrato in queste caselle. Inserisci quel caso ne *La tua prova*.

La modalità espressione regolare possiede alcune differenze rispetto alla modalità caratteri, oltre (chiaramente) ad usare le espressioni regolari. La prima è che le funzioni sono applicate a parti della stringa trovata dalla stringa di ricerca, non all'intero campo. La seconda è che le funzioni sono applicate alla stringa di sostituzione, non all'intero campo.

La terza e più importante è che la stringa di sostituzione può fare riferimento a parti della stringa di ricerca usando i riferimenti inversi. Un riferimento inverso è `\n`, dove *n* è un numero intero che si riferisce all'*n*-esimo gruppo tra parentesi nell'espressione di ricerca. Per esempio, usando lo stesso esempio di prima, *vado a casa*, un'espressione di ricerca *vado* `(.)` `(....)`, e un'espressione di sostituzione *vado* `\2` `\1`, si otterrà come risultato *vado casa a*. Leggi il *Tutto sull'utilizzo delle espressioni regolari in calibre* (pagina 205) se hai bisogno di maggiori informazioni sui riferimenti inversi.

Un utile schema: immagina di voler cambiare le maiuscole di un intero campo. Il modo più facile per farlo è di usare la modalità caratteri, ma immaginiamo ulteriormente che tu voglia usare la modalità espressione regolare. L'espressione di ricerca dovrebbe essere `(^.*$)`, l'espressione di sostituzione `\1`, e la funzione per il cambio delle maiuscole desiderato dovrebbe essere selezionata.

Per finire, nella modalità espressione regolare puoi copiare valori da un campo all'altro. Rendi semplicemente diversi i campi sorgente e di destinazione. La copia può sostituire il campo di destinazione, aggiungere un prefisso (aggiungere all'inizio) o un suffisso (aggiungere alla fine) al campo. La casella di spunta "usa virgola" dice a calibre di aggiungere (o no) una virgola tra il testo e il campo di destinazione nelle modalità prefisso e suffisso. Se la destinazione è molteplice (p. es. i tag), non puoi togliere la spunta da questa casella.

Cerca e sostituisci è effettuato dopo che tutti gli altri cambiamenti ai metadati nelle altre schede sono stati applicati. Questo può causare confusione perché le caselle di prova mostreranno l'informazione prima degli altri cambi, ma l'operazione verrà applicata dopo di essi. Se hai dubbi a proposito di quello che succederà, non effettuare cerca/sostituisci con altri cambiamenti.

8.2.2 Scaricamento in blocco dei metadati

Se vuoi scaricare i metadati per più libri allo stesso tempo, fai clic con il tasto destro sul pulsante *Modifica metadati* e seleziona *Scarica metadati*. Puoi scegliere se scaricare soli i metadati, solo le copertine o entrambi.

8.3 Aggiunta di file di dati a un libro

Calibre può memorizzare un numero qualsiasi di file di dati aggiuntivi associati a un libro. Possono essere copertine alternative, materiale supplementare, ecc. Non possono essere visualizzati direttamente o utilizzati come fonti di conversione. Non sono nemmeno indicizzati dal motore di ricerca Full text di calibre. Per visualizzarli/aggiungerli/eliminarli, selezionare il libro e fare clic con il tasto destro del mouse sul pulsante *Modifica metadati* e scegliere *Gestisci file di dati*. Verrà visualizzata una finestra in cui è possibile eseguire operazioni su questi file. In alternativa, è possibile fare clic con il tasto destro del mouse sul pulsante *Aggiungi libri* e scegliere *Aggiungi file di dati ai record del libro selezionati* per aggiungere più rapidamente i file di dati.

Domande frequenti

Contenuti

- *Conversione di formato degli e-book* (pagina 121)
- *Integrazione dei dispositivi* (pagina 124)
- *Gestione della biblioteca* (pagina 131)
- *Varie* (pagina 136)

9.1 Conversione di formato degli e-book

Contenuti

- *Da e verso quali formati è supportata la conversione in calibre?* (pagina 122)
- *Quali sono i migliori formati di partenza da cui convertire?* (pagina 122)
- *Ho convertito un file PDF, ma il risultato presenta vari problemi?* (pagina 122)
- *Come posso convertire il mio file contenente caratteri non inglesi o virgolette intelligenti?* (pagina 122)
- *Che succede con l'indice nei file MOBI?* (pagina 123)
- *Come posso convertire una serie di file HTML in un ordine specifico?* (pagina 123)
- *Il file EPUB che ho prodotto con calibre non è valido?* (pagina 124)
- *Come posso utilizzare alcune delle funzionalità avanzate degli strumenti di conversione?* (pagina 124)

9.1.1 Da e verso quali formati è supportata la conversione in calibre?

calibre supporta la conversione di molti formati di input verso molti formati di output. È in grado di convertire qualsiasi formato di input nella seguente lista verso ogni formato di output.

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

Formati di uscita: AZW3, EPUB, DOCX, FB2, HTMLZ, OEB, LIT, LRF, MOBI, PDB, PMLZ, RB, PDF, RTF, SNB, TCR, TXT, TXTZ, ZIP

Nota: PRC è un formato generico, calibre supporta i file PRC con intestazioni TextRead e MOBIBook. Anche PDB è un formato generico. calibre supporta i file PDB di tipo eReader, Plucker (solo in input), PML e zTxt. Il supporto a DJVU è solo per la conversione di file DJVU che contengono testo incorporato. Questi sono tipicamente generati da software OCR. I libri MOBI possono essere di due tipi: Mobi6 e KF8. calibre supporta pienamente entrambi. I file MOBI hanno spesso estensione .azw o .azw3. Sono supportati i file DOCX di Microsoft Word 2007 e successivi.

9.1.2 Quali sono i migliori formati di partenza da cui convertire?

In ordine decrescente di preferenza: LIT, MOBI, AZW, EPUB, AZW3, FB2, FBZ, DOCX, HTML, PRC, ODT, RTF, PDB, TXT, PDF

9.1.3 Ho convertito un file PDF, ma il risultato presenta vari problemi?

PDF è un formato terribile da convertire. Per una serie di problemi diversi che si incontrano per convertirlo, vedi: *Conversione di documenti PDF* (pagina 67).

9.1.4 Come posso convertire il mio file contenente caratteri non inglesi o virgolette intelligenti?

Ci sono due fattori relativi a questo problema

1. La conoscenza del file sorgente: calibre prova a indovinare quale codifica di caratteri usa il tuo file sorgente, ma spesso questo è impossibile, perciò devi dirgli che codifica utilizzare. Questo può essere fatto nell'interfaccia grafica attraverso il campo *Codifica caratteri in ingresso* nella sezione *Aspetto → Testo* della finestra di conversione. Gli strumenti da linea di comando hanno l'opzione `ebook-convert-txt-input --input-encoding`.
2. Quando si aggiungono file HTML a calibre, potrebbe essere necessario indicare la codifica caratteri dei file. Per farlo vai su *Preferenze → Avanzate → Plugin → Tipo di file* e configura il plugin *HTML to ZIP* indicando la codifica caratteri dei tuoi file HTML. Ora quando aggiungerai file HTML a calibre, saranno elaborati correttamente. I file HTML da sorgenti diverse, spesso hanno codifiche diverse, perciò dovrai cambiare questa impostazione ripetutamente. Una codifica comune per molti files da internet è «cp1252» e suggerisco di provare prima questa codifica. Nota che quando si convertono file HTML, lascia vuoto il campo della codifica dei caratteri in ingresso menzionato precedentemente. Questo perché il plugin *HTML to ZIP* converte automaticamente i file HTML con la codifica standard (UTF-8).

9.1.5 Che succede con l'indice nei file MOBI?

La prima cosa da notare è che la maggior parte degli e-book ha due indici. Uno è l'indice tradizionale, come quello che si trova nei libri. Questo indice è parte del contenuto principale del documento e può avere lo stile che preferisci; questo indice è chiamato «indice dei contenuti».

Poi c'è l'«indice dei metadati»: un indice che non fa parte del testo del libro e a cui si può solitamente accedere attraverso bottoni speciali su un lettore. Per esempio, nel visualizzatore di e-book di calibre, per vedere questo indice usi il pulsante Mostra indice. Questo indice non può essere formattato dal creatore del libro, come viene rappresentato è deciso dal programma di visualizzazione.

Nel formato MOBI, la situazione è un po' confusa perché, unico tra i formati principali per gli e-book, *non* ha un supporto decente per l'indice dei metadati. Un libro MOBI simula la presenza di un indice dei metadati mettendo un indice *aggiuntivo* alla fine del libro. Quando clicchi su Vai all'indice sul tuo Kindle, è a questo indice aggiuntivo che ti porta.

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 Come posso convertire una serie di file HTML in un ordine specifico?

Per eseguire la conversione di una serie di file HTML in un ordine specifico, devi creare una tabella dei file da elaborare. Si tratta di un altro file HTML, che contiene i link a tutti gli altri file nell'ordine desiderato. Questo file è del tipo:

```
<html>
  <body>
    <h1>Table of Contents</h1>
    <p style="text-indent:0pt">
      <a href="file1.html">First File</a><br/>
      <a href="file2.html">Second File</a><br/>
      .
      .
      .
```

(continues on next page)

(continua dalla pagina precedente)

```
</p>
</body>
</html>
```

Then, just add this HTML file to the GUI and use the *Convert* button to create your e-book. You can use the option in the Table of Contents section in the conversion dialog to control how the Table of Contents is generated.

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 Il file EPUB che ho prodotto con calibre non è valido?

Non è garantito che un EPUB prodotto da calibre sia valido. L'unica garanzia che calibre può dare è che se gli dai in pasto XHTML 1.1 + CSS 2.1 validi produrrà un EPUB valido. calibre fa del suo meglio per assicurarsi che gli EPUB che produce funzionino effettivamente su un'ampia gamma di dispositivi, un obiettivo che è incompatibile con la produzione di EPUB validi, ma che è molto più importante per la grande maggioranza degli utenti. Se ti serve uno strumento che produca sempre EPUB validi, calibre non fa per te. Questo significa che se vuoi inviare un EPUB prodotto con calibre a uno store online che usa uno strumento di controllo della validità degli EPUB, dovrai assicurarti tu stesso che l'EPUB sia valido, calibre non lo farà per te – in altre parole devi dare in pasto a calibre XHTML + CSS validi nei documenti in ingresso.

9.1.8 Come posso utilizzare alcune delle funzionalità avanzate degli strumenti di conversione?

You can get help on any individual feature of the converters by mousing over it in the GUI or running `ebook-convert dummy.html .epub -h` at a terminal. A good place to start is to look at the following demo file that demonstrates some of the advanced features [html-demo.zip](https://calibre-ebook.com/downloads/html-demo.zip)⁴⁷.

9.2 Integrazione dei dispositivi

Contenuti

- *Quali sono i device supportati da calibre?* (pagina 125)
- *Come posso aiutare a rendere il mio dispositivo supportato in calibre?* (pagina 125)
- *Il mio dispositivo non è stato rilevato da calibre?* (pagina 126)
- *Il mio dispositivo è non standard o diverso. Cosa posso fare per collegarlo?* (pagina 126)
- *Come utilizzare calibre con il mio iPad/iPhone/iPod touch?* (pagina 126)
- *How do I use calibre with my Android phone/tablet or Kindle Fire?* (pagina 127)

⁴⁷ <https://calibre-ebook.com/downloads/html-demo.zip>

- *Can I access my calibre books using the web browser in my Kindle or other reading device?* (pagina 128)
- *Non posso inviare email usando calibre?* (pagina 128)
- *My device is getting mounted read-only in Linux, so calibre cannot connect to it?* (pagina 129)
- *Why does calibre not support collections on the Kindle or shelves on the Nook?* (pagina 129)
- *I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?* (pagina 130)
- *Covers for books I send to my e-ink Kindle show up momentarily and then are replaced by a generic cover?* (pagina 130)
- *Le copertine dei miei file MOBI hanno smesso di essere mostrate in Kindle per PC/Kindle per Android/iPad ecc.* (pagina 131)
- *I transferred some books to my Kindle using calibre and they did not show up?* (pagina 131)

9.2.1 Quali sono i device supportati da calibre?

calibre can directly connect to all the major (and most of the minor) e-book reading devices, smartphones, tablets, etc. In addition, using the *Connect to folder* function you can use it with any e-book reader that exports itself as a USB disk. Finally, you can connect wirelessly to any device that has a web browser using the calibre Content server.

9.2.2 Come posso aiutare a rendere il mio dispositivo supportato in calibre?

Se il tuo dispositivo appare come disco USB nel sistema operativo, aggiungere il supporto a calibre è molto semplice. Ci servono solo alcune informazioni da parte tua:

- Una lista completa dei formati di e-book che il tuo dispositivo supporta.
- C'è una cartella speciale del dispositivo in cui tutti i file degli e-book devono essere messi? Inoltre, il dispositivo rileva i file inseriti in sottocartelle?
- Ci servono anche informazioni sul tuo dispositivo che calibre raccoglierà automaticamente. Per prima cosa, se il tuo dispositivo supporta le schede SD, inseriscile. Dopodiché connetti il dispositivo al tuo computer. Dentro calibre vai in *Preferenze* → *Varie* e fai clic sul pulsante «Debug del riconoscimento dispositivi». Questo produrrà delle informazioni di debug. Copiale in un file e ripeti il processo, questa volta con il dispositivo disconnesso dal 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](https://calibre-ebook.com/it/bugs)⁴⁸.

⁴⁸ <https://calibre-ebook.com/it/bugs>

9.2.3 Il mio dispositivo non è stato rilevato da calibre?

Segui questi passi per trovare il problema:

- Make sure that you are connecting only a single device to your computer at a time. Do not have another calibre supported device like an iPhone/iPad etc. at the same time.
- If you are connecting an Apple iDevice (iPad, iPod Touch, iPhone), Apple no longer allows third party software to connect to their devices using a USB cable. Instead use a wireless connection, via the calibre Content server.
- If you are connecting a Kindle Fire or other Android device, read the note under *Utilizzare un cavo USB* (pagina 127).
- On macOS if you get permission errors when connecting a device to calibre, you can fix that by looking under *System Preferences > Security and Privacy > Privacy > Files and Folders*.
- Make sure you are running the latest version of calibre (currently 6.27.0). The latest version can always be downloaded from [the calibre website](https://calibre-ebook.com/it/download)⁴⁹. 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
- If all the above steps fail, go to *Preferences → Miscellaneous* and click *Debug device detection* with your device attached and post the output as a ticket on [the calibre bug tracker](https://bugs.launchpad.net/calibre)⁵⁰.

9.2.4 Il mio dispositivo è non standard o diverso. Cosa posso fare per collegarlo?

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 Come utilizzare calibre con il mio iPad/iPhone/iPod touch?

An easy way to browse your calibre collection from your Apple device is by using *Il Server dei Contenuti di Calibre* (pagina 105), 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*.

⁴⁹ <https://calibre-ebook.com/it/download>

⁵⁰ <https://bugs.launchpad.net/calibre>

The Content server allows you to read books directly in Safari itself. In addition, there are many apps for your iDevice that can connect to the calibre Content server. Examples include: Marvin, Mapleread and iBooks itself.

Sto usando il server dei contenuti

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 *Il Server dei Contenuti di Calibre* (pagina 105) 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 e-book reading applications for Android: Some examples (in no particular order): [FBReader](#)⁵¹, [Moon+](#)⁵², [Mantano](#)⁵³, [Aldiko](#)⁵⁴, [Kindle](#)⁵⁵.

Utilizzare un cavo 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.

⁵¹ <https://play.google.com/store/apps/details?id=org.geometerplus.zlibrary.ui.android&hl=en>

⁵² <https://play.google.com/store/apps/details?id=com.flyersoft.moonreader&hl=en>

⁵³ <https://play.google.com/store/apps/details?id=com.mantano.reader.android.lite&hl=en>

⁵⁴ <https://play.google.com/store/apps/details?id=com.aldiko.android&hl=en>

⁵⁵ https://play.google.com/store/apps/details?id=com.amazon.kindle&feature=related_apps

Over the air

calibre has a builtin web server, the *Il Server dei Contenuti di Calibre* (pagina 105). 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 *Il Server dei Contenuti di Calibre* (pagina 105) for details on running the server and finding out the right IP address to use.

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

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

9.2.7 Can I access my calibre books using the web browser in my Kindle or other reading device?

calibre has a *Content server* that exports the books in calibre as a web page. See *Il Server dei Contenuti di Calibre* (pagina 105) 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 Non posso inviare email usando calibre?

Because of the large amount of spam in email, sending email can be tricky, as different mail servers use different strategies to block email. The most common problem is if you are sending email directly (without a mail relay) in calibre. Many servers (for example, Amazon) block email that does not come from a well known relay. The most robust way to setup email sending in calibre is to do the following:

- Create a free GMX account at [GMX](https://www.gmx.com)⁵⁶.
- Goto *Preferences* → *Sharing* → *Sharing books by email* in calibre and click the *Use GMX* button and fill in the information asked for.
- Log into your GMX account on the website and enable SMTP sending (*Settings* → *POP3 & IMAP* → *Send and receive emails via external program*)

⁵⁶ <https://www.gmx.com>

- calibre will then be able to use GMX to send the mail.
- If you are sending to your Kindle, remember to update the email preferences on your Amazon Kindle page to allow email sent from your GMX email address. Also note that Amazon does not allow email delivery of AZW3 and new style (KF8) MOBI files. Finally, Amazon recently started sending confirmation emails that you have to click on back to your GMX account before the book is actually delivered.

Even after doing this, you may have problems. One common source of problems is that some poorly designed antivirus programs block calibre from opening a connection to send email. Try adding an exclusion for calibre in your antivirus program.

Nota: Microsoft/GMX can disable your account if you use it to send large amounts of email. So, when using these services to send mail calibre automatically restricts itself to sending one book every five minutes. If you don't mind risking your account being blocked you can reduce this wait interval by going to *Preferences* → *Advanced* → *Tweaks* in calibre.

Nota: Google recently deliberately broke their email sending protocol (SMTP) support in an attempt to force everyone to use their web interface so they can show you more ads. They are trying to claim that SMTP is insecure, that is incorrect and simply an excuse. Use some other email provider instead.

Nota: Se ti preoccupa dare a calibre accesso al tuo account di posta elettronica, puoi creare un nuovo e gratuito account email con GMX o Outlook da usare solamente per 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 ⁵⁷

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](https://www.mobileread.com/forums/showthread.php?t=244202)⁵⁷.

⁵⁷ <https://www.mobileread.com/forums/showthread.php?t=244202>

Nota: Amazon have removed the ability to manipulate collections completely in their newer models, like the Kindle Touch and Kindle Fire, making even the above plugin useless, unless you root your Kindle and install custom firmware.

9.2.11 I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?

The Kobo has very buggy firmware. Connecting to it has been known to fail at random. Certain combinations of motherboard, USB ports/cables/hubs can exacerbate this tendency to fail. If you are getting an error when connecting to your touch with calibre try the following, each of which has solved the problem for *some* calibre users.

- Connect the Kobo directly to your computer, not via USB Hub
- Try a different USB cable and a different USB port on your computer
- Log out of the Kobo and log in again, this causes it to rebuild the database, fixing corrupted database errors.
- Try upgrading the firmware on your Kobo Touch to the latest
- Try resetting the Kobo (sometimes this cures the problem for a little while, but then it re-appears, in which case you have to reset again and again)
- Try only putting one or two books onto the Kobo at a time and do not keep large collections on the Kobo

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

This happens because of an Amazon bug. They try to download a cover for the book from their servers and when that fails, they replace the existing cover that calibre created with a generic cover. For details see [this forum thread](#)⁵⁸. 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) Send the book to the Kindle with calibre
- 2) Disconnect the Kindle and wait for Amazon to destroy the cover
- 3) Reconnect the Kindle to 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: Se la soluzione alternativa non funziona, assicurati che il firmware Kindle sia almeno la versione 5.12.5, rilasciata ad aprile 2020.

⁵⁸ <https://www.mobileread.com/forums/showthread.php?t=329945>

9.2.13 Le copertine dei miei file MOBI hanno smesso di essere mostrate in Kindle per PC/Kindle per Android/iPad ecc.

This is caused by a bug in the Amazon software. You can work around it by going to *Preferences* → *Conversion* → *Output Options* → *MOBI output* and setting the *Enable sharing of book content* option. If you are reconverting a previously converted book, you will also have to enable the option in the conversion dialog for that individual book (as per book conversion settings are saved and take precedence).

Note that doing this will mean that the generated MOBI will show up under personal documents instead of Books on the Kindle Fire and Amazon whispersync will not work, but the covers will. It's your choice which functionality is more important to you. I encourage you to contact Amazon and ask them to fix this bug.

The bug in Amazon's software is that when you put a MOBI file on a Kindle, unless the file is marked as a Personal document, Amazon assumes you bought the book from it and tries to download the cover thumbnail for it from its servers. When the download fails, it refuses to fallback to the cover defined in the MOBI file. This is likely deliberate on Amazon's part to try to force authors to sell only through them. In other words, the Kindle only displays covers for books marked as Personal Documents or books bought directly from Amazon.

If you send a MOBI file to an e-ink Kindle with calibre using a USB connection, calibre works around this Amazon bug by uploading a cover thumbnail itself. However, that workaround is only possible when using a USB connection and sending with calibre. Note that if you send using email, Amazon will automatically mark the MOBI file as a Personal Document and the cover will work, but the book will show up in Personal Documents.

9.2.14 I transferred some books to my Kindle using calibre and they did not show up?

Books sent to the Kindle only show up on the Kindle after they have been *indexed* by the Kindle. This can take some time. If the book still does not show up after some time, then it is likely that the Kindle indexer crashed. Sometimes a particular book can cause the indexer to crash. Unfortunately, Amazon has not provided any way to deduce which book is causing a crash on the Kindle. Your only recourse is to either reset the Kindle, or delete all files from its memory using Windows Explorer (or whatever file manager you use) and then send the books to it again, one by one, until you discover the problem book. Once you have found the problem book, delete it off the Kindle and do a MOBI to MOBI or MOBI to AZW3 conversion in calibre and then send it back. This will most likely take care of the problem.

9.3 Gestione della biblioteca

Contenuti

- *Dove sono archiviati i file dei libri?* (pagina 132)
- *Come gestisce calibre i nomi degli autori e l'ordinamento?* (pagina 132)
- *Why doesn't calibre let me store books in my own folder structure?* (pagina 133)
- *Why doesn't calibre have a column for foo?* (pagina 134)
- *Can I have a column showing the formats or the ISBN?* (pagina 134)
- *Come posso trasferire i miei dati di calibre da un computer a un altro?* (pagina 134)
- *The list of books in calibre is blank!* (pagina 135)
- *I am getting errors with my calibre library on a networked drive/NAS?* (pagina 135)

9.3.1 Dove sono archiviati i file dei libri?

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 Come gestisce calibre i nomi degli autori e l'ordinamento?

Author names are complex, especially across cultures, see [this note](#)⁵⁹ for some of the complexities. calibre has a very flexible strategy for managing author names. The first thing to understand is that books and authors are separate entities in calibre. A book can have more than one author, and an author can have more than one book. You can manage the authors of a book by the edit metadata dialog. You can manage individual authors by right clicking on the author in the Tag browser on the left of the main calibre window and selecting *Manage authors*. Using this dialog you can change the name of an author and also how that name is sorted. This will automatically change the name of the author in all the books of that author. When a book has multiple authors, separate their names using the & character.

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*

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

Note that you can set an individual author's sort value to whatever you want using *Manage authors*. This is useful when dealing with names that calibre will not get right, such as complex multi-part names like Miguel de Cervantes Saavedra or when dealing with Asian names like Sun Tzu.

With all this flexibility, it is possible to have calibre manage your author names however you like. For example, one common request is to have calibre display author names LN, FN. To do this, and if the note below does not apply to you, then:

- Set the `author_sort_copy_method` tweak to `copy` as described above.
- Restart calibre. Do not change any book metadata before doing the remaining steps.
- Change all author names to LN, FN using the Manage authors dialog.
- After you have changed all the authors, press the *Recalculate all author sort values* button.
- Press OK, at which point calibre will change the authors in all your books. This can take a while.

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.
 - Apri la finestra Gestisci Autori. Premi il pulsante `opia tutti i valori di ordinamento autore in autore`.
 - Check through the authors to be sure you are happy. You can still press Cancel to abandon the changes. Once you press OK, there is no undo.
 - Press OK, at which point calibre will change the authors in all your books. This can take a while.
-

9.3.3 Why doesn't calibre let me store books in my own folder structure?

The whole point of calibre's library management features is that they provide a search and sort based interface for locating books that is *much* more efficient than any possible folder scheme you could come up with for your collection. Indeed, once you become comfortable using calibre's interface to find, sort and browse your collection, you won't ever feel the need to hunt through the files on your disk to find a book again. By managing books in its own folder structure of Author -> Title -> Book files, calibre is able to achieve a high level of reliability and standardization. To illustrate why a search/tagging based interface is superior to folders, consider the following. Suppose your book collection is nicely sorted into folders with the following scheme:

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

Now this makes it very easy to find for example all science fiction books by Isaac Asimov in the Foundation series. But suppose you want to find all unread science fiction books. There's no easy way to do this with this folder scheme, you would instead need a folder scheme that looks like:

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

In calibre, you would instead use tags to mark genre and read status and then just use a simple search query like `tag:scifi` and not `tag:read`. calibre even has a nice graphical interface, so you don't need to learn its search language instead you can just click on tags to include or exclude them from the search.

To those of you that claim that you need access to the filesystem, so that you can have access to your books over the network, calibre has an excellent Content server that gives you access to your calibre library over the net.

If you are worried that someday calibre will cease to be developed, leaving all your books marooned in its folder structure, explore the powerful *Save to disk* feature in calibre that lets you export all your files into a folder structure of arbitrary complexity based on their metadata.

Finally, the reason there are numbers at the end of every title folder, is for *robustness*. That number is the id number of the book record in the calibre database. The presence of the number allows you to have multiple records with the same title and author names. It is also part of what allows calibre to magically regenerate the database with all metadata if the database file gets corrupted. Given that calibre's mission is to get you to stop storing metadata in filenames and stop using the filesystem to find things, the increased robustness afforded by the id numbers is well worth the uglier folder names.

If you are still not convinced, then I'm afraid calibre is not for you. Look elsewhere for your book cataloguing needs. Just so we're clear, **this is not going to change**. Kindly do not contact us in an attempt to get us to change this.

9.3.4 Why doesn't calibre have a column for foo?

calibre is designed to have columns for the most frequently and widely used fields. In addition, you can add any columns you like. Columns can be added via *Preferences* → *Interface* → *Add your own columns*. Watch the tutorial [UI Power tips](#)⁶⁰ to learn how to create your own columns, or read [this blog post](#)⁶¹.

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 [Il linguaggio dei modelli di calibre](#) (pagina 153).

9.3.5 Can I have a column showing the formats or the ISBN?

Yes, you can. Follow the instructions in the answer above for adding custom columns.

9.3.6 Come posso trasferire i miei dati di calibre da un computer a un altro?

You can export all calibre data (books, settings and plugins) and then import it on another computer. First let's see how to export the data:

- Right click the calibre icon in the main calibre toolbar and select *Export/import all calibre data*. Note that if there is currently a device connected, this menu option will not be available – so, disconnect any connected devices. Then click the button labelled *Export all your calibre data*. You will see a list of all your calibre libraries. Click OK and choose an empty folder somewhere on your computer. The exported data will be saved in this folder. Simply copy this folder to your new computer and follow the instructions below to import the data.
- Install calibre on your new computer and run through the *Welcome wizard*, it does not matter what you do there, as you will be importing your old settings in the next step. You will now have an empty calibre, with just the *Getting Started* guide in your library. Once again, right click the calibre button and choose *Export/import all calibre data*. Then click the button labelled *Import previously exported data*. Select the folder with the exported data that you copied over earlier. You will now have a list of libraries you can import. Go through the list one by one, and select the new location for each library (a location is just an empty folder somewhere on your computer). Click OK. After the import completes, calibre will restart, with all your old libraries, settings and calibre plugins.

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.

⁶⁰ <https://calibre-ebook.com/it/demo#tutorials>

⁶¹ <https://blog.calibre-ebook.com/calibre-custom-columns/>

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

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

9.3.7 The list of books in calibre is blank!

In order to understand why that happened, you have to understand what a calibre library is. At the most basic level, a calibre library is just a folder. Whenever you add a book to calibre, that book's files are copied into this folder (arranged into sub folders by author and title). Inside the calibre library folder, at the top level, you will see a file called `metadata.db`. This file is where calibre stores the metadata like title/author/rating/tags etc. for *every* book in your calibre library. The list of books that calibre displays is created by reading the contents of this `metadata.db` file.

There can be two reasons why calibre is showing a empty list of books:

- Your calibre library folder changed its location. This can happen if it was on an external disk and the drive letter for that disk changed. Or if you accidentally moved the folder. In this case, calibre cannot find its library and so starts up with an empty library instead. To remedy this, do a right-click on the calibre icon in the calibre toolbar and select *Switch/create library*. Click the little blue icon to select the new location of your calibre library and click OK. If you don't know the new location search your computer for the file `metadata.db`.
- Your `metadata.db` file was deleted/corrupted. In this case, you can ask calibre to rebuild the `metadata.db` from its backups. Right click the calibre icon in the calibre toolbar and select *Library maintenance->Restore database*. calibre will automatically rebuild `metadata.db`.

9.3.8 I am getting errors with my calibre library on a networked drive/NAS?

Do not put your calibre library on a networked drive.

A filesystem is a complex beast. Most network filesystems lack various filesystem features that calibre uses. Some don't support file locking, some don't support hardlinking, some are just flaky. Additionally, calibre is a single user application, if you accidentally run two copies of calibre on the same networked library, bad things will happen. Finally, different OSes impose different limitations on filesystems, so if you share your networked drive across OSes, once again, bad things *will* happen.

Consider using the calibre Content server to make your books available on other computers. Run calibre on a single computer and access it via the Content server or a Remote Desktop solution.

If you must share the actual library, use a file syncing tool like DropBox or rsync instead of a networked drive. If you are using a file-syncing tool it is **essential** that you make sure that both calibre and the file syncing tool do not try to access the calibre library at the same time. In other words, **do not** run the file syncing tool and calibre at the same time.

Even with these tools there is danger of data corruption/loss, so only do this if you are willing to live with that risk. In particular, be aware that **Google Drive** is incompatible with calibre, if you put your calibre library in Google Drive, **you will suffer data loss**. See [this thread](https://www.mobileread.com/forums/showthread.php?t=205581)⁶² for details.

⁶² <https://www.mobileread.com/forums/showthread.php?t=205581>

9.4 Varie

Contenuti

- *Amazon smetterà di consegnare i file MOBI?* (pagina 136)
- *I want calibre to download news from my favorite news website.* (pagina 137)
- *Perché il nome calibre?* (pagina 137)
- *Perché calibre mostra solo alcune delle mie famiglie di caratteri su macOS?* (pagina 137)
- *calibre non si avvia su Windows?* (pagina 137)
- *calibre si blocca/non risponde di tanto in tanto?* (pagina 138)
- *The calibre E-book viewer and Edit book tools do not work on Windows?* (pagina 139)
- *Using the viewer or doing any conversions results in a permission denied error on Windows* (pagina 139)
- *calibre is not starting/crashing on macOS?* (pagina 139)
- *Quando apro il Lettore e-book di calibre vedo solo una schermata nera o bianca?* (pagina 140)
- *Ho scaricato il programma di installazione, ma non funziona?* (pagina 140)
- *My antivirus program claims calibre is a virus/trojan?* (pagina 140)
- *Come effettuo il backup di calibre?* (pagina 141)
- *Come posso utilizzare in calibre i libri acquistati in formato EPUB (ovvero cosa faccio con i file .acsm)?* (pagina 141)
- *Mi appare un errore «Permesso Negato»?* (pagina 141)
- *Can I have the comment metadata show up on my reader?* (pagina 142)
- *Come faccio a far usare a calibre il mio proxy HTTP?* (pagina 142)
- *Mi piacerebbe che una funzionalità fosse aggiunta a calibre. Cosa posso fare?* (pagina 142)
- *Perché calibre non ha un aggiornamento automatico?* (pagina 142)
- *Qual è la licenza di calibre?* (pagina 143)
- *Come avvio calibre dalla mia chiavetta USB?* (pagina 143)
- *Come posso avviare parti di calibre come lo scaricamento delle notizie e il server dei contenuti sul mio server Linux personale?* (pagina 143)

9.4.1 Amazon smetterà di consegnare i file MOBI?

Amazon [have announced](#)⁶³ that they will stop accepting MOBI files emailed to @kindle.com email addresses. You can instruct calibre to send EPUB instead of MOBI by going to *Preferences* → *Sharing books by email* and then removing MOBI from the list of formats to send to your @kindle.com email address and adding EPUB instead.

Tieni però presente che l'elaborazione di EPUB da parte di Amazon è molto carente, svariati file EPUB che funzionano in ogni altro caso verranno rifiutati. In casi del genere puoi provare questo espediente:

1. Converti il file EPUB in MOBI

⁶³ <https://blog.the-ebook-reader.com/2022/05/03/amazon-dropping-mobi-support-on-send-to-kindle-apps/>

2. Then convert the MOBI file back to EPUB and send the resulting EPUB file

Ciò rimuoverà tutta la formattazione avanzata, i caratteri incorporati, ecc. ma aumenterà notevolmente la probabilità che l'EPUB venga accettato da Amazon.

Nota: If you were previously using email delivery of periodicals downloaded by calibre, you will be better off sending those by USB cable or downloading them from the calibre Content server via the Kindle's built-in browser. However, if you want to continue using email delivery you can try changing the output format in Preferences->Behavior to EPUB, then calibre will download the news in EPUB format. Whether Amazon will accept the EPUB or not is a whole other question.

9.4.2 I want calibre to download news from my favorite news website.

If you are reasonably proficient with computers, you can teach calibre to download news from any website of your choosing. To learn how to do this see *Aggiunta del tuo sito di notizie preferito* (pagina 29).

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

9.4.3 Perché il nome calibre?

Segli la tua preferita:

- Convertitore e (And) biblioteca (LIBRARY) per E-book
- Un prodotto di grosso *calibro*
- Un omaggio al SONY Librie, che fu il primo lettore di e-book basato su e-ink
- Lo ha scelto mia moglie ;-)

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 Perché calibre mostra solo alcune delle mie famiglie di caratteri su macOS?

calibre embeds fonts in e-book files it creates. E-book files support embedding only TrueType and OpenType (.ttf and .otf) fonts. Most fonts on macOS systems are in .dfont format, thus they cannot be embedded. calibre shows only TrueType and OpenType fonts found on your system. You can obtain many such fonts on the web. Simply download the .ttf/.otf files and add them to the Library/Fonts folder in your home folder.

9.4.5 calibre non si avvia su Windows?

Sono possibili svariate cause:

- 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:
 - Disinstalla calibre
 - Riavvia il computer
 - Re-install calibre. But do not start calibre from the installation wizard.

⁶⁴ <https://www.mobileread.com/forums/forumdisplay.php?f=228>

- 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.
 - Avvia calibre
 - From now on you should be able to start calibre normally.
- If you get an error about a Python function terminating unexpectedly after upgrading calibre, first uninstall calibre, then delete the folders (if they exist) `C:\Program Files\Calibre` and `C:\Program Files\Calibre2`. Now re-install and you should be fine.
 - If you get an error in the *Welcome wizard* on an initial run of calibre, try choosing a folder like `C:\library` as the calibre library (calibre sometimes has trouble with library locations if the path contains non-English characters, or only numbers, etc.)
 - Try running it as administrator (Right click on the icon and select *Run as administrator*)

If it still won't launch, start a command prompt (press the Windows key and R; then type `cmd.exe` in the Run dialog that appears). At the command prompt type the following command and press Enter:

```
calibre-debug -g
```

Post any output you see in a help message on the [Forum](#)⁶⁵.

9.4.6 calibre si blocca/non risponde di tanto in tanto?

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.
- In alcune versioni di Windows a 64 bit sono presenti software/impostazioni di sicurezza che impediscono a calibre 64 bit di funzionare correttamente. Se stai usando la versione a 64 bit di calibre prova a passare alla versione a 32 bit.
- If the crash happens when you are trying to copy text from the calibre E-book viewer, it is most likely caused by some clipboard monitoring/managing application you have running. Turn it off and you should be fine.
- If the crashes happen specifically when you are using a file dialog, like clicking on the *Add books* button or the *Save to Disk* button, then you have some software that has installed broken Shell extensions on your computer. Known culprits include: *SpiderOak*, *odrive sync* and *Dell Backup and Recovery* and *NetDrive*. If you have one of these, uninstall them and you will be fine. You can also use the [NirSoft Shell Extension Viewer](#)⁶⁶ to see what shell extensions are installed on your system and disable them individually, if you don't want to uninstall the full program. Remember to use «Restart Explorer» or reboot your computer after disabling the shell extensions.

If none of the above apply to you, then there is some other program on your computer that is interfering with calibre. First reboot your computer in safe mode, to have as few running programs as possible, and see if the crashes still happen. If they do not, then you know it is some program causing the problem. The most likely such culprit is a program that

⁶⁵ <https://www.mobileread.com/forums/forumdisplay.php?f=166>

⁶⁶ <https://www.nirsoft.net/utils/shexview.html>

modifies other programs” behavior, such as an antivirus, a device driver, something like RoboForm (an automatic form filling app) or an assistive technology like Voice Control or a Screen Reader.

The only way to find the culprit is to eliminate the programs one by one and see which one is causing the issue. Basically, stop a program, run calibre, check for crashes. If they still happen, stop another program and repeat.

9.4.7 The calibre E-book viewer and Edit book tools do not work on Windows?

These two programs use hardware acceleration as they embed a version of the Chrome browser to render HTML. If they do not work it will be because of incompatibility with your system’s GPU (graphics) drivers. Try updating these first, and reboot. If that does not fix it, you can set the `QTWEBENGINE_CHROMIUM_FLAGS` environment variable to the value `--disable-gpu` to turn off hardware acceleration. See [this page](#)⁶⁷ for details.

9.4.8 Using the viewer or doing any conversions results in a permission denied error on Windows

Something on your computer is preventing calibre from accessing its own temporary files. Most likely the permissions on your Temp folder are incorrect. Go to the folder file: `C:\Users\USERNAME\AppData\Local` in Windows Explorer and then right click on the file: `Temp` folder, select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

Some users have reported that running the following command in an Administrator Command Prompt fixed their permissions. To get an Administrator Command Prompt search for `cmd.exe` in the start menu, then right click on the command prompt entry and select *Run as administrator*. At the command prompt type the following command and press Enter:

```
icacls "%appdata%\..\Local\Temp" /reset /T
```

Alternately, you can run calibre as Administrator, but doing so will cause some functionality, such as drag and drop to not work.

Finally, some users have reported that disabling UAC fixes the problem.

9.4.9 calibre is not starting/crashing on macOS?

One common cause of failures on macOS is the use of accessibility technologies that are incompatible with the graphics toolkit calibre uses. Try turning off VoiceOver if you have it on. Also go to System Preferences->System->Universal Access and turn off the setting for enabling access for assistive devices in all the tabs. Another cause can be some third party apps that modify system behavior, such as Smart Scroll.

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

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

then the problem is probably a corrupted font cache. You can clear the cache by following these [instructions](#)⁶⁸. If that doesn’t solve it, look for a corrupted font file on your system, in `~/Library/Fonts` or the like. An easy way to check for corrupted fonts in macOS is to start the «Font Book» application, select all fonts and then in the File menu, choose «Validate fonts».

⁶⁷ <https://doc.qt.io/qt-5/qtwebengine-debugging.html>

⁶⁸ <https://www.macworld.com/article/1139383/fontcacheclear.html>

9.4.10 Quando apro il Lettore e-book di calibre vedo solo una schermata nera o bianca?

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 *Variabili d'ambiente* (pagina 280) for how to change environment variables.

9.4.11 Ho scaricato il programma di installazione, ma non funziona?

Downloading from the Internet can sometimes result in a corrupted download. If the calibre installer you downloaded is not opening, try downloading it again. If re-downloading it does not work, download it from [an alternate location](#)⁶⁹. If the installer still doesn't work, then something on your computer is preventing it from running.

- Try temporarily disabling your antivirus program (Microsoft Security Essentials, or Kaspersky or Norton or McAfee or whatever). This is most likely the culprit if the upgrade process is hanging in the middle.
- Similarly, if the installer is failing/rolling back and you have Microsoft PowerToys running, quit it.
- Try rebooting your computer and running a registry cleaner like [Wise registry cleaner](#)⁷⁰.
- Try a clean install. That is, uninstall calibre, delete `C:\Program Files\Calibre2` (or wherever you previously chose to install calibre). Then re-install calibre. Note that uninstalling does not touch your books or settings.
- Try downloading the installer with an alternate browser. For example if you are using Microsoft Edge, try using Firefox or Chrome instead.
- If you get an error about a missing DLL on Windows, then most likely, the permissions on your temporary folder are incorrect. Go to the folder `C:\Users\USERNAME\AppData\Local` in Windows Explorer and then right click on the `Temp` folder and select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

Se ancora non riesci a far funzionare il programma di installazione e sei su Windows, puoi usare l'[installazione portatile di calibre](#)⁷¹, che non necessita di un programma di installazione (è semplicemente un file ZIP).

9.4.12 My antivirus program claims calibre is a virus/trojan?

The first thing to check is that you are downloading calibre from the [official website](#)⁷². Make sure you are clicking the download links on the left, not the advertisements on the right. calibre is a very popular program and unscrupulous people try to setup websites offering it for download to fool the unwary.

If you have the official download and your antivirus program is still claiming calibre is a virus, then, your antivirus program is wrong. Antivirus programs use heuristics, patterns of code that «look suspicious» to detect viruses. It's rather like racial profiling. calibre is a completely open source product. You can actually browse the source code yourself (or hire someone to do it for you) to verify that it is not a virus. Please report the false identification to whatever company you buy your antivirus software from. If the antivirus program is preventing you from downloading/installing calibre, disable it temporarily, install calibre and then re-enable it.

⁶⁹ <https://github.com/kovidgoyal/calibre/releases/latest>

⁷⁰ <https://www.wisecleaner.com>

⁷¹ https://calibre-ebook.com/it/download_portable

⁷² <https://calibre-ebook.com/it/download>

9.4.13 Come effettuo il backup di calibre?

The most important thing to backup is the calibre library folder, that contains all your books and metadata. This is the folder you chose for your calibre library when you ran calibre for the first time. You can get the path to the library folder by clicking the calibre icon on the main toolbar. You must backup this complete folder with all its files and sub-folders.

You can switch calibre to using a backed up library folder by simply clicking the calibre icon on the toolbar and choosing your backup library folder. A backed up library folder backs up your custom columns and saved searches as well as all your books and metadata.

If you want to backup the calibre configuration/plugins, you have to backup the config folder. You can find this config folder via *Preferences* → *Miscellaneous*. Note that restoring configuration folders is not officially supported, but should work in most cases. Just copy the contents of the backup folder into the current configuration folder to restore.

9.4.14 Come posso utilizzare in calibre i libri acquistati in formato EPUB (ovvero cosa faccio con i file .acsm)?

Most purchased EPUB books have *DRM* (pagina 365). This prevents calibre from opening them. You can still use calibre to store and transfer them to your e-book reader. First, you must authorize your reader on a Windows machine with Adobe Digital Editions. Once this is done, EPUB books transferred with calibre will work fine on your reader. When you purchase an epub book from a website, you will get an «.acsm» file. This file should be opened with Adobe Digital Editions, which will then download the actual «.epub» e-book. The e-book file will be stored in the folder «My Digital Editions», from where you can add it to calibre.

9.4.15 Mi appare un errore «Permesso Negato»?

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.
- Se il file è aperto in un altro programma.
- 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 Come faccio a far usare a calibre il mio proxy HTTP?

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 Mi piacerebbe che una funzionalità fosse aggiunta a calibre. Cosa posso fare?

Hai due possibilità:

1. Crea una patch trafficando con calibre e inviamela per la revisione e l'inclusione. Vedi [Sviluppo](#)⁷³.
2. [Open a bug requesting the feature](#)⁷⁴. Remember that while you may think your feature request is extremely important/essential, calibre developers might not agree. Fortunately, calibre is open source, which means you always have the option of implementing your feature yourself, or hiring someone to do it for you. Furthermore, calibre has a comprehensive plugin architecture, so you might be able to develop your feature as a plugin, see [Scrivere i propri plugin per estendere le funzionalità di calibre](#) (pagina 216).

9.4.19 Perché calibre non ha un aggiornamento automatico?

Per diverse ragioni:

- *There is no need to update every week.* If you are happy with how calibre works turn off the update notification and be on your merry way. Check back to see if you want to update once a year or so. There is a check box to turn off the update notification, on the update notification itself.
- calibre downloads currently use [about 150TB of bandwidth a month](#)⁷⁵. Implementing automatic updates would greatly increase that and end up costing thousands of dollars a month, which someone has to pay.
- If I implement a dialog that downloads the update and launches it, instead of going to the website as it does now, that would save the most ardent calibre updater, *at most five clicks a week*. There are far higher priority things to do in calibre development.

⁷³ <https://calibre-ebook.com/it/get-involved>

⁷⁴ <https://calibre-ebook.com/it/bugs>

⁷⁵ <https://calibre-ebook.com/dynamic/downloads>

- 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](#) (pagina 337).
- There are third party automatic updaters for calibre made by calibre users in the [calibre forum](#)⁷⁶.

9.4.20 Qual è la licenza di calibre?

calibre is licensed under the GNU General Public License v3 (an open source license). This means that you are free to redistribute calibre as long as you make the source code available. So if you want to put calibre on a CD with your product, you must also put the calibre source code on the CD. The source code is available [for download](#)⁷⁷. You are free to use the results of conversions from calibre however you want. You cannot use either code or libraries from calibre in your software without making your software open source. For details, see [The GNU GPL v3](#)⁷⁸.

9.4.21 Come avvio calibre dalla mia chiavetta USB?

Una versione portatile di calibre è disponibile `:website:` qui <download_portable>``.

9.4.22 Come posso avviare parti di calibre come lo scaricamento delle notizie e il server dei contenuti sul mio server Linux personale?

First, you must install calibre onto your Linux server. If your server is using a modern Linux distribution, you should have no problems installing calibre onto it.

Nota: calibre needs GLIBC ≥ 2.31 and libstdc++ $\geq 6.0.28$. If you have an older server, you will either need to compile these from source, or use calibre 3.48 which requires GLIBC ≥ 2.17 or 2.85.1 which requires GLIBC ≥ 2.13 or calibre 1.48 which requires only GLIBC ≥ 2.10 . In addition, although the calibre command line utilities do not need a running X server, some of them do require the X server libraries to be installed on your system. This is because of Qt, which is used for various image processing tasks, and links against these libraries. If you get an ImportError about some Qt modules, you are likely missing some X libraries. Typical candidates are: `libxcb-xinerama0`, `libegl1`, `libopengl0`.

Puoi avviare il server calibre tramite il comando:

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

Puoi scaricare news e convertirle in un e-book con il comando:

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

Se invece vuoi generare file MOBI, utilizza `outputfile.mobi` insieme a `--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:

⁷⁶ <https://www.mobileread.com/forums/forumdisplay.php?f=238>

⁷⁷ <https://download.calibre-ebook.com>

⁷⁸ <https://www.gnu.org/licenses/gpl.html>

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

Remember to read the *Interfaccia a riga di comando* (pagina 295) section of the calibre User Manual to learn more about these, and other commands.

Qui troverete le guide che vi spiegano come iniziare a utilizzare le funzionalità avanzate di calibre, come gli XPath e i modelli.

10.1 Gestire sottogruppi di libri, ad esempio il «genere»

Alcune persone desiderano organizzare i libri nella loro biblioteca in sottogruppi, in modo simile alle sottocartelle. La ragione più comunemente addotta è la creazione di gerarchie nei generi, ma ce ne sono molte altre. Un utente chiedeva un modo di organizzare i libri di testo per materia e numero di corso. Un altro voleva tenere traccia dei regali per argomento e beneficiario. Questa guida si servirà dell'esempio dei generi per il resto di questa sezione.

Prima di continuare, ricorda che non stiamo parlando di cartelle nel disco rigido. I sottogruppi non sono cartelle. I libri non verranno copiati da nessuna parte. La struttura dei file della biblioteca di calibre non verrà influenzata. Invece, stiamo presentando un modo di organizzare e mostrare sottogruppi di libri all'interno di una biblioteca di calibre.

- *Configurazione* (pagina 147)
- *Ricerca* (pagina 149)
- *Restrizioni* (pagina 150)
- *Utili funzioni di modello* (pagina 150)

I requisiti comunemente previsti per sottogruppi come i generi sono:

- Un sottogruppo (p.es., un genere) deve contenere (puntare a) libri, non categorie di libri. Questo è quello che distingue i sottogruppi dalle categorie utente di calibre.
- Un libro può essere in più di un sottogruppo (genere). Questo distingue i sottogruppi dalle cartelle di file reali.
- I sottogruppi (generi) devono formare una gerarchia; i sottogruppi possono contenere altri sottogruppi.

I tag ti danno i primi due requisiti. Se aggiungi un tag a un libro con il suo genere puoi usare il Navigatore dei Tag (o la ricerca) per trovare i libri con quel genere, e abbiamo il primo. Molti libri possono avere lo stesso (o gli stessi) tag, e questo è il secondo. Il problema è che i tag non soddisfano il terzo requisito. Non permettono una gerarchia.



La funzionalità di gerarchia di calibre ti fornisce il terzo requisito – la capacità di vedere i generi in un “albero” e la capacità di cercare con facilità i libri per genere o sottogenere. Per esempio, poniamo che la tua struttura dei generi sia simile a questa:

```

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

Usando la funzionalità di gerarchia, puoi vedere questi generi nel Navigatore dei Tag in un diagramma ad albero, come mostrato nell'immagine dello schermo. In questo esempio il livello più esterno (Genere) è una colonna personalizzata che contiene i generi. I generi contenenti sottogeneri appaiono con un piccolo triangolo a fianco. Il clic su quel triangolo espanderà l'elemento e mostrerà i sottogeneri, come puoi vedere per History e Science Fiction.

Il clic su un genere permette di cercare tutti i libri con quel genere o che sono «figli» di quel genere. Per esempio, facendo clic su Science Fiction si otterranno tutti e tre i generi figli, Alternate History, Military, e Space Opera. Facendo clic su Alternate History si otterranno i libri inclusi in quel genere, mentre saranno ignorati quelli in Military e Space Opera. Naturalmente, un libro può avere più di un genere. Se un libro ha sia Space Opera sia Military come generi, vedrai quel libro quando fai clic su ciascuno di essi. La ricerca viene discussa più approfonditamente più avanti.

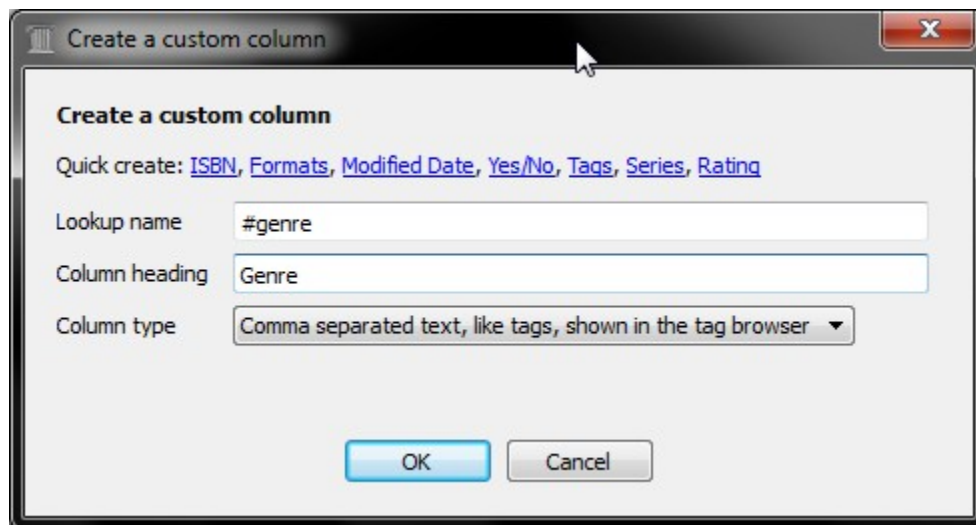
Un'altra cosa che puoi vedere dall'immagine è che il genere Military appare due volte, una sotto History e una sotto Science Fiction. Dato che i generi sono gerarchizzati, questi sono due generi separati. Un libro può essere incluso in uno, nell'altro, o (difficilmente in questo caso) in entrambi. Per esempio, i libri della serie «La Seconda Guerra Mondiale» di Winston Churchill potrebbe essere in «Science Fiction.Military», e del resto anche in «Science Fiction.Space Opera.»

Una volta che un genere esiste, vale a dire che almeno un libro ha quel genere, può facilmente essere applicato ad altri libri trascinandoli dalla vista biblioteca sopra il genere che vuoi che i libri abbiano. Puoi anche applicare i generi modificando i metadati dei libri; maggiori dettagli saranno dati più avanti.

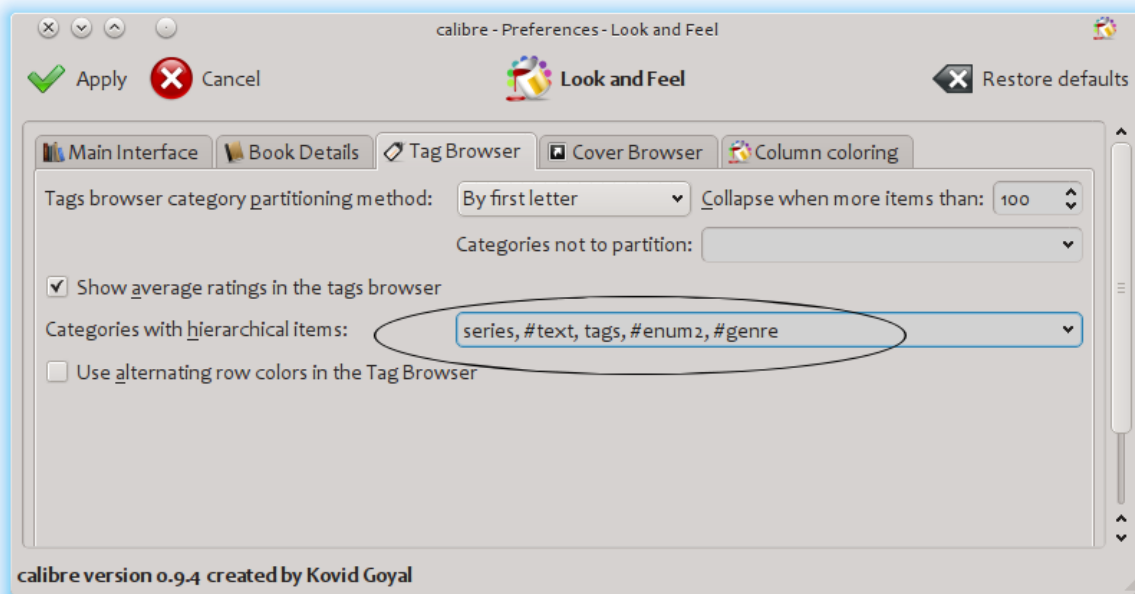
10.1.1 Configurazione

A questo punto, la domanda potrebbe essere «Come è stato configurato tutto quanto?» Ci sono tre passi da seguire: 1) crea la colonna personalizzata, 2) di" a calibre che la nuova colonna deve essere trattata come gerarchica, e 3) aggiungi dei generi.

La colonna personalizzata viene creata nel solito modo, usando Preferenze -> Aggiungi colonne personalizzate. Questo esempio usa «#genre» come nome di riferimento e «Genre» come intestazione per la colonna. Il tipo di colonna è «Testo separato da virgola, mostrato nel navigatore dei Tag».

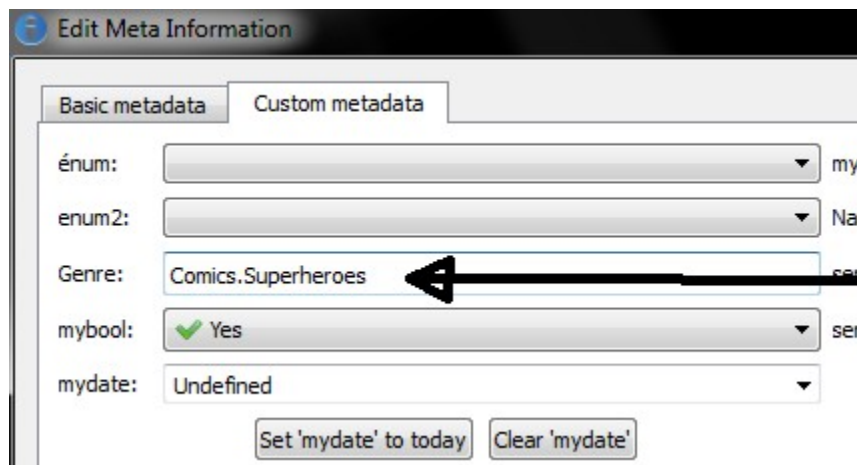


Quindi, dopo aver riavviato calibre, devi dirgli che la colonna deve essere trattata come gerarchica. Vai in *Preferenze* → *Aspetto* → *Navigatore dei Tag* e inserisci il nome di riferimento «#genre» nella casella «Categorie con elementi gerarchici». Fai clic su *Applica*, e la configurazione è completa.



In questo momento non ci sono generi nella colonna. Ci manca l'ultimo passaggio: come applicare un genere a un libro. Un genere non esiste in calibre finché non appare in almeno un libro. Per imparare come applicare un genere per la prima volta, dobbiamo addentrarci nei dettagli di come appare un genere nei metadati di un libro.

Una gerarchia di "oggetti" è costruita creando un elemento che consiste di frasi separate da punti. Proseguendo con l'esempio dei generi, questi elementi sarebbero «History.Military», «Mysteries.Vampire», «Science Fiction.Space Opera», ecc. Pertanto per creare un nuovo genere, hai bisogno di prendere un libro che dovrebbe averlo, modificarne i metadati ed inserire il nuovo genere nella colonna che hai creato. Andando avanti con il nostro esempio, se volessi assegnare un nuovo genere «Fumetti» con un sottogenere «Supereroi» ad un libro (albo), ne dovresti modificare i metadati, selezionare la scheda dei metadati personalizzati e quindi inserire «Fumetti.Supereroi» come mostrato nell'immagine seguente (ignora le altre colonne personalizzate):



Dopo aver fatto tutto quello di cui sopra, vedrai nel Navigatore dei Tag:



Da qui in avanti, per applicare questo nuovo genere ad un libro (verosimilmente un fumetto), potrai o trascinare il libro sopra il genere, o aggiungerlo al libro usando lo strumento di modifica metadati esattamente nello stesso modo sopradescritto.

Nota: La vista gerarchica funziona solamente se il Navigatore dei tag è impostato con l'ordinamento per nome. Questa è l'impostazione predefinita, e può essere verificata facendo clic sul pulsante *Configura* in fondo al Navigatore dei tag.

10.1.2 Ricerca

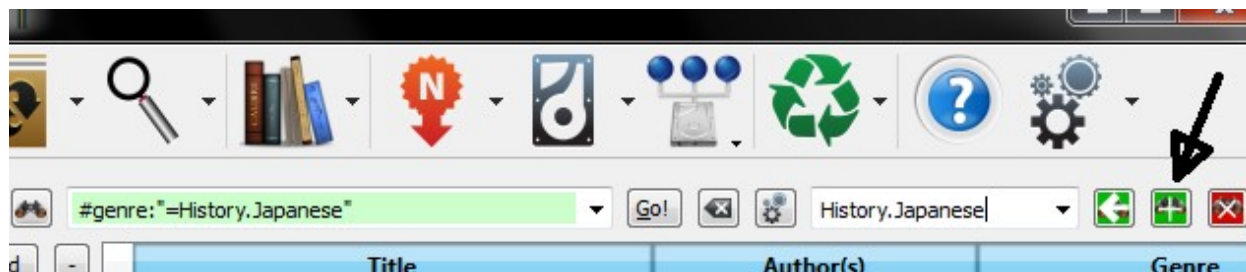


Il modo più facile di cercare per genere è attraverso il Navigatore dei Tag, facendo clic sul genere che desideri vedere. Il clic su un genere con «figli» ti mostrerà i libri con quel genere e tutti i generi figli. Tuttavia, ciò fa nascere una domanda. Solo perché un genere ha dei figli, questo non significa che non sia anch'esso un genere a sé. Per esempio, un libro può avere il genere «History» ma non «History.Military». Come cercare libri che hanno solo «History»?

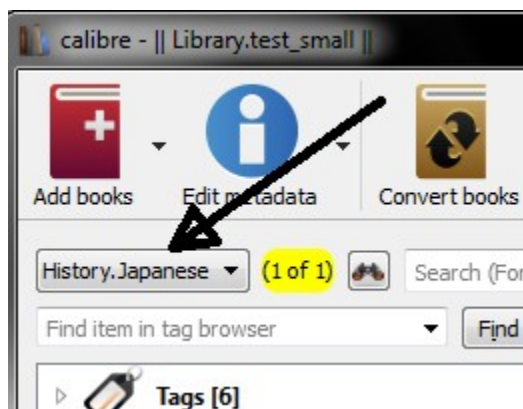
Il meccanismo di ricerca del Navigatore dei Tag si accorge se un elemento ha dei figli. Se ne ha, il clic sull'elemento scorre tra 5 modalità di ricerca invece delle normali tre. La prima è il classico più verde, che ti mostra solamente i libri con quel genere (p.es. History). La seconda è un doppio più (mostrato sopra), che ti mostra i libri con quel genere e tutti i sottogeneri (es. History e History.Military). La terza è il classico meno rosso, che ti mostra i libri senza quello specifico genere. La quarta è un doppio meno, che ti mostra i libri senza quel genere o quei sottogeneri. La quinta è lo stato iniziale, nessun segno, che significa nessuna ricerca.

10.1.3 Restrizioni

Se ricerchi un genere e poi crei per esso una ricerca salvata, puoi usare la casella “Filtra per” per creare una biblioteca virtuale di libri con quel genere. Ciò è utile se vuoi fare altre ricerche all’interno di quel genere o se vuoi gestire/aggiornare i metadati per i libri con quel genere. Proseguendo con il nostro esempio, puoi creare una Ricerca Salvata di nome “History.Japanese” prima facendo clic sul genere Japanese nel Navigatore dei Tag per ottenere una ricerca nel campo di ricerca, inserendo History.Japanese nel campo della ricerca salvata, quindi facendo clic sul pulsante «Salva ricerca» (la casella verde con il più bianco, sul lato destro).



Dopo aver creato la ricerca salvata, puoi usarla come restrizione.



10.1.4 Utili funzioni di modello

Potresti voler usare le informazioni sul genere in un modello, come per il salvataggio su disco o per l’invio al dispositivo. La domanda allora potrebbe essere «Come ottengo il nome del genere o dei generi più esterni? Esiste una funzione di modello di calibre, subitems, che rende le cose più facili.

Per esempio, poniamo che tu voglia aggiungere il livello più esterno dei generi al modello di salvataggio su disco per creare cartelle con il nome del genere, come in «History/The Gathering Storm - Churchill, Winston». Per fare ciò, devi estrarre il primo livello della gerarchia e aggiungerlo all’inizio insieme ad una barra obliqua ad indicare che deve creare una cartella. Il modello riportato di seguito lo permette:

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

Vedi *Il linguaggio dei modelli* (pagina 153) per maggiori informazioni sui modelli e sulla funzione `subitems()`.

10.2 Guida di XPath

In questa guida ti sarà fornita un'introduzione graduale a XPath⁷⁹, un linguaggio per query che può essere utilizzato per selezionare parti arbitrarie di documenti HTML⁸⁰ in calibre. XPath è uno standard ampiamente utilizzato e una ricerca su Google rivelerà parecchie informazioni. Questa guida, tuttavia, si concentra sull'utilizzo di XPath per compiti relativi agli e-book, come trovare le intestazioni dei capitoli in un documento HTML non strutturato.

Contenuti

- *Selezione per nome del tag* (pagina 151)
- *Selezione per attributi* (pagina 152)
- *Selezione per contenuto del tag* (pagina 152)
- *Libro di esempio* (pagina 152)
- *Funzioni interne di XPath* (pagina 153)

10.2.1 Selezione per nome del tag

La forma più semplice di selezione è selezionare i tag per nome. Ad esempio, immagina di voler selezionare tutti i tag «<h2>» in un documento. La query di XPath per farlo è semplicemente:

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

Il prefisso // significa *cerca ad ogni livello del documento*. Adesso immagina di dover cercare dei tag che si trovano dentro tag <a>. Ciò si può ottenere con:

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

Se vuoi cercare dei tag in un livello particolare del documento, modifica il prefisso:

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

Questo troverà solo <p>A very short e-book to demonstrate the use of XPath.</p> nel *Libro di esempio* (pagina 152), ma nessuno degli altri tag <p>. Il prefisso h: negli esempi precedenti è richiesto per trovare i tag XHTML. Ciò è necessario perché, internamente, calibre rappresenta tutti i contenuti come XHTML. In XHTML i tag hanno un *namespace* e h: è il prefisso di namespace per i tag HTML.

Adesso, immagina di voler selezionare sia i tag <h1> che quelli <h2>. Per farlo, abbiamo bisogno di un costrutto XPath chiamato *predicato*. Un *predicato* è semplicemente un test usato per selezionare i tag. I test possono essere arbitrariamente potenti e continuando lungo questa guida vedrai degli esempi sempre più potenti. Un predicato si crea racchiudendo l'espressione di test in parentesi quadre:

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

Ci sono diverse nuove funzionalità in questa espressione XPath. La prima è l'uso del carattere jolly *. Significa *trova qualsiasi tag*. Ora dai un'occhiata all'espressione di test `name()='h1' or name()='h2'`. *name()* è un esempio di *funzione interna*. Essa valuta semplicemente il nome del tag. Quindi, usandola, possiamo selezionare tag i cui nomi sono

⁷⁹ <https://en.wikipedia.org/wiki/XPath>

⁸⁰ <https://en.wikipedia.org/wiki/HTML>

h1 o *h2*. Tieni presente che la funzione *name()* ignora gli spazi di nomi in modo che non sia necessario il prefisso *h:*. XPath ha diverse utili funzioni interne. Alcune di loro saranno introdotte in questa guida.

10.2.2 Selezione per attributi

Per selezionare i tag in base alle loro caratteristiche, è richiesto l'uso di predicati

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

Qui, l'operatore @ si riferisce agli attributi del tag. È possibile utilizzare alcune delle *funzioni incorporate di XPath* per eseguire una corrispondenza più sofisticata sui valori degli attributi.

10.2.3 Selezione per contenuto del tag

Usando XPath, puoi persino selezionare i tag in base al testo che contengono. Il modo migliore per farlo è usare la potenza delle *espressioni regolari* tramite la funzione integrata *re: test()*:

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

Qui l'operatore . si riferisce al contenuto del tag, proprio come l'operatore @ si riferiva ai suoi attributi.

10.2.4 Libro di esempio

```
<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 Funzioni interne di XPath

name()

Il nome del tag attuale

contains()

`contains(s1, s2)` restituisce *true* se *s1* contiene *s2*.

re:test()

`re:test(src, pattern, flags)` restituisce *true* se la stringa *src* corrisponde all'espressione regolare *pattern*. Un flag particolarmente utile è *i*, che non considera la differenza MAIUSCOLE/minuscole. Un primo aiuto sulla sintassi per le espressioni regolari è disponibile in *sintassi delle regexp* <<https://docs.python.org/2.7/library/re.html>> _

10.3 Il linguaggio dei modelli di calibre

Il linguaggio del modello di calibre è un linguaggio specifico utilizzato in tutto calibre per attività quali l'indicazione dei percorsi dei file, la formattazione dei valori e il calcolo del valore per le colonne specificate dall'utente. Esempi:

- Specify the folder structure and file names when saving files from the calibre library to the disk or e-book reader.
- Define rules for adding icons and colors to the calibre book list.
- Define *virtual columns* that contain data from other columns.
- Advanced library searching.
- Advanced metadata search and replace.

The language is built around the notion of a *template*, which specifies which book metadata to use, computations on that metadata, and how it is to be formatted.

10.3.1 Modelli di base

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`.

Oltre ai campi basati sulle colonne standard puoi anche usare:

- `{formats}` - A list of formats available in the calibre library for a book
- `{identifiers:select(isbn)}` - Il codice ISBN del libro

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 Formattazione avanzata

In addition to metadata substitution, templates can conditionally include additional text and control how substituted data is formatted.

Inclusione di testo condizionale

Sometimes you want text to appear in the output only if a field is not empty. A common case is `series` and `series_index` where you want either nothing or the two values separated by a hyphen. calibre handles this case using a special template expression syntax.

For example and using the above Foundation example, assume you want the template to produce *Foundation - 3 - Second Foundation*. This template produces that output:

```
{series} - {series_index} - {title}
```

However, if a book has no series the template will produce *- - the title*, which is probably not what you want. Generally, people want the result be the title without the extraneous hyphens. You can accomplish this using the following template syntax:

```
{field:|prefix_text|suffix_text}
```

This template expression says that if `field` has the value `XXXX` then the result will be *prefix_textXXXXsuffix_text*. If `field` is empty (has no value) then the result will be the empty string (nothing) because the prefix and suffix are ignored. The prefix and suffix can contain blanks.

Do not use subtemplates (`{ ... }`) or functions (see below) in the prefix or the suffix.

Using this syntax, we can solve the above no-series problem with the template:

```
{series}{series_index:| - | - }{title}
```

The hyphens will be included only if the book has a series index, which it has only if it has a series. Continuing the Foundation example again, the template will produce *Foundation - 1 - Second Foundation*.

Note:

- 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}`.

Formattazione

Suppose you want the `series_index` to be formatted as three digits with leading zeros. This does the trick:

```
{series_index:0>3s} - Three digits with leading zeros
```

For trailing zeros, use:

```
{series_index:0<3s} - Three digits with trailing zeros
```

If you use series indices with fractional values, e.g., 1.1, you might want the decimal points to line up. For example, you might want the indices 1 and 2.5 to appear as 01.00 and 02.50 so that they will sort correctly on a device that does lexical sorting. To do this, use:

```
{series_index:0>5.2f} - Five characters consisting of two digits with leading zeros, a decimal point, then 2 digits after the decimal point.
```

If you want only the first two letters of the data, use:

```
{author_sort:.2} - Only the first two letters of the author sort name
```

Much of the calibre template language formatting comes from Python. For more details on the syntax of these advanced formatting operations see the [Python documentation](https://docs.python.org/3/library/string.html#formatstrings)⁸¹.

10.3.3 Uso dei modelli per definire colonne personalizzate

Templates can be used to display information that isn't in calibre metadata, or to display metadata differently from calibre's normal format. For example, you might want to show the `ISBN`, a field that calibre does not display. You can accomplish this creating a custom column with the type *Column built from other columns* (hereafter called *composite columns*) and providing a template to generate the displayed text. The column will display the result of evaluating the template. For example, to display the ISBN, create the column and enter `{identifiers:select(isbn)}` in the template box. To display a column containing the values of two series custom columns, separated by a comma, use `{#series1:||,}{#series2}`.

Composite columns can use any template option, including formatting.

Note: You cannot edit the data displayed in a composite column. Instead you edit the source columns. If you edit a composite column, for example by double-clicking it, calibre will open the template for editing, not the underlying data.

10.3.4 Modelli e pannelli di controllo

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.

⁸¹ <https://docs.python.org/3/library/string.html#formatstrings>

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:

- un pannello che corrisponde esattamente per formato e dispositivo, p.es., EPUB e ANDROID
- a plugboard with an exact match on format and the special any device choice, e.g., EPUB and any device
- un pannello con la scelta speciale any format`e una corrispondenza esatta per il dispositivo, p.es. ``any format e ANDROID
- a plugboard with any format and any device

The tags and authors fields have special treatment, because both of these fields can hold more than one item. A book can have many tags and many authors. When you specify that one of these two fields is to be changed, the template's result is examined to see if more than one item is there. For tags, the result is cut apart wherever calibre finds a comma. For example, if the template produces the value `Thriller, Horror`, then the result will be two tags, `Thriller` and `Horror`. There is no way to put a comma in the middle of a tag.

The same thing happens for authors, but using a different character for the cut, a & (ampersand) instead of a comma. For example, if the template produces the value `Blogs, Joe&Posts, Susan`, then the book will end up with two authors, `Blogs, Joe` and `Posts, Susan`. If the template produces the value `Blogs, Joe;Posts, Susan`, then the book will have one author with a rather strange name.

Plugboards affect the metadata written into the book when it is saved to disk or written to the device. Plugboards do not affect the metadata used by `save to disk` and `send to device` to create the file names. Instead, file names are constructed using the templates entered on the appropriate preferences window.

10.3.5 Uso delle funzioni nei modelli - Modalità a Funzione Singola

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* (pagina 172) and *General Program Mode* (pagina 159).

Some functions require regular expressions. In the template language regular expression matching is case-insensitive.

In the function documentation below, the notation `[something]*` means that `something` can be repeated zero or more times. The notation `[something]+` means that the `something` is repeated one or more times (must exist at least one time).

The functions intended for use in Single Function Mode are:

- `capitalize()` – returns the value with the first letter upper case and the rest lower case.
- `contains(pattern, text if match, text if not match)` – checks if the value is matched by the regular expression `pattern`. Returns `text if match` if the pattern matches the value, otherwise returns `text if no match`.
- `count(separator)` – interprets the value as a list of items separated by `separator` and returns the number of items in the list. Most lists use a comma as the separator, but `authors` uses an ampersand (&). Examples: `{tags:count(,)}`, `{authors:count(&)}`. Aliases: `count()`, `list_count()`
- `format_number(template)` – interprets the value as a number and formats that number using a Python formatting template such as `{0:5.2f}` or `{0:,d}` or `${0:5,.2f}`. The formatting template must begin with `{0:` and end with `}` as in the above examples. Exception: you can leave off the leading `<{0:` and trailing `>}` if the format template contains only a format. See the template language and the [Python documentation](#)⁸² for more examples. Returns the empty string if formatting fails.
- `human_readable()` – expects the value to be a number and returns a string representing that number in KB, MB, GB, etc.
- `ifempty(text if empty)` – if the value is not empty then return the value of the field, otherwise return `text if empty`.
- `in_list(separator, [pattern, found_val,]* not_found_val)` – interpret the value as a list of items separated by `separator`, checking the `pattern` against each item in the list. If the `pattern` matches an item then return `found_val`, otherwise return `not_found_val`. The pair `pattern` and `found_value` can be repeated as many times as desired, permitting returning different values depending on the item's value. The patterns are checked in order, and the first match is returned.
- `language_strings(localize)` – return the [language names](#)⁸³ for the [language codes](#)⁸⁴ passed in as the value. Example: `{languages:language_strings()}`. If `localize` is zero, return the strings in English. If `localize` is not zero, return the strings in the language of the current locale. `Lang_codes` is a comma-separated list.
- `list_item(index, separator)` – interpret the value as a list of items separated by `separator`, returning the “index”th item. The first item is number zero. The last item has the index `-1` as in `list_item(-1, separator)`. If the item is not in the list, then the empty string is returned.
- `lookup([pattern, key,]* else_key)` – The patterns will be checked against the value in order. If a pattern matches then the value of the field named by `key` is returned. If no pattern matches then the value of the field named by `else_key` is returned. See “switch” (below).
- `lowercase()` – returns the value of the field in lower case.
- `rating_to_stars(use_half_stars)` – Returns the rating as string of star (★) characters. The value must be a number between 0 and 5. Set `use_half_stars` to 1 if you want half star characters for fractional numbers available with custom ratings columns.

⁸² <https://docs.python.org/3/library/string.html#formatstrings>

⁸³ https://www.loc.gov/standards/iso639-2/php/code_list.php

⁸⁴ https://www.loc.gov/standards/iso639-2/php/code_list.php

- `re(pattern, replacement)` – return the value after applying the regular expression. All instances of `pattern` in the value are replaced with `replacement`. The template language uses case insensitive [Python regular expressions](https://docs.python.org/3/library/re.html)⁸⁵.
- `select(key)` – interpret the value as a comma-separated list of items with each item having the form `id:value` (the calibre identifier format). The function finds the first pair with the `id` equal to `key` and returns the corresponding value. If no `id` matches then the function returns the empty string.
- `shorten(car sin, testo cent, car dest)` – Restituisce una versione accorciata del valore, che consiste dei caratteri `car sin` dell'inizio del valore, seguiti da `testo cent`, seguiti dai caratteri `car dest` della fine del valore. `car sin` e `car dest` devono essere interi non negativi. Esempio: immagina di voler mostrare il titolo con una lunghezza di al più 15 caratteri. Un modello che lo fa è `{title:shorten(9,-,5)}`. Per un libro con il titolo *Antiche Leggi Inglese ai Tempi di Ivanhoe* il risultato sarà *Antiche L-anhoe*: i primi 9 caratteri del titolo, un -, poi gli ultimi 5 caratteri. Se la lunghezza del valore è minore di `car sin + car dest + la lunghezza di testo cent` il valore sarà restituito intatto. Per esempio, il titolo *La Cupola* non verrebbe cambiato.
- `str_in_list(separator, [string, found_val,]+ not_found_val)` – interpret the value as a list of items separated by `separator` then compare `string` against each value in the list. The `string` is not a regular expression. If `string` is equal to any item (ignoring case) then return the corresponding `found_val`. If `string` contains separators then it is also treated as a list and each subvalue is checked. The `string` and `found_value` pairs can be repeated as many times as desired, permitting returning different values depending on `string`'s value. If none of the strings match then `not_found_value` is returned. The strings are checked in order. The first match is returned.
- `subitems(start_index, end_index)` – This function breaks apart lists of tag-like hierarchical items such as genres. It interprets the value as a comma-separated list of tag-like items, where each item is a period-separated list. It returns a new list made by extracting from each item the components from `start_index` to `end_index`, then merging the results back together. Duplicates are removed. The first subitem in a period-separated list has an index of zero. If an index is negative then it counts from the end of the list. As a special case, an `end_index` of zero is assumed to be the length of the list.

Esempi:

- Assuming a `#genre` column containing *A.B.C*:
 - * `{#genre:subitems(0,1)}` restituisce «A»
 - * `{#genre:subitems(0,2)}` restituisce «A.B»
 - * `{#genre:subitems(1,0)}` restituisce «B.C»
- Assuming a `#genre` column containing *«A.B.C, D.E»*:
 - * `{#genre:subitems(0,1)}` restituisce «A, D»
 - * `{#genre:subitems(0,2)}` restituisce «A.B, D.E»
- `sublist(start_index, end_index, separator)` – interpret the value as a list of items separated by `separator`, returning a new list made from the items from `start_index` to `end_index`. The first item is number zero. If an index is negative, then it counts from the end of the list. As a special case, an `end_index` of zero is assumed to be the length of the list.

Examples assuming that the tags column (which is comma-separated) contains «A, B,C»:

- `{tags:sublist(0,1,\,)}` restituisce «A»
- `{tags:sublist(-1,0,\,)}` restituisce «C»
- `{tags:sublist(0,-1,\,)}` restituisce «A, B»

⁸⁵ <https://docs.python.org/3/library/re.html>

- `swap_around_articles(separator)` – returns the value with articles moved to the end. The value can be a list, in which case each item in the list is processed. If the value is a list then you must provide the `separator`. If no `separator` is provided then the value is treated as being a single value, not a list. The *articles* are those used by calibre to generate the `title_sort`.
- `swap_around_comma()` – given a value of the form `B, A`, return `A B`. This is most useful for converting names in LN, FN format to FN LN. If there is no comma in the value then the function returns the value unchanged.
- `switch([pattern, value,]+ else_value)` – for each `pattern, value` pair, checks if the value matches the regular expression `pattern` and if so returns the associated value. If no `pattern` matches, then `else_value` is returned. You can have as many `pattern, value` pairs as you wish. The first match is returned.
- `test(text if not empty, text if empty)` – return `text if not empty` if the value is not empty, otherwise return `text if empty`.
- `titlecase()` – returns the value of the field in title case.
- `transliterate()` – Return a string in a latin alphabet formed by approximating the sound of the words in the source field. For example, if the source field is `Фёдор Михайлович Достоевский` this function returns `Fiodor Mikhailovich Dostoievskii`.
- `uppercase()` – returns the value of the field in upper case.

Using functions and formatting in the same template

Suppose you have an integer custom column `#myint` that you want displayed with leading zeros, as in `003`. One way to do this is to use a format of `0>3s`. However, by default if a number (integer or float) equals zero then the value is displayed as the empty string so zero values will produce the empty string, not `000`. If you want to see `000` values then you use both the format string and the `ifempty` function to change the empty value back to a zero. The template would be:

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

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

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

10.3.6 Modalità Programma Generale

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

```
program          ::= 'program:' expression_list
expression_list  ::= top_expression [ ';' top_expression ]*
top_expression   ::= or_expression
or_expression    ::= and_expression [ '||' and_expression ]*
and_expression   ::= not_expression [ '&&' not_expression ]*
not_expression   ::= [ '!' not_expression ]* | concatenate_expr
concatenate_expr ::= compare_expr [ '&' compare_expr ]*
compare_expr     ::= add_sub_expr [ compare_op add_sub_expr ]
compare_op       ::= '==' | '!=' | '>=' | '>' | '<=' | '<' | 'in' | 'inlist' |
                  '==#' | '!=#' | '>=#' | '>#' | '<=#' | '<#'
add_sub_expr     ::= times_div_expr [ add_sub_op times_div_expr ]*
add_sub_op       ::= '+' | '-'
times_div_expr   ::= unary_op_expr [ times_div_op unary_op_expr ]*
times_div_op     ::= '*' | '/'
```

(continues on next page)

(continua dalla pagina precedente)

```

unary_op_expr    ::= [ add_sub_op unary_op_expr ]* | expression
expression      ::= identifier | constant | function | assignment | field_reference |
                    if_expr | for_expr | break_expr | continue_expr |
                    '(' expression_list ')' | function_def
field_reference  ::= '$' [ '$' ] [ '#' ] identifier
identifier       ::= id_start [ id_rest ]*
id_start        ::= letter | underscore
id_rest         ::= id_start | digit
constant        ::= " string " | ' string ' | number
function        ::= identifier '(' expression_list [ ',' expression_list ]* ')'
function_def     ::= 'def' identifier '(' top_expression [ ',' top_expression ]* ')' ':'
                    ↪ '
                    expression_list 'fed'
assignment      ::= identifier '=' top_expression
if_expr         ::= 'if' condition 'then' expression_list
                    [ elif_expr ] [ 'else' expression_list ] 'fi'
condition       ::= top_expression
elif_expr       ::= 'elif' condition 'then' expression_list elif_expr | ''
for_expr        ::= for_list | for_range
for_list        ::= 'for' identifier 'in' list_expr
                    [ 'separator' separator_expr ] ':' expression_list 'rof'
for_range       ::= 'for' identifier 'in' range_expr ':' expression_list 'rof'
range_expr      ::= 'range' '(' [ start_expr ',' ] stop_expr
                    [ ',' step_expr [ ',' limit_expr ] ] ')'
list_expr       ::= top_expression
break_expr      ::= 'break'
continue_expr   ::= 'continue'
separator_expr  ::= top_expression
start_expr      ::= top_expression
stop_expr       ::= top_expression
step_expr       ::= top_expression
limit_expr      ::= top_expression

```

Note:

- a top_expression always has a value. The value of an expression_list is the value of the last top_expression in the list. For example, the value of the expression list 1; 2; 'foobar'; 3 is 3.
- In a logical context, any non-empty value is True
- In a logical context, the empty value is False
- Strings and numbers can be used interchangeably. For example, 10 and '10' are the same thing.
- Comments are lines starting with a “#” character. Comments beginning later in a line are not supported.

Operator precedence

The operator precedence (order of evaluation) from highest (evaluated first) to lowest (evaluated last) is:

- Function calls, constants, parenthesized expressions, statement expressions, assignment expressions, field references.
- Unary plus (+) and minus (-). These operators evaluate right to left.

These and all the other arithmetic operators return integers if the expression results in a fractional part equal to zero. For example, if an expression returns 3.0 it is changed to 3.

- Multiply (*) and divide (/). These operators are associative and evaluate left to right. Use parentheses if you want to change the order of evaluation.
- Add (+) and subtract (-). These operators are associative and evaluate left to right.

- Numeric and string comparisons. These operators return '1' if the comparison succeeds, otherwise the empty string (''). Comparisons are not associative: `a < b < c` is a syntax error.
- String concatenation (&). The & operator returns a string formed by concatenating the left-hand and right-hand expressions. Example: `'aaa' & 'bbb'` returns `'aaabbb'`. The operator is associative and evaluates left to right.
- Unary logical not (!). This operator returns '1' if the expression is False (evaluates to the empty string), otherwise ''.
- Logical and (&&). This operator returns "1" if both the left-hand and right-hand expressions are True, or the empty string '' if either is False. It is associative, evaluates left to right, and does [short-circuiting](#)⁸⁶.
- 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](#)⁸⁷. It is an *inclusive or*, returning '1' if both the left- and right-hand expressions are True.

Field references

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

```
* $authors ==> field('authors')
* $#genre ==> field('#genre')
* $$pubdate ==> raw_field('pubdate')
* $$#my_int ==> raw_field('#my_int')
```

If expressions

If expressions first evaluate the condition. If the condition is True (a non-empty value) then the `expression_list` in the `then` clause is evaluated. If it is False then if present the `expression_list` in the `elif` or `else` clause is evaluated. The `elif` and `else` parts are optional. The words `if`, `then`, `elif`, `else`, and `fi` are reserved; you cannot use them as identifier names. You can put newlines and white space wherever they make sense. The condition is a `top_expression` not an `expression_list`; semicolons are not allowed. The `expression_lists` are semicolon-separated sequences of `top_expressions`. An if expression returns the result of the last `top_expression` in the evaluated `expression_list`, or the empty string if no expression list was evaluated.

Esempi:

```
* program: if field('series') then 'yes' else 'no' fi
* program:
    if field('series') then
        a = 'yes';
        b = 'no'
    else
        a = 'no';
        b = 'yes'
    fi;
    strcat(a, '-', b)
```

Nested if example:

```
program:
    if field('series') then
        if check_yes_no(field('#mybool'), '', '', '1') then
            'yes'
```

(continues on next page)

⁸⁶ https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html

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

(continua dalla pagina precedente)

```

    else
        'no'
    fi
else
    'no series'
fi

```

As said above, an `if` produces a value. This means that all the following are equivalent:

```

* program: if field('series') then 'foo' else 'bar' fi
* program: if field('series') then a = 'foo' else a = 'bar' fi; a
* program: a = if field('series') then 'foo' else 'bar' fi; a

```

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

```

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

```

For expressions

The `for` expression iterates over a list of values, processing them one at a time. The `list_expression` must evaluate either to a metadata field lookup name e.g., `tags` or `#genre`, or to a list of values. The [range\(\) function](#) (pagina 171) (see below) generates a list of numbers. If the result is a valid lookup name then the field's value is fetched and the separator specified for that field type is used. If the result isn't a valid lookup name then it is assumed to be a list of values. The list is assumed to be separated by commas unless the optional keyword `separator` is supplied, in which case the list values must be separated by the result of evaluating the `separator_expr`. A separator cannot be used if the list is generated by `range()`. Each value in the list is assigned to the specified variable then the `expression_list` is evaluated. You can use `break` to jump out of the loop, and `continue` to jump to the beginning of the loop for the next iteration.

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

```

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

```

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

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

Function definition

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

Arguments are positional. When a function is called the supplied arguments are matched left to right against the defined parameters, with the value of the argument assigned to the parameter. It is an error to provide more arguments than defined

parameters. Parameters can have default values, such as `a = 25`. If an argument is not supplied for that parameter then the default value is used, otherwise the parameter is set to the empty string.

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

A function must be defined before it can be used.

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

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

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

    to_plural(years, 'year') & to_plural(months, 'month') & to_plural(days, 'day')
```

Relational operators

Relational operators return '1' if the comparison is true, otherwise the empty string ("").

There are two forms of relational operators: string comparisons and numeric comparisons.

String comparisons do case-insensitive string comparison using lexical order. The supported string comparison operators are `==`, `!=`, `<`, `<=`, `>`, `>=`, `in`, and `inlist`. For the `in` operator, the result of the left hand expression is interpreted as a regular expression pattern. The `in` operator is True if the value of left-hand regular expression matches the value of the right hand expression. The `inlist` operator is true if the left hand regular expression matches any one of the items in the right hand list where the items in the list are separated by commas. The matches are case-insensitive.

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

Esempi:

- program: `field('series') == 'foo'` returns '1' if the book's series is "foo", otherwise ' '.
- program: `'f.o' in field('series')` returns '1' if the book's series matches the regular expression `f.o` (e.g., *foo*, *Off Onyx*, etc.), otherwise ' '.
- program: `'science' inlist field('#genre')` returns '1' if any of the book's genres match the regular expression `science`, e.g., *Science*, *History of Science*, *Science Fiction* etc.), otherwise ' '.
- program: `'^science$' inlist field('#genre')` returns '1' if any of the book's genres exactly match the regular expression `^science$`, e.g., *Science*. The genres *History of Science* and *Science Fiction* don't match. If there isn't a match then returns ' '.
- program: `if field('series') != 'foo' then 'bar' else 'mumble' fi` returns 'bar' if the book's series is not *foo*. Otherwise it returns 'mumble'.
- program: `if field('series') == 'foo' || field('series') == '1632' then 'yes' else 'no' fi` returns 'yes' if series is either 'foo' or '1632', otherwise 'no'.
- program: `if '^(foo|1632)$' in field('series') then 'yes' else 'no' fi` returns 'yes' if series is either 'foo' or '1632', otherwise 'no'.

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

Additional available functions

The following functions are available in addition to those described in *Single Function Mode* (pagina 156).

In *GPM* the functions described in *Single Function Mode* all require an additional first parameter specifying the value to operate upon. All parameters are `expression_lists` (see the grammar above).

- `add(x [, y]*)` – returns the sum of its arguments. Throws an exception if an argument is not a number. In most cases you can use the `+` operator instead of this function.
- `and(value [, value]*)` – returns the string «1» if all values are not empty, otherwise returns the empty string. You can have as many values as you want. In most cases you can use the `&&` operator instead of this function. One reason not to replace `and` with `&&` is if short-circuiting can change the results because of side effects. For example, `and(a=' ', b=5)` will always do both assignments, where the `&&` operator won't do the second.
- `assign(id, val)` – assigns `val` to `id`, then returns `val`. `id` must be an identifier, not an expression. In most cases you can use the `=` operator instead of this function.
- `approximate_formats()` – return a comma-separated list of formats associated with the book. There is no guarantee that the list is correct, although it probably is. This and other zero-parameter functions can be called in Template Program Mode (see below) using the template `{ : 'approximate_formats()' }`. Note that resulting format names are always uppercase, as in EPUB. The `approximate_formats()` function is significantly faster than the `formats_...` functions discussed below.
- `author_links(val_separator, pair_separator)` – returns a string containing a list of authors and those authors' link values in the form:

```
author1 val_separator author1_link pair_separator author2 val_separator author2_
↪link etc.
```

An author is separated from its link value by the `val_separator` string with no added spaces. `author:linkvalue` pairs are separated by the `pair_separator` string argument with no added spaces. It is up to you to choose separator strings that do not occur in author names or links. An author is included even if the author link is empty.

- `author_sorts(val_separator)` – returns a string containing a list of author's sort values for the authors of the book. The sort is the one in the author metadata information (different from the `author_sort` in books). The returned list has the form `author sort 1 val_separator author sort 2` etc. with no added spaces. The author sort values in this list are in the same order as the authors of the book. If you want spaces around `val_separator` then include them in the `val_separator` string.
- `book_count(query, use_vl)` – returns the count of books found by searching for `query`. If `use_vl` is 0 (zero) then virtual libraries are ignored. This function and its companion `book_values()` are particularly useful in template searches, supporting searches that combine information from many books such as looking for series with only one book. It cannot be used in composite columns unless the `tweak allow_template_database_functions_in_composites` is set to `True`. It can be used only in the GUI.

For example this template search uses this function and its companion to find all series with only one book:

- 1) Define a stored template (using *Preferences* → *Advanced* → *Template functions*) named `series_only_one_book` (the name is arbitrary). The template is:


```

program:
  vals = globals(vals='');
  if !vals then
    all_series = book_values('series', 'series:true', ',', 0);
    for series in all_series:
      if book_count('series="' & series & '"', 0) == 1 then
        vals = list_join(',', vals, ',', series, ',')
      fi
    rof;
    set_globals(vals)
  fi;
  str_in_list(vals, ',', $series, 1, '')

```

The first time the template runs (the first book checked) it stores the results of the database lookups in a global template variable named `vals`. These results are used to check subsequent books without redoing the lookups.

2) Use the stored template in a template search:

```
template:"program: series_only_one_book()#@#:n:1"
```

Using a stored template instead of putting the template into the search eliminates problems caused by the requirement to escape quotes in search expressions.

- `book_values(column, query, sep, use_vl)` – returns a list of the unique values contained in the column `column` (a lookup name), separated by `sep`, in the books found by searching for `query`. If `use_vl` is 0 (zero) then virtual libraries are ignored. This function and its companion `book_count()` are particularly useful in template searches, supporting searches that combine information from many books such as looking for series with only one book. It cannot be used in composite columns unless the tweak `allow_template_database_functions_in_composites` is set to `True`. It can be used only in the GUI.
- `booksize()` – returns the value of the calibre “size” field. Returns “” if there are no formats.
- `check_yes_no(field_name, is_undefined, is_false, is_true)` – checks if the value of the yes/no field named by the lookup name `field_name` is one of the values specified by the parameters, returning 'yes' if a match is found otherwise returning the empty string. Set the parameter `is_undefined`, `is_false`, or `is_true` to 1 (the number) to check that condition, otherwise set it to 0. Example:

`check_yes_no("#bool", 1, 0, 1)` returns 'Yes' if the yes/no field `#bool` is either `True` or `undefined` (neither `True` nor `False`).

More than one of `is_undefined`, `is_false`, or `is_true` can be set to 1.

- `ceiling(x)` – returns the smallest integer greater than or equal to `x`. Throws an exception if `x` is not a number.
- `character(character_name)` – returns the character named by `character_name`. For example, `character('newline')` returns a newline character (`'\n'`). The supported character names are `newline`, `return`, `tab`, and `backslash`.
- `cmp(x, y, lt, eq, gt)` – compares `x` and `y` after converting both to numbers. Returns `lt` if `x <# y`, `eq` if `x ==# y`, otherwise `gt`. This function can usually be replaced with one of the numeric compare operators (`==#`, `<#`, `>#`, etc).
- `connected_device_name(storage_location_key)` – if a device is connected then return the device name, otherwise return the empty string. Each storage location on a device has its own device name. The `storage_location_key` names are `'main'`, `'carda'` and `'cardb'`. This function works only in the GUI.

- `connected_device_uuid(storage_location_key)` – if a device is connected then return the device uuid (unique id), otherwise return the empty string. Each storage location on a device has a different uuid. The `storage_location_key` location names are 'main', 'carda' and 'cardb'. This function works only in the GUI.
 - `current_library_name()` – return the last name on the path to the current calibre library.
 - `current_library_path()` – return the full path to the current calibre library.
 - `current_virtual_library_name()` – restituisce il nome della biblioteca virtuale corrente se ce n'è una, altrimenti la stringa vuota. Le maiuscole del nome della biblioteca sono conservate. Esempio: `program:current_virtual_library_name()`. Questa funzione può essere usata solo nell'interfaccia grafica.
 - `date_arithmetic(date, calc_spec, fmt)` – Calculate a new date from `date` using `calc_spec`. Return the new date formatted according to optional `fmt`: if not supplied then the result will be in ISO format. The `calc_spec` is a string formed by concatenating pairs of `vW` (valueWhat) where `v` is a possibly-negative number and `W` is one of the following letters:
 - s: aggiunge `v` secondi a `date`
 - m: aggiunge `v` minuti a `date`
 - h: aggiunge `v` ore a `date`
 - d: aggiunge `v` giorni a `date`
 - w: aggiunge `v` settimane a `date`
 - y: aggiunge `v` anni a `date`, dove un anno è di 365 giorni.
- Esempio: '1s3d-1m' aggiungerà 1 secondo e 3 giorni, e sottrarrà 1 minuto da `date`.
- `days_between(date1, date2)` – return the number of days between `date1` and `date2`. The number is positive if `date1` is greater than `date2`, otherwise negative. If either `date1` or `date2` are not dates, the function returns the empty string.
 - `divide(x, y)` – returns `x / y`. Throws an exception if either `x` or `y` are not numbers. This function can usually be replaced by the `/` operator.
 - `eval(string)` – evaluates the string as a program, passing the local variables. This permits using the template processor to construct complex results from local variables. In *Template Program Mode* (pagina 172), because the `{` and `}` characters are interpreted before the template is evaluated you must use `[[` for the `{` character and `]]` for the `}` character. They are converted automatically. Note also that prefixes and suffixes (the `|prefix|suffix` syntax) cannot be used in the argument to this function when using *Template Program Mode* (pagina 172).
 - `extra_file_size(file_name)` – returns the size in bytes of the extra file `file_name` in the book's data/ folder if it exists, otherwise -1. See also the functions `has_extra_files()`, `extra_file_names()` and `extra_file_modtime()`. This function can be used only in the GUI.
 - `extra_file_modtime(nome_file, formato_string)` – restituisce la data di modifica del file aggiuntivo `nome_file` nella cartella `data/` del libro se esiste, altrimenti -1. La data di modifica è formattata secondo `formato_string` (vedi `format_date()` per dettagli). Se `formato_string` è la stringa vuota, restituisce la data di modifica come numero di secondi in virgola mobile dall'inizio dell'epoca (epoch). Vedi anche le funzioni `has_extra_files()`, `extra_file_names()` e `extra_file_size()`. L'epoca dipende dal sistema operativo. Questa funzione può essere usata solo nell'interfaccia grafica.
 - `extra_file_names(sep [, pattern])` returns a `sep`-separated list of extra files in the book's data/ folder. If the optional parameter `pattern`, a regular expression, is supplied then the list is filtered to files that match `pattern`. The pattern match is case insensitive. See also the functions `has_extra_files()`, `extra_file_modtime()` and `extra_file_size()`. This function can be used only in the GUI.
 - `field(lookup_name)` – returns the value of the metadata field with lookup name `lookup_name`.

- `field_exists(field_name)` – checks if a field (column) with the lookup name `field_name` exists, returning '1' if so and the empty string if not.
- `finish_formatting(val, fmt, prefix, suffix)` – apply the format, prefix, and suffix to a value in the same way as done in a template like `{series_index:05.2f| - | - }`. This function is provided to ease conversion of complex single-function- or template-program-mode templates to *GPM* Templates. For example, the following program produces the same output as the above template:

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

Another example: for the template `{series:re((^[^s])[^s]+(s|$)),\1)}{series_index:0>2s| - | - }{title}` use:

```
program:
  strcat(
    re(field('series'), '([^\s])^\s+(\s|$)', '\1'),
    finish_formatting(field('series_index'), '0>2s', ' - ', ' - '),
    field('title')
  )
```

- `first_matching_cmp(val, [cmp, result,]* else_result)` – compares `val < cmp` in sequence, returning the associated result for the first comparison that succeeds. Returns `else_result` if no comparison succeeds. Example:

```
i = 10;
first_matching_cmp(i, 5, "small", 10, "middle", 15, "large", "giant")
```

returns "large". The same example with a first value of 16 returns "giant".

- `first_non_empty(value [, value]*)` – returns the first value that is not empty. If all values are empty, then the empty string is returned. You can have as many values as you want.
- `floor(x)` – returns the largest integer less than or equal to `x`. Throws an exception if `x` is not a number.
- `format_date(val, format_string)` – format the value, which must be a date string, using the `format_string`, returning a string. The formatting codes are:

- `d` : the day as number without a leading zero (1 to 31)
- `dd` : the day as number with a leading zero (01 to 31)
- `ddd` : the abbreviated localized day name (e.g. «Mon» to «Sun»).
- `dddd` : the long localized day name (e.g. «Monday» to «Sunday»).
- `M` : the month as number without a leading zero (1 to 12).
- `MM` : the month as number with a leading zero (01 to 12)
- `MMM` : the abbreviated localized month name (e.g. «Jan» to «Dec»).
- `MMMM` : the long localized month name (e.g. «January» to «December»).
- `yy` : the year as two digit number (00 to 99).
- `yyyy` : l'anno come numero di quattro cifre.
- `h` : the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
- `hh` : the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
- `m` : the minutes without a leading 0 (0 to 59)
- `mm` : the minutes with a leading 0 (00 to 59)

- `s` : the seconds without a leading 0 (0 to 59)
- `ss` : the seconds with a leading 0 (00 to 59)
- `ap` : use a 12-hour clock instead of a 24-hour clock, with “ap” replaced by the localized string for am or pm.
- `AP` : use a 12-hour clock instead of a 24-hour clock, with “AP” replaced by the localized string for AM or PM.
- `iso` : the date with time and timezone. Must be the only format present.
- `to_number` : convert the date & time into a floating point number (a *timestamp*)
- `from_number` : convert a floating point number (a *timestamp*) into an `iso` formatted date. If you want a different date format then add the desired formatting string after `from_number` and a colon (:). Example: `from_number:MMM dd yyyy`

You might get unexpected results if the date you are formatting contains localized month names, which can happen if you changed the date format tweaks to contain MMMM. In this case, instead of using the `field()` function as in:

```
format_date(field('pubdate'), 'yyyy')
```

use the `raw_field()` function as in:

```
format_date(raw_field('pubdate'), 'yyyy')
```

- `format_date_field(field_name, format_string)` – format the value in the field `field_name`, which must be the lookup name of date field, either standard or custom. See `format_date()` for the formatting codes. This function is much faster than `format_date` and should be used when you are formatting the value in a field (column). It can’t be used for computed dates or dates in string variables. Examples:

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

- `formats_modtimes(date_format_string)` – return a comma-separated list of colon-separated items `FMT:DATE` representing modification times for the formats of a book. The `date_format_string` parameter specifies how the date is to be formatted. See the `format_date()` function for details. You can use the `select` function to get the modification time for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_paths()` – return a comma-separated list of colon-separated items `FMT:PATH` giving the full path to the formats of a book. You can use the `select` function to get the path for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_sizes()` – return a comma-separated list of colon-separated `FMT:SIZE` items giving the sizes in bytes of the formats of a book. You can use the `select` function to get the size for a specific format. Note that format names are always uppercase, as in EPUB.
- `fractional_part(x)` – returns the value after the decimal point. For example, `fractional_part(3.14)` returns 0.14. Throws an exception if `x` is not a number.
- `has_cover()` – return 'Yes' if the book has a cover, otherwise the empty string.
- `has_extra_files([pattern])` – returns the count of extra files, otherwise “” (the empty string). If the optional parameter `pattern` (a regular expression) is supplied then the list is filtered to files that match `pattern` before the files are counted. The pattern match is case insensitive. See also the functions `extra_file_names()`, `extra_file_size()` and `extra_file_modtime()`. This function can be used only in the GUI.
- `identifier_in_list(val, id_name [, found_val, not_found_val])` – treat `val` as a list of identifiers separated by commas. An identifier has the format `id_name:value`. The `id_name` parameter is the `id_name` text to search for, either `id_name` or `id_name:regexp`. The first case matches if there is any

identifier matching that `id_name`. The second case matches if `id_name` matches an identifier and the regexp matches the identifier's value. If `found_val` and `not_found_val` are provided then if there is a match then return `found_val`, otherwise return `not_found_val`. If `found_val` and `not_found_val` are not provided then if there is a match then return the `identifier:value` pair, otherwise the empty string ('').

- `is_marked()` – check whether the book is *marked* in calibre. If it is then return the value of the mark, either 'true' (lower case) or a comma-separated list of named marks. Returns '' (the empty string) if the book is not marked. This function works only in the GUI.
- `language_codes(lang_strings)` – return the [language codes](#)⁸⁸ for the language names passed in `lang_strings`. The strings must be in the language of the current locale. `Lang_strings` is a comma-separated list.
- `list_contains(value, separator, [pattern, found_val,]* not_found_val)` – (Alias of `in_list`) Interpreting the value as a list of items separated by `separator`, evaluate the `pattern` against each value in the list. If the `pattern` matches any value then return `found_val`, otherwise return `not_found_val`. The `pattern` and `found_value` can be repeated as many times as desired, permitting returning different values depending on the search. The patterns are checked in order. The first match is returned. Aliases: `in_list()`, `list_contains()`
- `list_count(value, separator)` – interprets value as a list of items separated by `separator`, returning the count of items in the list. Aliases: `count()`, `list_count()`
- `list_count_matching(list, pattern, separator)` – interprets `list` as a list of items separated by `separator`, returning the number of items in the list that match the regular expression `pattern`. Aliases: `list_count_matching()`, `count_matching()`
- `list_difference(list1, list2, separator)` – return a list made by removing from `list1` any item found in `list2` using a case-insensitive comparison. The items in `list1` and `list2` are separated by `separator`, as are the items in the returned list.
- `list_equals(list1, sep1, list2, sep2, yes_val, no_val)` – return `yes_val` if `list1` and `list2` contain the same items, otherwise return `no_val`. The items are determined by splitting each list using the appropriate separator character (`sep1` or `sep2`). The order of items in the lists is not relevant. The comparison is case-insensitive.
- `list_intersection(list1, list2, separator)` – return a list made by removing from `list1` any item not found in `list2`, using a case-insensitive comparison. The items in `list1` and `list2` are separated by `separator`, as are the items in the returned list.
- `list_join(with_separator, list1, separator1 [, list2, separator2]*)` – return a list made by joining the items in the source lists (`list1` etc) using `with_separator` between the items in the result list. Items in each source list[123...] are separated by the associated separator[123...]. A list can contain zero values. It can be a field like `publisher` that is single-valued, effectively a one-item list. Duplicates are removed using a case-insensitive comparison. Items are returned in the order they appear in the source lists. If items on lists differ only in letter case then the last is used. All separators can be more than one character.

Esempio:

```
program:
  list_join('#@#', $authors, '&', $tags, ',')
```

You can use `list_join` on the results of previous calls to `list_join` as follows:

```
program:
  a = list_join('#@#', $authors, '&', $tags, ',');
  b = list_join('#@#', a, '@#', $genre, ',', $people, '&', 'some value', ',')
```

⁸⁸ https://www.loc.gov/standards/iso639-2/php/code_list.php

You can use expressions to generate a list. For example, assume you want items for authors and #genre, but with the genre changed to the word «Genre: « followed by the first letter of the genre, i.e. the genre «Fiction» becomes «Genre: F». The following will do that:

```
program:
  list_join('#@#', $authors, '&', list_re($#genre, ',', '^(.)*$', 'Genre: \1'),
  ↪ ',')
```

- `list_re(src_list, separator, include_re, opt_replace)` – Construct a list by first separating `src_list` into items using the `separator` character. For each item in the list, check if it matches `include_re`. If it does then add it to the list to be returned. If `opt_replace` is not the empty string then apply the replacement before adding the item to the returned list.
- `list_re_group(src_list, separator, include_re, search_re [, template_for_group]*)` – Like `list_re` except replacements are not optional. It uses `re_group(item, search_re, template ...)` when doing the replacements.
- `list_remove_duplicates(list, separator)` – return a list made by removing duplicate items in `list`. If items differ only in case then the last is returned. The items in `list` are separated by `separator`, as are the items in the returned list.
- `list_sort(list, direction, separator)` – return `list` sorted using a case-insensitive lexical sort. If `direction` is zero, `list` is sorted ascending, otherwise descending. The list items are separated by `separator`, as are the items in the returned list.
- `list_split(list_val, sep, id_prefix)` – splits `list_val` into separate values using `sep`, then assigns the values to local variables named `id_prefix_N` where `N` is the position of the value in the list. The first item has position 0 (zero). The function returns the last element in the list.

Esempio:

```
list_split('one:two:foo', ':', 'var')
```

è equivalente a:

```
var_0 = 'one';
var_1 = 'two';
var_2 = 'foo'
```

- `list_union(list1, list2, separator)` – return a list made by merging the items in `list1` and `list2`, removing duplicate items using a case-insensitive comparison. If items differ in case, the one in `list1` is used. The items in `list1` and `list2` are separated by `separator`, as are the items in the returned list. Aliases: `merge_lists()`, `list_union()`
- `mod(x, y)` – returns the floor of the remainder of `x / y`. Throws an exception if either `x` or `y` is not a number.
- `multiply(x [, y]*)` – returns the product of its arguments. Throws an exception if any argument is not a number. This function can usually be replaced by the `*` operator.
- `not(value)` – returns the string «1» if the value is empty, otherwise returns the empty string. This function can usually be replaced with the unary `not(!)` operator.
- `ondevice()` – return the string 'Yes' if `ondevice` is set, otherwise return the empty string.
- `or(value [, value]*)` – returns the string '1' if any value is not empty, otherwise returns the empty string. You can have as many values as you want. This function can usually be replaced by the `||` operator. A reason it cannot be replaced is if short-circuiting will change the results because of side effects.

- `print(a [, b]*)` – prints the arguments to standard output. Unless you start calibre from the command line (`calibre-debug -g`), the output will go into a black hole. The `print` function always returns its first argument.
- `range(start, stop, step, limit)` – returns a list of numbers generated by looping over the range specified by the parameters `start`, `stop`, and `step`, with a maximum length of `limit`. The first value produced is “start”. Subsequent values `next_v = current_v + step`. The loop continues while `next_v < stop` assuming `step` is positive, otherwise while `next_v > stop`. An empty list is produced if `start` fails the test: `start >= stop` if `step` is positive. The `limit` sets the maximum length of the list and has a default of 1000. The parameters `start`, `step`, and `limit` are optional. Calling `range()` with one argument specifies `stop`. Two arguments specify `start` and `stop`. Three arguments specify `start`, `stop`, and `step`. Four arguments specify `start`, `stop`, `step` and `limit`. Examples:

```
range(5) -> '0, 1, 2, 3, 4'
range(0, 5) -> '0, 1, 2, 3, 4'
range(-1, 5) -> '-1, 0, 1, 2, 3, 4'
range(1, 5) -> '1, 2, 3, 4'
range(1, 5, 2) -> '1, 3'
range(1, 5, 2, 5) -> '1, 3'
range(1, 5, 2, 1) -> error(limit exceeded)
```

- `raw_field(lookup_name [, optional_default])` – returns the metadata field named by `lookup_name` without applying any formatting. It evaluates and returns the optional second argument `optional_default` if the field's value is undefined (`None`).
- `raw_list(lookup_name, separator)` – returns the metadata list named by `lookup_name` without applying any formatting or sorting, with the items separated by `separator`.
- `re_group(value, pattern [, template_for_group]*)` – return a string made by applying the regular expression `pattern` to `value` and replacing each matched instance with the the value returned by the corresponding template. In *Template Program Mode* (pagina 172), like for the `template` and the `eval` functions, you use `[[for { and]]` for `}`.

The following example looks for a series with more than one word and uppercases the first word:

```
program: re_group(field('series'), "(\\S* )(\\S*)", "{$:uppercase()}", "{$}")'
```

- `round(x)` – returns the nearest integer to `x`. Throws an exception if `x` is not a number.
- `series_sort()` – returns the series sort value.
- `strcat(a [, b]*)` – can take any number of arguments. Returns a string formed by concatenating all the arguments.
- `strcat_max(max, string1 [, prefix2, string2]*)` – Returns a string formed by concatenating the arguments. The returned value is initialized to `string1`. Strings made from `prefix`, `string` pairs are added to the end of the value as long as the resulting string length is less than `max`. Prefixes can be empty. Returns `string1` even if `string1` is longer than `max`. You can pass as many `prefix`, `string` pairs as you wish.
- `strcmp(x, y, lt, eq, gt)` – does a case-insensitive lexical comparison of `x` and `y`. Returns `lt` if `x < y`, `eq` if `x == y`, otherwise `gt`. This function can often be replaced by one of the lexical comparison operators (`==`, `>`, `<`, etc.).
- `strcmpcase(x, y, lt, eq, gt)` – does a case-sensitive lexical comparison of `x` and `y`. Returns `lt` if `x < y`, `eq` if `x == y`, otherwise `gt`.

Note: This is NOT the default behavior used by calibre, for example, in the lexical comparison operators (`==`, `>`, `<`, etc.). This function could cause unexpected results, preferably use `strcmp()` whenever possible.

- `strlen(value)` – Returns the length of the string value.

- `substr(str, start, end)` – returns the start'th through the end'th characters of `str`. The first character in `str` is the zero'th character. If `end` is negative, then it indicates that many characters counting from the right. If `end` is zero, then it indicates the last character. For example, `substr('12345', 1, 0)` returns '2345', and `substr('12345', 1, -1)` returns '234'.
- `subtract(x, y)` – returns `x - y`. Throws an exception if either `x` or `y` are not numbers. This function can usually be replaced by the `-` operator.
- `switch_if([test_expression, value_expression,]+ else_expression)` – for each `test_expression, value_expression` pair, checks if `test_expression` is `True` (non-empty) and if so returns the result of `value_expression`. If no `test_expression` is `True` then the result of `else_expression` is returned. You can have as many `test_expression, value_expression` pairs as you want.
- `today()` – return a date+time string for today (now). This value is designed for use in *format_date* or *days_between*, but can be manipulated like any other string. The date is in ISO⁸⁹ date/time format.
- `template(x)` – evaluates `x` as a template. The evaluation is done in its own context, meaning that variables are not shared between the caller and the template evaluation.
- `to_hex(val)` – returns the string `val` encoded in hex. This is useful when constructing calibre URLs.
- `urls_from_identifiers(identifiers, sort_results)` – given a comma-separated list of `identifiers`, where an *identifier* is a colon-separated pair of values (`id_name:id_value`), returns a comma-separated list of HTML URLs generated from the identifiers. The list not sorted if `sort_results` is 0 (character or number), otherwise it is sorted alphabetically by the identifier name. The URLs are generated in the same way as the built-in identifiers column when shown in *Book details*.

10.3.7 Programmi più complessi nelle espressioni di modello - Modalità Programma di Modello

Template Program Mode (TPM) is a blend of *General Program Mode* (pagina 159) and *Single Function Mode* (pagina 156). *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* (pagina 156) because you cannot make reference to another metadata field within a template expression. In *TPM* you can, as the following expression demonstrates:

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

The example shows several things:

- *TPM* is used if the expression begins with `:` and ends with `}`. Anything else is assumed to be in *Single Function Mode* (pagina 156).
- the variable `$` stands for the field named in the template: the expression is operating upon, `#series` in this case.
- functions must be given all their arguments. There is no default value. For example, the standard built-in functions must be given an additional initial parameter indicating the source field.
- white space is ignored and can be used anywhere within the expression.
- constant strings are enclosed in matching quotes, either `'` or `"`.

⁸⁹ https://en.wikipedia.org/wiki/ISO_8601

All the functions listed under *Single Function Mode* and *General Program Mode* can be used in *TPM*.

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

As with *General Program Mode*, for functions documented under *Single Function Mode* (pagina 156) you must supply the value the function is to act upon as the first parameter in addition to the documented parameters. In *TPM* you can use \$ to access the value specified by the `lookup` name for the template expression.

10.3.8 Python Template Mode

Python Template Mode (PTM) lets you write templates using native python and the [calibre API](#)⁹⁰. The database API will be of most use; further discussion is beyond the scope of this manual. PTM templates are faster and can do more complicated operations but you must know how to write code in python using the calibre API.

A PTM template begins with:

```
python:
def evaluate(book, context):
    # book is a calibre metadata object
    # context is an instance of calibre.utils.formatter.PythonTemplateContext,
    # which currently contains the following attributes:
    # db: a calibre legacy database object.
    # globals: the template global variable dictionary.
    # arguments: is a list of arguments if the template is called by a GPM template,
    ↪ otherwise None.
    # funcs: used to call Built-in/User functions and Stored GPM/Python templates.
    # Example: context.funcs.list_re_group()

    # your Python code goes here
    return 'a string'
```

You can add the above text to your template using the context menu, usually accessed with a right click. The comments are not significant and can be removed. You must use python indenting.

The context object supports `str(context)` that returns a string of the context's contents, and `context.attributes` that returns a list of the attribute names in the context.

The `context.funcs` attribute allows calling Built-in and User template functions, and Stored GPM/Python templates, so that you can execute them directly in your code. The functions are retrieved using their names. If the name conflicts with a Python keyword, add an underscore to the end of the name. Examples:

```
context.funcs.list_re_group()
context.funcs.assert_()
```

Here is an example of a PTM template that produces a list of all the authors for a series. The list is stored in a *Column built from other columns, behaves like tags*. It shows in *Book details* and has the *on separate lines* checked (in *Preferences* → *Look & feel* → *Book details*). That option requires the list to be comma-separated. To satisfy that requirement the template converts commas in author names to semicolons then builds a comma-separated list of authors. The authors are then sorted, which is why the template uses `author_sort`.

```
python:
def evaluate(book, context):
```

(continues on next page)

⁹⁰ <https://manual.calibre-ebook.com/develop.html#api-documentation-for-various-parts-of-calibre>

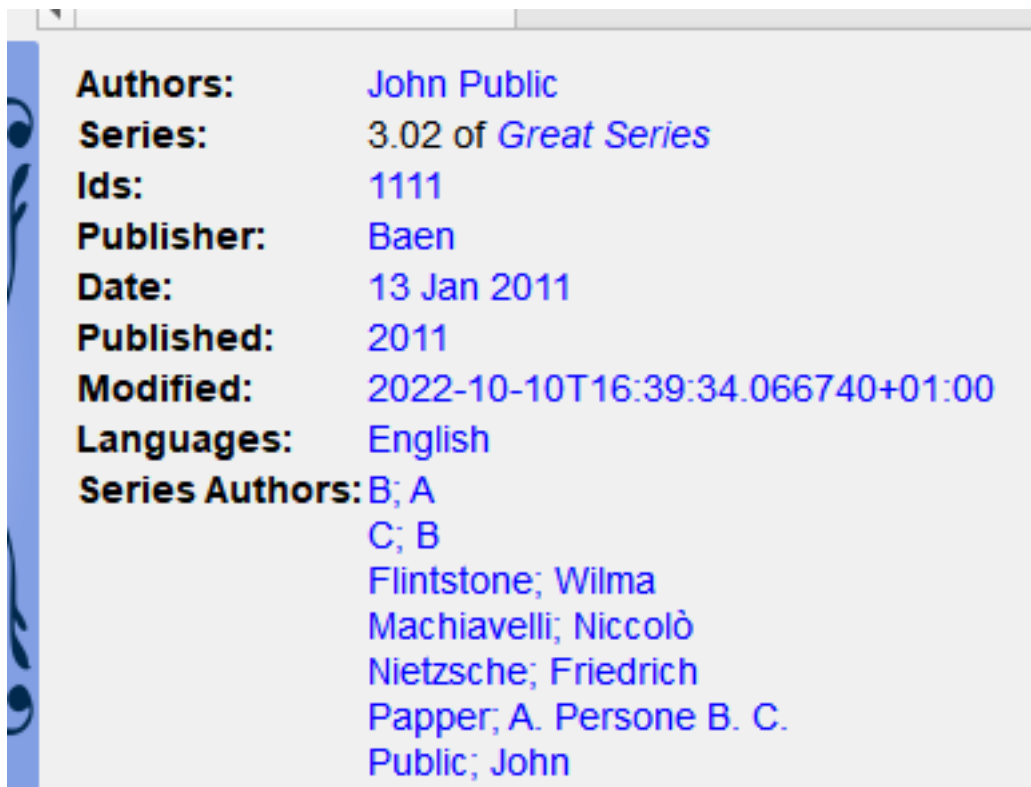
(continua dalla pagina precedente)

```

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 Modelli salvati

Both *General Program Mode* (pagina 159) and *Python Template Mode* (pagina 173) 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 `f00`:

- `f00()` – call the template passing no arguments.
- `f00(a, b)` call the template passing the values of the two variables `a` and `b`.

- `foo(if field('series') then field('series_index') else 0 fi)` – if the book has a series then pass the `series_index`, otherwise pass the value 0.

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

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

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

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

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

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

10.3.10 Fornire informazioni aggiuntive ai modelli

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

Developer: how to pass additional information

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

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

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

Template writer: how to access the additional information

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

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

where `id` is any legal variable name. This function checks whether the additional information provided by the developer contains the name. If it does then the function assigns the provided value to a template local variable with that name. If

the name is not in the additional information and if an `expression` is provided, the `expression` is evaluated and the result is assigned to the local variable. If neither a value nor an expression is provided, the function assigns the empty string (`' '`) to the local variable.

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

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

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

10.3.11 Note sulla differenza tra le modalità

The three program modes, *Single Function Mode* (pagina 156) (SFM), *Template Program Mode* (pagina 172) (TPM), and *General Program Mode* (pagina 159) (GPM), work differently. SFM is intended to be “simple” so it hides a lot of programming language bits.

Differenze:

- 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* (pagina 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*.

10.3.12 Funzioni di modello Python definite dall'utente

You can add your own Python functions to the template processor. Such functions can be used in any of the three template programming modes. The functions are added by going to *Preferences* → *Advanced* → *Template functions*. Instructions are shown in that dialog.

10.3.13 Note particolari per i modelli di salvataggio/invio

Special processing is applied when a template is used in a *save to disk* or *send to device* template. The values of the fields are cleaned, replacing characters that are special to file systems with underscores, including slashes. This means that field text cannot be used to create folders. However, slashes are not changed in prefix or suffix strings, so slashes in these strings will cause folders to be created. Because of this, you can create variable-depth folder structure.

For example, assume we want the folder structure *series/series_index - title*, with the caveat that if series does not exist, then the title should be in the top folder. The template to do this is:

```
{series:||/}{series_index:|| - }{title}
```

The slash and the hyphen appear only if series is not empty.

The lookup function lets us do even fancier processing. For example, assume that if a book has a series, then we want the folder structure *series/series_index - title.fmt*. If the book does not have a series then we want the folder structure *genre/author_sort/title.fmt*. If the book has no genre then we want to use “Unknown”. We want two completely different paths, depending on the value of series.

To accomplish this, we:

1. Create a composite field (give it lookup name #aa) containing `{series}/{series_index} - {title}`. If the series is not empty, then this template will produce *series/series_index - title*.
2. Create a composite field (give it lookup name #bb) containing `{#genre:ifempty(Unknown)}/{author_sort}/{title}`. This template produces *genre/author_sort/title*, where an empty genre is replaced with *Unknown*.
3. Set the save template to `{series:lookup(., #aa, #bb)}`. This template chooses composite field #aa if series is not empty and composite field #bb if series is empty. We therefore have two completely different save paths, depending on whether or not *series* is empty.

10.3.14 Suggerimenti

- Use the Template Tester to test templates. Add the tester to the context menu for books in the library and/or give it a keyboard shortcut.
- Templates can use other templates by referencing composite columns built with the desired template. Alternatively, you can use Stored Templates.
- In a plugboard, you can set a field to empty (or whatever is equivalent to empty) by using the special template `{ }`. This template will always evaluate to an empty string.
- The technique described above to show numbers even if they have a zero value works with the standard field *series_index*.

10.3.15 Riferimenti delle funzioni

Reference for all built-in template language functions

Here, we document all the built-in functions available in the calibre template language. Every function is implemented as a class in python and you can click the source links to see the source code, in case the documentation is insufficient. The functions are arranged in logical groups by type.

- *Arithmetic* (pagina 181)
 - *add*(*x* [, *y*]*) (pagina 181)
 - *ceiling*(*x*) (pagina 182)
 - *divide*(*x*, *y*) (pagina 182)
 - *floor*(*x*) (pagina 182)
 - *fractional_part*(*x*) (pagina 182)
 - *mod*(*x*) (pagina 182)
 - *multiply*(*x* [, *y*]*) (pagina 182)
 - *round*(*x*) (pagina 182)
 - *subtract*(*x*, *y*) (pagina 183)
- *Boolean* (pagina 183)
 - *and*(*valore* [, *valore*]*) (pagina 183)
 - *not*(*valore*) (pagina 183)
 - *or*(*valore* [, *valore*]*) (pagina 183)
- *Date functions* (pagina 183)
 - *days_between*(*data1*, *data2*) (pagina 183)
 - *today*() (pagina 183)
- *Formatting values* (pagina 184)
 - *finish_formatting*(*val*, *fmt*, *prefisso*, *suffisso*) (pagina 184)
 - *format_date_field*(*nome_campo*, *formato_stringa*) (pagina 184)
 - *format_number*(*v*, *template*) (pagina 184)
 - *human_readable*(*v*) (pagina 184)
 - *rating_to_stars*(*value*, *use_half_stars*) (pagina 184)
 - *urls_from_identifiers*(*identificatori*, *ordina_risultati*) (pagina 185)
- *Get values from metadata* (pagina 185)
 - *annotation_count*() (pagina 185)
 - *approximate_formats*() (pagina 185)
 - *author_links*(*separatore_val*, *separatore_coppia*) (pagina 185)
 - *author_sorts*(*separatore_val*) (pagina 185)
 - *booksize*() (pagina 186)

- *connected_device_name(posizione_memoria)* (pagina 186)
- *connected_device_uuid(posizione_memoria)* (pagina 186)
- *current_library_name()* (pagina 186)
- *current_library_path()* (pagina 186)
- *current_virtual_library_name()* (pagina 186)
- *field(nome_di_riferimento)* (pagina 187)
- *formats_modtimes(formato_data)* (pagina 187)
- *formats_paths()* (pagina 187)
- *formats_sizes()* (pagina 187)
- *has_cover()* (pagina 187)
- *is_marked()* (pagina 187)
- *language_codes(stringhe_lingua)* (pagina 188)
- *language_strings(codici_lingua, localizza)* (pagina 188)
- *ondevice()* (pagina 188)
- *raw_field(nome_di_riferimento [, optional_default])* (pagina 188)
- *raw_list(nome_di_riferimento, separatore)* (pagina 188)
- *series_sort()* (pagina 188)
- *user_categories()* (pagina 189)
- *virtual_libraries()* (pagina 189)
- *If-then-else* (pagina 189)
 - *contains(val, schema, testo con risultati, testo senza risultati)* (pagina 189)
 - *field_exists(nome_campo)* (pagina 189)
 - *ifempty(val, text if empty)* (pagina 189)
 - *test(val, testo se non vuoto, testo se vuoto)* (pagina 189)
- *Iterating over values* (pagina 190)
 - *first_non_empty(valore [, valore]*)* (pagina 190)
 - *lookup(val, [schema, campo,]+ campo_alternativo)* (pagina 190)
 - *switch(val, [schema, valore,]+ valore_alternativo)* (pagina 190)
 - *switch_if([test_expression, value_expression,]+ else_expression)* (pagina 190)
- *List lookup* (pagina 190)
 - *identifier_in_list(val, nome_id [, val_trovato, val_non_trovato])* (pagina 190)
 - *in_list(val, separator, [pattern, found_val,]+ not_found_val)* (pagina 191)
 - *list_item(val, indice, separatore)* (pagina 191)
 - *select(val, key)* (pagina 191)
 - *str_in_list(val, separator, [string, found_val,]+ not_found_val)* (pagina 191)

- *List manipulation* (pagina 191)
 - *count(val, separatore)* (pagina 191)
 - *list_count_matching(list, pattern, separator)* (pagina 192)
 - *list_difference(lista1, lista2, separatore)* (pagina 192)
 - *list_equals(lista1, sep1, lista2, sep2, val_si, val_no)* (pagina 192)
 - *list_intersection(lista1, lista2, separatore)* (pagina 192)
 - *list_join(with_separator, list1, separator1 [, list2, separator2]*)* (pagina 192)
 - *list_re(lista_sorg, separatore, includi_re, sost_opz)* (pagina 193)
 - *list_re_group(src_list, separator, include_re, search_re [, group_template]+)* (pagina 193)
 - *list_remove_duplicates(lista, separatore)* (pagina 193)
 - *list_sort(lista, direzione, separatore)* (pagina 193)
 - *list_split(elenca_val, sep, prefisso_id)* (pagina 193)
 - *list_union(lista1, lista2, separatore)* (pagina 194)
 - *range(start, stop, passo, limite)* (pagina 194)
 - *subitems(val, start_index, end_index)* (pagina 194)
 - *sublist(val, start_index, end_index, separator)* (pagina 194)
- *Other* (pagina 195)
 - *arguments(id[=expression] [, id[=expression]]*)* (pagina 195)
 - *assign(id, val)* (pagina 195)
 - *globals(id[=expression] [, id[=expression]]*)* (pagina 195)
 - *print(a[, b]*)* (pagina 195)
- *Recursion* (pagina 195)
 - *eval(modello)* (pagina 195)
 - *template(x)* (pagina 196)
- *Relational* (pagina 196)
 - *cmp(x, y, lt, eq, gt)* (pagina 196)
 - *first_matching_cmp(val, [cmp1, result1,]+, else_result)* (pagina 196)
 - *strcmp(x, y, lt, eq, gt)* (pagina 196)
 - *strcmpcase(x, y, lt, eq, gt)* (pagina 196)
- *String case changes* (pagina 197)
 - *capitalize(val)* (pagina 197)
 - *lowercase(val)* (pagina 197)
 - *titlecase(val)* (pagina 197)
 - *uppercase(val)* (pagina 197)
- *String manipulation* (pagina 197)

- *character(nome_carattere)* (pagina 197)
- *re(val, pattern, replacement)* (pagina 197)
- *re_group(val, schema [, modello_del_gruppo]*)* (pagina 198)
- *shorten(val, car sin, testo cent, car dest)* (pagina 198)
- *strcat(a [, b]*)* (pagina 198)
- *strcat_max(max, stringa1 [, prefisso2, stringa2]*)* (pagina 198)
- *strlen(a)* (pagina 198)
- *substr(str, inizio, fine)* (pagina 199)
- *swap_around_articles(val, separatore)* (pagina 199)
- *swap_around_comma(val)* (pagina 199)
- *to_hex(val)* (pagina 199)
- *transliterate(a)* (pagina 199)
- *Template database functions* (pagina 199)
 - *book_count(query, use_vl)* (pagina 199)
 - *book_values(column, query, sep, use_vl)* (pagina 200)
 - *extra_file_modtime(nome_file, formato_stringa)* (pagina 200)
 - *extra_file_names(sep [, schema])* (pagina 200)
 - *extra_file_size(nome_file)* (pagina 200)
 - *has_extra_files([schema])* (pagina 200)
- *other* (pagina 200)
 - *set_globals(id[=expression] [, id[=expression]])* (pagina 200)
- *API of the Metadata objects* (pagina 201)

Arithmetic

add(x [, y]*)

class calibre.utils.formatter_functions.BuiltinAdd

add(x [, y])* – restituisce la somma dei suoi argomenti. Genera un’eccezione se un argomento non è un numero. Questa funzione può spesso essere sostituita dall’operatore +.

ceiling(x)

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

ceiling(x) – restituisce il più piccolo intero maggiore di o uguale a x. Genera un’eccezione se x non è un numero.

divide(x, y)

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

divide(x, y) – restituisce x / y . Genera un’eccezione se uno tra x e y non è un numero. Questa funzione può spesso essere sostituita dall’operatore `/`.

floor(x)

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

floor(x) – restituisce il più grande intero minore di o uguale a x. Genera un’eccezione se x non è un numero.

fractional_part(x)

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

fractional_part(x) – restituisce il valore dopo la virgola. Per esempio `fractional_part(3.14)` restituisce 0.14. Genera un’eccezione se x non è un numero.

mod(x)

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

mod(x) – restituisce floor(resto di x / y). Genera un’eccezione se x oppure y non sono numeri.

multiply(x [, y]*)

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

multiply(x [, y]*) – restituisce il prodotto dei suoi argomenti. Genera un’eccezione se uno dei suoi argomenti non è un numero. Questa funzione può spesso essere sostituita dall’operatore `*`.

round(x)

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

round(x) – restituisce l’intero più vicino a x. Genera un’eccezione se x non è un numero.

subtract(x, y)

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

subtract(x, y) – restituisce x - y. Genera un'eccezione se uno tra x e y non è un numero. Questa funzione può spesso essere sostituita dall'operatore -.

Boolean**and(valore [, valore]*)**

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

and(valore [, valore]*) – restituisce la stringa «1» se tutti i valori non sono vuoti, altrimenti restituisce la stringa vuota. Questa funzione lavora correttamente con test o first_non_empty. Puoi avere quanti valori desideri. In molti casi l'operatore && può sostituire questa funzione.

not(valore)

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

not(valore) – restituisce la stringa «1» se il valore è vuoto, altrimenti restituisce la stringa vuota. Questa funzione lavora correttamente con test o first_non_empty. In molti casi l'operatore ! può sostituire questa funzione.

or(valore [, valore]*)

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

or(valore [, valore]*) – restituisce la stringa «1» se un qualunque valore non è vuoto, altrimenti restituisce la stringa vuota. Questa funzione lavora correttamente con test o first_non_empty. Puoi avere quanti valori desideri. In molti casi l'operatore || può sostituire questa funzione.

Date functions**days_between(data1, data2)**

class calibre.utils.formatter_functions.**BuiltinDaysBetween**

days_between(data1, data2) – restituisce il numero di giorni tra data1 e data2. Il numero è positivo se data1 è maggiore di data2, altrimenti è negativo. Se data1 o data2 non sono date, la funzione restituisce la stringa vuota.

today()

class calibre.utils.formatter_functions.**BuiltinToday**

today() – restituisce una stringa della data odierna. Questo valore è pensato per l'utilizzo in format_date o days_between, ma può essere manipolato come qualsiasi altra stringa. La data è espressa in formato ISO.

Formatting values

`finish_formatting(val, fmt, prefisso, suffisso)`

class `calibre.utils.formatter_functions.BuiltinFinishFormatting`

`finish_formatting(val, fmt, prefisso, suffisso)` – applica formato, prefisso e suffisso a un valore nello stesso modo in cui viene fatto in un modello come `{series_index:05.2f} - | - |`. Per esempio, il seguente programma produce lo stesso output del modello precedente: `program: finish_formatting(field(«series_index»), «05.2f», « - », « - »)`

`format_date_field(nome_campo, formato_stringa)`

class `calibre.utils.formatter_functions.BuiltinFormatDateField`

`format_date_field(nome_campo, formato_stringa)` – formatta il valore nel campo “nome_campo”, che deve essere il nome di riferimento di un campo date, che sia predefinito o personalizzato. Vedi “format_date” per i codici di formattazione. Questa funzione è molto più veloce di `format_date` e dovrebbe essere usata quando vuoi formattare il valore di un campo (colonna). Non può essere usata per date calcolate o date presenti in variabili di tipo stringa. Esempio: `format_date_field(“pubdate”, “yyyy.MM.dd”)`

`format_number(v, template)`

class `calibre.utils.formatter_functions.BuiltinFormatNumber`

`format_number(v, template)` – formatta il numero `v` usando uno schema di formattazione in Python come «`{0:5.2f}`» o «`{0:d}`» o «`}${0:5.2f}`». La parte `field_name` dello schema deve essere uno 0 (zero) (lo «`{0:}`» negli esempi precedenti). Vedi il linguaggio dei modelli e la documentazione di Python per più esempi. Puoi tralasciare lo «`{0:}`» iniziale e il «`}`» finale se lo schema contiene solo un formato. Restituisce la stringa vuota se la formattazione fallisce.

`human_readable(v)`

class `calibre.utils.formatter_functions.BuiltinHumanReadable`

`human_readable(v)` – restituisce una stringa che rappresenta il numero `v` in KB, MB, GB, ecc.

`rating_to_stars(value, use_half_stars)`

class `calibre.utils.formatter_functions.BuiltinRatingToStars`

`rating_to_stars(value, use_half_stars)` – Restituisce la valutazione come stringa di caratteri stella. `value` è un numero tra 0 e 5. Imposta `use_half_stars` a 1 se vuoi caratteri mezza stella per colonne di valutazione personalizzate che supportano valutazioni non intere, come 2.5.

urls_from_identifiers(identificatori, ordina_risultati)

class calibre.utils.formatter_functions.**BuiltinUrlsFromIdentifiers**

urls_from_identifiers(identificatori, ordina_risultati) – dato un elenco separato da virgole di identificatori, dove un identificatore è una coppia di valori separati dai due punti (nome:valore_id), restituisce un elenco separato da virgole di URL HTML generati dagli identificatori. L'elenco non viene riordinato se ordina_risultati è 0 (carattere o numero), viceversa è presentato in ordine alfabetico secondo il nome dell'identificatore. Gli URL sono generati nello stesso modo delle colonne di identificatori predefinite, quando sono mostrati nei Dettagli libro.

Get values from metadata**annotation_count()**

class calibre.utils.formatter_functions.**BuiltinAnnotationCount**

annotation_count() – restituisce il numero totale di annotazioni di ogni tipo allegate al libro corrente. Questa funzione può essere usata solo nell'interfaccia grafica.

approximate_formats()

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

approximate_formats() – restituisce una lista separata da virgole di formati che in un dato momento erano associati al libro. Non c'è garanzia che questa lista sia corretta, sebbene probabilmente sia così. Questa funzione può essere invocata in modalità programma di modello usando lo schema «{:”approximate_formats()”}». Tieni presente che i nomi di formato sono sempre in maiuscolo, come in EPUB. Questa funzione può essere usata solo nell'interfaccia grafica. Se vuoi usare questi valori negli schemi di salvataggio su disco o di invio al dispositivo devi creare una «Colonna costruita da altre colonne» personalizzata, usare la funzione dello schema di quella colonna, e usare il valore di quella colonna nei tuoi schemi di salvataggio/invio

author_links(separatore_val, separatore_coppia)

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

author_links(separatore_val, separatore_coppia) – restituisce una stringa contenente un elenco di autori e dei rispettivi collegamenti autori, nella forma autore1 separatore_val collegamentoautore1 separatore_coppia autore2 separatore_val collegamentoautore2 ecc. Un autore è separato dal suo collegamento dalla stringa separatore_val senza spazi aggiuntivi, mentre le coppie autore:collegamento sono separate dalla stringa separatore_coppia senza spazi aggiuntivi. Sta a te scegliere le stringhe dei separatori in modo che non siano presenti nei nomi degli autori o nei collegamenti. Un autore viene incluso anche se il suo collegamento autore è vuoto.

author_sorts(separatore_val)

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

author_sorts(separatore_val) – restituisce una stringa contenente un elenco di valori di ordinamento autore per gli autori del libro. L'ordinamento è quello del metadato autore (diverso dall'author_sort dei libri). L'elenco è restituito nella forma ordinamento autore 1 separatore_val ordinamento autore 2 ecc. I valori di ordinamento autore di questo elenco sono nello stesso ordine degli autori del libro. Se desideri avere spazi attorno a separatore_val includili nella stringa del separatore

booksize()

class calibre.utils.formatter_functions.BuiltinBooksize

booksize() – restituisce il valore del campo dimensione. Questa funzione può essere usata solo nell'interfaccia grafica. Se vuoi usare questo valore negli schemi di salvataggio su disco o di invio al dispositivo devi creare una «Colonna costruita da altre colonne» personalizzata, usare la funzione dello schema di quella colonna, e usare il valore di quella colonna nei tuoi schemi di salvataggio/invio

connected_device_name(posizione_memoria)

class calibre.utils.formatter_functions.BuiltinConnectedDeviceName

connected_device_name(posizione_memoria) – se un dispositivo è connesso restituisce il suo nome, altrimenti restituisce la stringa vuota. Ognuna delle posizioni di memoria di un dispositivo può avere un nome differente. I nomi delle posizioni sono “main”, “carda” e “cardb”. Questa funzione può essere usata solo nell'interfaccia grafica.

connected_device_uuid(posizione_memoria)

class calibre.utils.formatter_functions.BuiltinConnectedDeviceUUID

connected_device_uuid(posizione_memoria) – se un dispositivo è connesso restituisce il suo uuid (id univoco), altrimenti restituisce la stringa vuota. Ognuna delle posizioni di memoria di un dispositivo ha un uuid differente. I nomi delle posizioni sono “main”, “carda” e “cardb”. Questa funzione può essere usata solo nell'interfaccia grafica.

current_library_name()

class calibre.utils.formatter_functions.BuiltinCurrentLibraryName

current_library_name() – restituisce l'ultimo nome nel percorso della biblioteca di calibre corrente. Questa funzione può essere invocata in modalità programma di modello usando lo schema «{:"current_library_name()"}».

current_library_path()

class calibre.utils.formatter_functions.BuiltinCurrentLibraryPath

current_library_path() – restituisce il percorso della biblioteca di calibre corrente. Questa funzione può essere invocata in modalità programma di modello usando lo schema «{:"current_library_path()"}».

current_virtual_library_name()

class calibre.utils.formatter_functions.BuiltinCurrentVirtualLibraryName

current_virtual_library_name() – restituisce il nome della biblioteca virtuale attuale se ce n'è una, altrimenti la stringa vuota. Le maiuscole del nome della biblioteca sono conservate. Esempio: «program:current_virtual_library_name()».

field(nome_di_riferimento)

class calibre.utils.formatter_functions.**BuiltinField**

field(nome_di_riferimento) – restituisce il campo metadati identificato da nome_di_riferimento

formats_modtimes(formato_data)

class calibre.utils.formatter_functions.**BuiltinFormatsModtimes**

formats_modtimes(formato_data) – restituisce una lista separata da virgole di elementi separati da due punti (:) che rappresentano le date di modifica dei formati di un libro. Il parametro formato_data specifica quale sarà il formato della data. Vedi la funzione format_date per dettagli. Puoi usare la funzione select per ottenere la data di modifica di un formato specifico. Tieni presente che i nomi dei formati sono sempre in maiuscolo, come in EPUB.

formats_paths()

class calibre.utils.formatter_functions.**BuiltinFormatsPaths**

formats_paths() – restituisce una lista separata da virgole di elementi separati da due punti (:) che rappresentano il percorso completo ai formati di un libro. Puoi usare la funzione select per ottenere il percorso di un formato specifico. Tieni presente che i nomi di formato sono sempre in maiuscolo, come in EPUB.

formats_sizes()

class calibre.utils.formatter_functions.**BuiltinFormatsSizes**

formats_sizes() – restituisce una lista separata da virgole di elementi separati da due punti (:) che rappresentano la dimensione in byte dei formati di un libro. Puoi usare la funzione select per ottenere la dimensione di un formato specifico. Tieni presente che i nomi di formato sono sempre in maiuscolo, come in EPUB.

has_cover()

class calibre.utils.formatter_functions.**BuiltinHasCover**

has_cover() – restituisce Yes se il libro ha una copertina, altrimenti restituisce una stringa vuota

is_marked()

class calibre.utils.formatter_functions.**BuiltinIsMarked**

is_marked() – controlla se il libro è “marcato” in calibre. Se lo è restituisce il valore della marcatura, o “true” o la lista separata da virgole di marcature con nome. Restituisce “” se il libro non è marcato.

language_codes(stringhe_lingua)

class calibre.utils.formatter_functions.**BuiltinLanguageCodes**

language_codes(stringhe_lingua) – restituisce i codici lingua per le stringhe fornite in stringhe_lingua. Le stringhe devono essere nella lingua corrente di calibre. stringhe_lingua è una lista separata da virgole.

language_strings(codici_lingua, localizza)

class calibre.utils.formatter_functions.**BuiltinLanguageStrings**

language_strings(codici_lingua, localizza) – restituisce le stringhe dei codici lingua forniti in codici_lingua. Se localizza è zero, restituisce le stringhe in inglese. Se localizza non è zero, restituisce le stringhe nella lingua corrente. codici_lingua è una lista separata da virgole.

ondevice()

class calibre.utils.formatter_functions.**BuiltinOndevice**

ondevice() – restituisce Yes se ondevice è impostato, altrimenti restituisce la stringa vuota. Questa funzione può essere usata solo nell'interfaccia grafica. Se vuoi usare questo valore negli schemi di salvataggio su disco o di invio al dispositivo devi creare una «Colonna costruita da altre colonne» personalizzata, usare la funzione dello schema di quella colonna, e usare il valore di quella colonna nei tuoi schemi di salvataggio/invio

raw_field(nome_di_riferimento [, optional_default])

class calibre.utils.formatter_functions.**BuiltinRawField**

raw_field(nome_di_riferimento [, optional_default]) – restituisce il campo metadati identificato da nome_di_riferimento senza applicare alcuna formattazione. Valuta e restituisce il secondo argomento opzionale “default” se il campo non è definito (“None”).

raw_list(nome_di_riferimento, separatore)

class calibre.utils.formatter_functions.**BuiltinRawList**

raw_list(nome_di_riferimento, separatore) – restituisce la lista di metadati identificata da nome_di_riferimento senza applicare alcuna formattazione od ordinamento e con gli elementi separati dal separatore.

series_sort()

class calibre.utils.formatter_functions.**BuiltinSeriesSort**

series_sort() – restituisce il valore di ordinamento delle serie

user_categories()

class calibre.utils.formatter_functions.BuiltinUserCategories

user_categories() – restituisce una lista separata da virgole delle categorie utente che contengono questo libro. Questa funzione può essere usata solo nell'interfaccia grafica. Se vuoi usare questi valori negli schemi di salvataggio su disco o di invio al dispositivo devi creare una «Colonna costruita da altre colonne» personalizzata, usare la funzione dello schema di quella colonna, e usare il valore di quella colonna nei tuoi schemi di salvataggio/invio

virtual_libraries()

class calibre.utils.formatter_functions.BuiltinVirtualLibraries

virtual_libraries() – restituisce una lista separata da virgole delle Biblioteche Virtuali che contengono questo libro. Questa funzione può essere usata solo nell'interfaccia grafica. Se vuoi usare questi valori negli schemi di salvataggio su disco o di invio al dispositivo devi creare una «Colonna costruita da altre colonne» personalizzata, usare la funzione dello schema di quella colonna, e usare il valore di quella colonna nei tuoi schemi di salvataggio/invio

If-then-else**contains(val, schema, testo con risultati, testo senza risultati)**

class calibre.utils.formatter_functions.BuiltinContains

contains(val, schema, testo con risultati, testo senza risultati) – controlla se val contiene corrispondenze con l'espressione regolare *schema*. Restituisce *testo con risultati* se vengono trovati dei risultati, altrimenti restituisce *testo senza risultati*

field_exists(nome_campo)

class calibre.utils.formatter_functions.BuiltinFieldExists

field_exists(nome_campo) – controlla se esiste un campo (colonna) chiamato nome_campo, restituisce 1 se esiste e "" se no.

ifempty(val, text if empty)

class calibre.utils.formatter_functions.BuiltinIfempty

ifempty(val, text if empty) – restituisce val se val non è vuoto, altrimenti restituisce *text if empty*

test(val, testo se non vuoto, testo se vuoto)

class calibre.utils.formatter_functions.BuiltinTest

test(val, testo se non vuoto, testo se vuoto) – restituisce “testo se non vuoto” se val non è vuoto, altrimenti restituisce “testo se vuoto”

Iterating over values

`first_non_empty(valore [, valore]*)`

class `calibre.utils.formatter_functions.BuiltinFirstNonEmpty`

`first_non_empty(valore [, valore]*)` – restituisce il primo valore non vuoto. Se tutti i valori sono vuoti, viene restituita la stringa vuota. Puoi avere un numero qualsiasi di valori.

`lookup(val, [schema, campo,]+ campo_alternativo)`

class `calibre.utils.formatter_functions.BuiltinLookup`

`lookup(val, [schema, campo,]+ campo_alternativo)` – come `switch`, a parte gli argomenti che sono nomi di campo (metadati), non testo. Il valore del campo appropriato sarà recuperato e usato. Nota che visto che le colonne composte sono campi, puoi usare questa funzione in un campo composto per utilizzare il valore di un altro campo composto. È estremamente utile per costruire percorsi di salvataggio variabili

`switch(val, [schema, valore,]+ valore_alternativo)`

class `calibre.utils.formatter_functions.BuiltinSwitch`

`switch(val, [schema, valore,]+ valore_alternativo)` – per ogni coppia *schema, valore*, verifica se *val* corrisponde all'espressione regolare *schema* e, se è così, restituisce quel *valore*. Se non c'è alcuna corrispondenza con gli schemi, viene restituito *valore_alternativo*. Puoi avere quante coppie *schema, valore* vuoi

`switch_if([test_expression, value_expression,]+ else_expression)`

class `calibre.utils.formatter_functions.BuiltinSwitchIf`

`switch_if([test_expression, value_expression,]+ else_expression)` – per ogni coppia «*test_expression, value_expression*», controlla se *test_expression* è Vera (non è vuota) e in tal caso restituisce il risultato di *value_expression*. Se nessuna *test_expression* è Vera, viene restituito il risultato di *else_expression*. Puoi avere quante coppie «*test_expression, value_expression*» desideri.

List lookup

`identifier_in_list(val, nome_id [, val_trovato, val_non_trovato])`

class `calibre.utils.formatter_functions.BuiltinIdentifierInList`

`identifier_in_list(val, nome_id [, val_trovato, val_non_trovato])` – tratta *val* come una lista di identificatori separati da virgola. Un identificatore ha la forma «*nome_id:valore*». Il parametro *nome_id* è il testo *nome_id* che si vuole cercare, può essere *nome_id* oppure *nome_id:regex*. Nel primo caso, si ha una corrispondenza se esiste un identificatore con il *nome_id* dato. Nel secondo caso, si ha una corrispondenza se *nome_id* corrisponde e se il valore dell'identificatore corrisponde all'espressione regolare *regex*. Se sono forniti *val_trovato* e *val_non_trovato*, viene restituito *val_trovato* in caso di corrispondenza e *val_non_trovato* in caso contrario. Se invece non sono forniti, viene restituita la coppia *identificatore:valore* in caso di corrispondenza e la stringa vuota in caso contrario.

in_list(val, separator, [pattern, found_val,]+ not_found_val)**class** calibre.utils.formatter_functions.**BuiltinInList**

in_list(val, separator, [pattern, found_val,]+ not_found_val) – tratta val come una lista di elementi separati da separator, se pattern corrisponde ad uno qualsiasi dei valori della lista restituisce found_val. Se pattern non corrisponde ad alcun valore della lista restituisce not_found_val. Le coppie pattern e found_value possono essere ripetute quante volte si desidera. I pattern sono controllati in ordine. Viene restituito il found_val della prima corrispondenza. Alias: in_list(), list_contains()

list_item(val, indice, separatore)**class** calibre.utils.formatter_functions.**BuiltinListitem**

list_item(val, indice, separatore) – interpreta il valore come lista di elementi separati da *separatore* e restituisce l'elemento n°indice°-simo. Il primo elemento è il numero zero. L'ultimo elemento può essere restituito usando *list_item(-1,separatore)*. Se l'elemento non è nella lista, è restituito il valore vuoto. Il separatore ha lo stesso significato che nella funzione count.

select(val, key)**class** calibre.utils.formatter_functions.**BuiltinSelect**

select(val, key) – interpreta il valore come lista separata da virgole di elementi, con elementi della forma «id:valore». Trova la coppia con l'id uguale a key, e restituisce il valore corrispondente. Restituisce la stringa vuota se nessuna corrispondenza viene trovata.

str_in_list(val, separator, [string, found_val,]+ not_found_val)**class** calibre.utils.formatter_functions.**BuiltinStrInList**

str_in_list(val, separator, [string, found_val,]+ not_found_val) – tratta val come una lista di elementi separati da separator, se la stringa corrisponde ad uno qualsiasi dei valori della lista restituisce found_val. Se la stringa non corrisponde ad alcun valore della lista restituisce not_found_val. Il confronto è per corrispondenza esatta (non «contiene») e non distingue le maiuscole. Le coppie string e found_value possono essere ripetute quante volte si desidera. I pattern sono controllati in ordine. Viene restituito il found_val della prima corrispondenza.

List manipulation**count(val, separatore)****class** calibre.utils.formatter_functions.**BuiltinCount**

count(val, separatore) – interpreta il valore come una lista di elementi separati da *separatore* e restituisce il numero di elementi della lista. La maggior parte delle liste usa una virgola come separatore, ma gli autori usano la e commerciale. Esempi: {tags:count(,)}, {authors:count(&)}. Alias: count(), list_count()

list_count_matching(list, pattern, separator)

class calibre.utils.formatter_functions.**BuiltinListCountMatching**

list_count_matching(list, pattern, separator) – interpreta “list” come una lista di elementi separati da “separator” e restituisce il numero di elementi nella lista che corrispondono all’espressione regolare “pattern”. Alias: list_count_matching(), count_matching()

list_difference(lista1, lista2, separatore)

class calibre.utils.formatter_functions.**BuiltinListDifference**

list_difference(lista1, lista2, separatore) – restituisce una lista creata rimuovendo da lista1 tutti gli elementi trovati nella lista2, facendo un confronto senza distinzione di maiuscole. Gli elementi in lista1 e in lista2 sono separati da separatore, come anche gli elementi nella lista restituita.

list_equals(lista1, sep1, lista2, sep2, val_si, val_no)

class calibre.utils.formatter_functions.**BuiltinListEquals**

list_equals(lista1, sep1, lista2, sep2, val_si, val_no) – restituisce val_si se lista1 e lista2 contengono gli stessi elementi, altrimenti restituisce val_no. Gli elementi sono determinati dividendo ogni lista usando il carattere separatore appropriato (sep1 o sep2). L’ordine degli elementi nelle liste non è rilevante. Il confronto non tiene conto delle maiuscole.

list_intersection(lista1, lista2, separatore)

class calibre.utils.formatter_functions.**BuiltinListIntersection**

list_intersection(lista1, lista2, separatore) – restituisce una lista creata rimuovendo da lista1 tutti gli elementi non trovati nella lista2, facendo un confronto senza distinzione di maiuscole. Gli elementi in lista1 e in lista2 sono separati da separatore, come anche gli elementi nella lista restituita.

list_join(with_separator, list1, separator1 [, list2, separator2]*)

class calibre.utils.formatter_functions.**BuiltinListJoin**

list_join(with_separator, list1, separator1 [, list2, separator2]*) – restituisce una lista creata unendo gli elementi nelle liste di origine (list1, ecc.), usando with_separator come separatore della lista risultante. Gli elementi in ognuna delle liste[123...] di origine sono divisi dal separatore[123...] associato. Una lista può contenere zero valori. Può anche essere un campo con un solo valore, come l’editore, che è di fatto una lista con un solo elemento. Eventuali duplicati sono rimossi con un confronto senza distinzione di maiuscole. Gli elementi sono restituiti nell’ordine in cui appaiono nelle liste di origine. Se elementi delle liste differiscono solo per le maiuscole, viene usato l’ultimo di essi. Tutti i separatori possono essere di più caratteri. Esempio:

program:

```
list_join("#@#", $authors, "&", $tags, ",")
```

Puoi usare list_join sui risultati di chiamate precedenti a list_join stessa in questo modo:

program:

```
a = list_join("#@#", $authors, "&", $tags, ","); b = list_join("#@#", a, "#@#", $#genre, ",", $#people, "&")
```

Puoi usare delle espressioni per generare un elenco. Per esempio, immagina di volere delle voci con autori e #genere, ma con quest'ultimo cambiato in modo da mostrare la parola "Genere: " seguita dalla prima lettera del genere; ad esempio il genere "Romanzi" diventa "Genere: R". Puoi farlo in questo modo:

program:

```
list_join("#@#", $authors, "&", list_re($#genre, ",", "^(.)*$", "Genre: 1"), ",",")
```

list_re(lista_sorg, separatore, includi_re, sost_opz)

class calibre.utils.formatter_functions.**BuiltinListRe**

list_re(lista_sorg, separatore, includi_re, sost_opz) – Costruisce una lista, separando come prima cosa lista_sorg in elementi singoli usando il carattere separatore. Per ogni elemento nella lista, controlla se ha corrispondenze con includi_re. Se ne ha, viene aggiunto alla lista da restituire. Se sost_opz non è la stringa vuota, applica la sostituzione prima di aggiungere l'elemento alla lista restituita.

list_re_group(src_list, separator, include_re, search_re [, group_template] +)

class calibre.utils.formatter_functions.**BuiltinListReGroup**

list_re_group(src_list, separator, include_re, search_re [, group_template] +) – Come list_re se non che le sostituzioni non sono opzionali. Usa re_group(list_item, search_re, group_template, ...) per fare le sostituzioni nella lista risultante.

list_remove_duplicates(lista, separatore)

class calibre.utils.formatter_functions.**BuiltinListRemoveDuplicates**

list_remove_duplicates(lista, separatore) – restituisce una lista creata rimuovendo gli elementi duplicati della lista sorgente. Se gli elementi sono diversi solo per le maiuscole, è restituito l'ultimo di essi. Gli elementi nella lista sorgente sono separati da separatore, come anche gli elementi nella lista restituita.

list_sort(lista, direzione, separatore)

class calibre.utils.formatter_functions.**BuiltinListSort**

list_sort(lista, direzione, separatore) – restituisce la lista ordinata usando un metodo che non tiene conto delle maiuscole. Se direzione è zero, la lista è in ordine ascendente, altrimenti discendente. Gli elementi della lista sono separati da separatore, come anche gli elementi della lista restituita.

list_split(elenca_val, sep, prefisso_id)

class calibre.utils.formatter_functions.**BuiltinListSplit**

list_split(elenca_val, sep, prefisso_id) – divide elenca_val in valori separati usando "sep", poi assegna i valori alle variabili chiamate "prefisso_id_N" dove N è la posizione del valore nella lista. Il primo elemento ha posizione 0 (zero). La funzione restituisce l'ultimo elemento della lista. Esempio: split("uno:due:ciao", ":", "var") è equivalente a var_0 = "uno"; var_1 = "due"; var_2 = "ciao".

list_union(lista1, lista2, separatore)

class calibre.utils.formatter_functions.**BuiltinListUnion**

list_union(lista1, lista2, separatore) – restituisce una lista creata unendo gli elementi nella lista 1 e nella lista 2, e rimuove gli elementi duplicati facendo un confronto senza distinzione di maiuscole. Se degli elementi sono diversi come maiuscole, viene usato quello della lista1. Gli elementi in lista1 e in lista2 sono separati da separatore, come anche gli elementi nella lista restituita. Alias: list_union(), merge_lists()

range(start, stop, passo, limite)

class calibre.utils.formatter_functions.**BuiltinRange**

range(start, stop, passo, limite) – restituisce un elenco di numeri generati scorrendo nell'intervallo specificato dai parametri start, stop, e passo, con una lunghezza massima data da limite. Il primo valore prodotto è "start". I valori successivi next_v sono dati da current_v+passo. Lo scorrimento continua finché next_v > stop. Viene prodotto un elenco vuoto se start non passa il test: start>=stop se passo è positivo. Il limite imposta la lunghezza massima dell'elenco e ha un valore predefinito di 1000. I parametri start, passo e limite sono opzionali. Se si chiama range() con un solo argomento, questo sarà stop. Con due argomenti si avranno start e stop. Con tre argomenti, start, stop e passo. Con quattro, start, stop, passo e limite. Esempi: range(5) -> "0,1,2,3,4". range(0,5) -> "0,1,2,3,4". range(-1,5) -> "-1,0,1,2,3,4". range(1,5) -> "1,2,3,4". range(1,5,2) -> "1,3". range(1,5,2,5) -> "1,3". range(1,5,2,1) -> error(limit exceeded).

subitems(val, start_index, end_index)

class calibre.utils.formatter_functions.**BuiltinSubitems**

subitems(val, start_index, end_index) – Questa funzione è usata per spezzare liste di elementi quali i generi. Interpreta value come una lista separata da virgole di elementi, dove ogni elemento è una lista separata da punti. Restituisce una nuova lista costruita prima trovando tutti gli elementi separati da punti e poi, per ognuno di questi elementi, estraendo i componenti che vanno da start_index a end_index e ricombinando i risultati. Il primo componente di una lista separata da punti ha indice zero. Se un indice è negativo, allora conta dalla fine della lista. Come caso speciale, un end_index uguale a zero è preso come lunghezza della lista. Esempio che usa il linguaggio dei modelli di base e che assume per #genre il valore «A.B.C»: {#genre:subitems(0,1)} restituisce «A». {#genre:subitems(0,2)} restituisce «A.B». {#genre:subitems(1,0)} restituisce «B.C». Assumendo per #genre il valore «A.B.C, D.E.F», {#genre:subitems(0,1)} restituisce «A, D». {#genre:subitems(0,2)} restituisce «A.B, D.E»

sublist(val, start_index, end_index, separator)

class calibre.utils.formatter_functions.**BuiltinSublist**

sublist(val, start_index, end_index, separator) – interpreta il valore come lista di elementi separati da separator, e restituisce una nuova lista che va dall'elemento start_index a quello end_index. Il primo elemento è il numero zero. Se un indice è negativo, allora conta dalla fine della lista. Come caso speciale, un end_index uguale a zero è preso come lunghezza della lista. Alcuni esempi che usano il linguaggio dei modelli di base e che presuppongono che la colonna dei tag (che è separata da virgole) contenga «A, B, C»: {tags:sublist(0,1,\,)} restituisce «A». {tags:sublist(-1,0,\,)} restituisce «C». {tags:sublist(0,-1,\,)} restituisce «A, B».

Other

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

class `calibre.utils.formatter_functions.BuiltinArguments`

`arguments(id[=expression] [, id[=expression]]*)` – Viene usata in un modello salvato per recuperare gli argomenti passati nella chiamata. Dichiarare e inizializza le variabili locali, che sono di fatto parametri. Le variabili sono posizionali; prendono il valore del parametro fornito nella chiamata alla stessa posizione. Se il parametro corrispondente non è fornito nella chiamata, `arguments` assegna alla variabile il valore predefinito fornito. Se nessun valore predefinito è presente, la variabile è impostata a stringa vuota.

`assign(id, val)`

class `calibre.utils.formatter_functions.BuiltinAssign`

`assign(id, val)` – assegna `val` a `id`, quindi restituisce `val`. `id` deve essere un identificatore, non un'espressione. Questa funzione può spesso essere sostituita con l'operatore `=`.

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

class `calibre.utils.formatter_functions.BuiltinGlobals`

`globals(id[=expression] [, id[=expression]]*)` – Recupera le «variabili globali» che possono essere passate alla funzione di formattazione. Dichiarare e inizializza le variabili locali con i nomi delle variabili globali passate. Se la variabile corrispondente non è presente in quelle globali passate, assegna a quella variabile il valore predefinito fornito. Se non è presente alcun valore predefinito la variabile è impostata a stringa vuota.

`print(a[, b]*)`

class `calibre.utils.formatter_functions.BuiltinPrint`

`print(a[, b]*)` – stampa gli argomenti sullo standard output. A meno che tu non avvii calibre dalla riga di comando (`calibre-debug -g`), l'output finirà in un buco nero.

Recursion

`eval(modello)`

class `calibre.utils.formatter_functions.BuiltinEval`

`eval(modello)` – valuta il modello, passando le variabili locali (quelle assegnate con “`assign`”) al posto dei metadati del libro. Questo permette di usare l'elaborazione dei modelli per costruire risultati complessi dalle variabili locali. Dato che i caratteri `{` e `}` sono speciali, bisogna usare `[[` per il carattere `{` e `]]` per il carattere `}`; vengono convertiti automaticamente. Tieni anche presente che non possono essere usati prefissi e suffissi (la sintassi `|prefisso|suffisso`) nell'argomento di questa funzione quando è utilizzata in modalità programma di modello.

template(x)

class calibre.utils.formatter_functions.BuiltinTemplate

template(x) – valuta x come modello. La valutazione è fatta solo nel suo contesto, che significa che le variabili non sono condivise tra chiamante e valutazione del modello. Dato che i caratteri { e } sono speciali, bisogna usare [[per il carattere { e]] per il carattere }; vengono convertiti automaticamente. Per esempio, template("[[title_sort]]") valuterà il modello {title_sort} e restituirà il suo valore. Tieni anche presente che non possono essere usati prefissi e suffissi (la sintassi *|prefisso|suffisso*) nell'argomento di questa funzione quando è utilizzata in modalità programma di modello.

Relational

cmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.BuiltinCmp

cmp(x, y, lt, eq, gt) – fa un confronto tra x e y dopo averli convertiti entrambi in numeri. Restituisce lt se $x < y$. Restituisce eq se $x == y$. Altrimenti restituisce gt. In molti casi gli operatori di confronto numerico ($>\#$, $<\#$, $==\#$ ecc.) possono sostituire questa funzione.

first_matching_cmp(val, [cmp1, result1,]+, else_result)

class calibre.utils.formatter_functions.BuiltinFirstMatchingCmp

first_matching_cmp(val, [cmp1, result1,]+, else_result) – confronta «val < cmpN» in sequenza, restituendo resultN per il primo confronto che ha successo. Esempio: first_matching_cmp(10,5,»small»,10,»middle»,15,»large»,»giant») restituisce «large». Lo stesso esempio con 16 come primo valore restituisce «giant».

strcmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.BuiltinStrcmp

strcmp(x, y, lt, eq, gt) – confronta x e y come stringhe, senza distinguere tra maiuscole e minuscole. Restituisce lt se $x < y$. Restituisce eq se $x == y$. Altrimenti restituisce gt. In molti casi possono essere usati direttamente gli operatori lessicali di confronto ($>$, $<$, $==$ ecc.) al posto di questa funzione.

strcmpcase(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.BuiltinStrcmpcase

strcmpcase(x, y, lt, eq, gt) – confronta x e y come stringhe, distinguendo tra maiuscole e minuscole. Restituisce lt se $x < y$. Restituisce eq se $x == y$. Altrimenti restituisce gt. Nota bene: Questo non è il comportamento predefinito applicato da calibre, per esempio, negli operatori lessicali di confronto ($==$, $>$, $<$, ecc.). Questa funzione potrebbe portare a risultati inaspettati, sarebbe meglio usare strcmp() dove possibile.

String case changes

capitalize(val)

class calibre.utils.formatter_functions.BuiltinCapitalize
capitalize(val) – restituisce il valore del campo con l'iniziale maiuscola

lowercase(val)

class calibre.utils.formatter_functions.BuiltinLowercase
lowercase(val) – restituisce in minuscolo il valore del campo

titlecase(val)

class calibre.utils.formatter_functions.BuiltinTitlecase
titlecase(val) – restituisce iniziali parole di val in maiuscolo

uppercase(val)

class calibre.utils.formatter_functions.BuiltinUppercase
uppercase(val) – restituisce in maiuscolo il valore del campo

String manipulation

character(nome_carattere)

class calibre.utils.formatter_functions.BuiltinCharacter
character(nome_carattere) – restituisce il carattere con nome nome_carattere. Per esempio, character("newline") restituisce un carattere nuova riga ("n"). I nomi di carattere supportati sono "newline", "return", "tab", e "backslash".

re(val, pattern, replacement)

class calibre.utils.formatter_functions.BuiltinRe
re(val, pattern, replacement) – restituisce val dopo aver applicato l'espressione regolare. Tutti le occorrenze di *pattern* sono sostituite con *replacement*. Come in tutto calibre, queste sono espressioni regolari compatibili con Python

re_group(val, schema [, modello_del_gruppo]*)**class** calibre.utils.formatter_functions.**BuiltinReGroup**

re_group(val, schema [, modello_del_gruppo]*) – restituisce una stringa creata applicando lo schema dell'espressione regolare a val e sostituendo ognuna delle corrispondenze con la stringa costruita sostituendo ogni gruppo trovato con il valore restituito dal modello corrispondente. Il valore inizialmente trovato del gruppo è disponibile come \$. In modalità programma di modello, come per le funzioni template ed eval, si usano [[per { e]] per }. Il seguente esempio in modalità programma di modello cerca le serie con più di una parola e rende maiuscola la prima parola: {series:"re_group(\$, «(S*)(.*)*», «[[:uppercase:]]», «[[:\$]]»)"}

shorten(val, car sin, testo cent, car dest)**class** calibre.utils.formatter_functions.**BuiltinShorten**

shorten(val, car sin, testo cent, car dest) – Restituisce una versione accorciata di val, che consiste dei caratteri *car sin* dell'inizio di val, seguiti da *testo cent*, seguiti dai caratteri *car dest* della fine della stringa. *car sin* e *car dest* devono essere interi. Per esempio, immagina che il titolo del libro sia *Antiche Leggi Inglesi ai Tempi di Ivanhoe*, e di voler fare in modo che il titolo occupi al più 15 caratteri. Se usi {title:shorten(9,-,5)}, il risultato sarà *Antiche L-anhoe*. Se la lunghezza del campo è minore di *car sin* + *car dest* + la lunghezza di *testo cent*, il campo sarà lasciato intatto. Per esempio, il titolo *La Cupola* non verrebbe cambiato.

strcat(a [, b]*)**class** calibre.utils.formatter_functions.**BuiltinStrcat**

strcat(a [, b]*) – può accettare un qualsiasi numero di argomenti. Restituisce la stringa formata concatenando tutti gli argomenti

strcat_max(max, stringa1 [, prefisso2, stringa2]*)**class** calibre.utils.formatter_functions.**BuiltinStrcatMax**

strcat_max(max, stringa1 [, prefisso2, stringa2]*) – Restituisce una stringa formata concatenando gli argomenti. Il valore restituito è inizializzato come stringa1. Le coppie *prefisso*, *stringa* sono aggiunte alla fine del valore purché la lunghezza della stringa risultante sia minore di *max*. Stringa 1 viene restituita anche se risulta più lunga di *max*. Puoi fornire quante coppie *prefisso*, *stringa* desideri.

strlen(a)**class** calibre.utils.formatter_functions.**BuiltinStrlen**

strlen(a) – restituisce la lunghezza della stringa fornita come argomento

substr(str, inizio, fine)

class calibre.utils.formatter_functions.**BuiltinSubstr**

substr(str, inizio, fine) – restituisce i caratteri di str da «inizio» a «fine». Il primo carattere in str è il carattere zero. Se fine è un numero negativo, indica quel numero di caratteri contando da destra. Se fine è zero, indica l'ultimo carattere. Ad esempio, substr("12345", 1, 0) restituisce "2345" e substr("12345", 1, -1) restituisce "234".

swap_around_articles(val, separatore)

class calibre.utils.formatter_functions.**BuiltinSwapAroundArticles**

swap_around_articles(val, separatore) – restituisce val con gli articoli spostati alla fine. Il valore può essere un elenco, nel qual caso ognuno dei suoi membri è elaborato. Se il valore è un elenco, devi fornire il separatore di valori. Se nessun separatore è fornito, il valore è trattato come singolo e non come elenco.

swap_around_comma(val)

class calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

swap_around_comma(val) – dato un valore in formato «B, A», restituisce «A B». Utile soprattutto per convertire nomi dal formato CG, NM a NM CG. Se non c'è virgola, la funzione restituisce val senza cambiamenti

to_hex(val)

class calibre.utils.formatter_functions.**BuiltinToHex**

to_hex(val) – restituisce la stringa codificata in esadecimale. Utile per costruire URL di calibre.

transliterate(a)

class calibre.utils.formatter_functions.**BuiltinTransliterate**

transliterate(a) – Restituisce una stringa in alfabeto latino creata in modo da approssimare il suono delle parole nella stringa sorgente. Per esempio, se la sorgente è «Фёдор Михайлович Достоевский» la funzione restituisce «Fiodor Mikhailovich Dostoevskii».

Template database functions**book_count(query, use_vl)**

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

book_count(query, use_vl) – restituisce il numero di libri trovato cercando query. Se use_vl è 0 (zero) le biblioteche virtuali vengono ignorate. Questa funzione può essere usata solo nell'interfaccia utente.

book_values(column, query, sep, use_vl)

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

book_values(column, query, sep, use_vl) – restituisce un elenco dei valori contenuti nella colonna «column», separati da «sep», tra i libri trovati cercando «query». Se use_vl è 0 (zero) le biblioteche virtuali vengono ignorate. Questa funzione può essere usata solo nell'interfaccia utente.

extra_file_modtime(nome_file, formato_stringa)

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

extra_file_modtime(nome_file, formato_stringa) – restituisce la data di modifica del file aggiuntivo “nome_file” nella cartella “data/” del libro se esiste, altrimenti -1.0. La data di modifica è formattata secondo “formato_stringa” (vedi format_date()). Se “formato_stringa” è vuoto, restituisce la data di modifica come numero di secondi in virgola mobile dall'inizio dell'epoca (epoch). L'epoca dipende dal sistema operativo. Questa funzione può essere usata solo nell'interfaccia grafica.

extra_file_names(sep [, schema])

class calibre.utils.formatter_functions.**BuiltinExtraFileNames**

extra_file_names(sep [, schema]) – restituisce una lista separata da sep di file aggiuntivi nella cartella “data/” del libro. Se il parametro opzionale “schema”, un'espressione regolare, è fornito la lista viene limitata ai file che corrispondono allo schema. La corrispondenza non tiene conto delle maiuscole. Questa funzione può essere usata solamente nell'interfaccia utente.

extra_file_size(nome_file)

class calibre.utils.formatter_functions.**BuiltinExtraFileSize**

extra_file_size(nome_file) – restituisce la dimensione in byte del file aggiuntivo “nome_file” nella cartella “data/” del libro se esiste, altrimenti -1. Questa funzione può essere usata solamente nell'interfaccia utente.

has_extra_files([schema])

class calibre.utils.formatter_functions.**BuiltinHasExtraFiles**

has_extra_files([schema]) – restituisce il numero di file aggiuntivi, altrimenti “” (la stringa vuota). Se il parametro opzionale “schema” (un'espressione regolare) è fornito, la lista viene limitata ai file che corrispondono allo schema prima che i file vengano contati. La corrispondenza non tiene conto delle maiuscole. Questa funzione può essere usata solamente nell'interfaccia utente.

other

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

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

set_globals(id[=expression] [, id[=expression]]*) – Imposta delle «variabili globali» che possono essere passate al formatter. Queste variabili prendono il nome dell'id fornito. Viene usato il valore dell'id, a meno che non sia fornita un'espressione.

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=('Sconosciuto'), 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()` (pagina 201) 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` (pagina 202).

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',
))
```

(continues on next page)

(continua dalla pagina precedente)

```

'''
The list of names that convert to identifiers when in get and set.
'''

TOP_LEVEL_IDENTIFIERS = frozenset((
    'isbn',
))

PUBLICATION_METADATA_FIELDS = frozenset((
    'title',                # title must never be None. Should be _('Unknown')
    # Pseudo field that can be set, but if not set is auto generated
    # from title and languages
    'title_sort',
    'authors',              # Ordered list. Must never be None, can be [_('Unknown')]
    'author_sort_map',      # Map of sort strings for each author
    # Pseudo field that can be set, but if not set is auto generated
    # from authors and languages
    'author_sort',
    'book_producer',
    'timestamp',            # Dates and times must be timezone aware
    'pubdate',
    'last_modified',
    'rights',
    # So far only known publication type is periodical:calibre
    # If None, means book
    'publication_type',
    'uuid',                 # A UUID usually of type 4
    'languages',            # ordered list of languages in this publication
    'publisher',            # Simple string, no special semantics
    # Absolute path to image file encoded in filesystem_encoding
    'cover',
    # Of the form (format, data) where format is, e.g. 'jpeg', 'png', 'gif'...
    'cover_data',
    # Either thumbnail data, or an object with the attribute
    # image_path which is the path to an image file, encoded
    # in filesystem_encoding
    'thumbnail',
))

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
))

```

(continues on next page)

(continua dalla pagina precedente)

```

CALIBRE_METADATA_FIELDS = frozenset((
    'application_id',    # An application id, currently set to the db_id.
    'db_id',             # the calibre primary key of the item.
    'formats',          # list of formats (extensions) for this book
    # a dict of user category names, where the value is a list of item names
    # from the book that are in that category
    'user_categories',
    # a dict of items to associated hyperlink
    'link_maps',
))

ALL_METADATA_FIELDS =
    SOCIAL_METADATA_FIELDS.union(
        PUBLICATION_METADATA_FIELDS).union(
        BOOK_STRUCTURE_FIELDS).union(
        USER_METADATA_FIELDS).union(
        DEVICE_METADATA_FIELDS).union(
        CALIBRE_METADATA_FIELDS)

# All fields except custom fields
STANDARD_METADATA_FIELDS = SOCIAL_METADATA_FIELDS.union(
    PUBLICATION_METADATA_FIELDS).union(
    BOOK_STRUCTURE_FIELDS).union(
    DEVICE_METADATA_FIELDS).union(
    CALIBRE_METADATA_FIELDS)

# Metadata fields that smart update must do special processing to copy.
SC_FIELDS_NOT_COPIED = frozenset(('title', 'title_sort', 'authors',
    'author_sort', 'author_sort_map',
    'cover_data', 'tags', 'languages',
    'identifiers'))

# Metadata fields that smart update should copy only if the source is not None
SC_FIELDS_COPY_NOT_NULL = frozenset(('device_collections', 'lpath', 'size', 'comments
    ↪', 'thumbnail'))

# Metadata fields that smart update should copy without special handling
SC_COPYABLE_FIELDS =
    SOCIAL_METADATA_FIELDS.union(
        PUBLICATION_METADATA_FIELDS).union(
        BOOK_STRUCTURE_FIELDS).union(
        DEVICE_METADATA_FIELDS).union(
        CALIBRE_METADATA_FIELDS) - \
    SC_FIELDS_NOT_COPIED.union(
    SC_FIELDS_COPY_NOT_NULL)

SERIALIZABLE_FIELDS =
    SOCIAL_METADATA_FIELDS.union(
        USER_METADATA_FIELDS).union(
        PUBLICATION_METADATA_FIELDS).union(
        CALIBRE_METADATA_FIELDS).union(
        DEVICE_METADATA_FIELDS) - \
    frozenset(('device_collections', 'formats',
        'cover_data'))

# these are rebuilt when needed

```


10.4 Tutto sull'utilizzo delle espressioni regolari in calibre

Le espressioni regolari sono usate in molte parti di calibre per eseguire manipolazioni avanzate del contenuto e dei metadati degli e-book. Questa guida è un'introduzione amichevole all'uso delle espressioni regolari in calibre.

Contenuti

- *Per prima cosa, una parola di avviso e una di incoraggiamento* (pagina 205)
- *Dove è possibile utilizzare le espressioni regolari all'interno di calibre?* (pagina 206)
- *Cosa diamine è un'espressione regolare?* (pagina 206)
- *Potresti spiegare meglio?* (pagina 206)
- *Non suona affatto male. E poi?* (pagina 206)
- *Ehi, ben fatto! Ora inizia ad avere un senso!* (pagina 207)
- *Bene, questi caratteri speciali sono tutti molto graziosi, ma se volessi trovare un punto, o un punto di domanda?* (pagina 207)
- *Quindi, quali sono i set più utili?* (pagina 208)
- *Ma se avessi alcune stringhe diverse tra loro da trovare, diventerebbe complicato?* (pagina 208)
- *Ti sei dimenticato...* (pagina 209)
- *All'inizio hai detto che c'era un modo di creare un'espressione regolare che non distingue le maiuscole?* (pagina 209)
- *Penso di stare iniziando a capire queste espressioni regolari... Come le posso usare in calibre?* (pagina 209)
 - *Conversioni* (pagina 209)
 - *Aggiunta di libri* (pagina 210)
 - *Modifica di gruppo dei metadati* (pagina 210)
- *Riferimenti rapidi* (pagina 211)
- *Riconoscimenti* (pagina 216)

10.4.1 Per prima cosa, una parola di avviso e una di incoraggiamento

Questa guida sarà, per forza di cose, abbastanza tecnica - dopotutto, le espressioni regolari sono uno strumento tecnico che fa cose tecniche. Dovrò usare dei tecnicismi e dei concetti che potrebbero sembrare complicati o contorti. Farò del mio meglio per spiegare quei concetti il più chiaramente possibile, ma non posso proprio fare a meno di utilizzarli. Detto ciò, non essere scoraggiato da questo gergo, perché ho provato a spiegare tutto ciò che appare per la prima volta. E sebbene le espressioni regolari stesse possano sembrare una magia oscura e misteriosa (o, per essere più prosaici, segni e lettere astrusi messi a caso in una stringa senza senso), ti assicuro che non sono poi così complicati. Anche coloro che capiscono molto bene le espressioni regolari hanno problemi a leggere le più complesse tra di esse, ma scriverle non è così complicato - l'espressione è costruita passo passo. E allora, fai un passo avanti e seguimi nella tana del Bianconiglio

10.4.2 Dove è possibile utilizzare le espressioni regolari all'interno di calibre?

Ci sono vari luoghi in cui calibre utilizza le espressioni regolari. C'è *Cerca e sostituisci* nelle opzioni di conversione, l'estrazione dei metadati dai nomi dei file nelle impostazioni di importazione e il *Cerca e sostituisci* nella modifica di gruppo dei metadati dei libri. Anche l'editor di libri di calibre può usare le espressioni regolari nella sua funzione *Cerca e sostituisci*. Infine puoi usare le espressioni regolari quando cerchi nell'elenco dei libri di calibre e nel Lettore E-book di calibre.

10.4.3 Cosa diamine è un'espressione regolare?

Un'espressione regolare è un modo per descrivere insiemi di stringhe. Una singola espressione regolare può *associare* una serie di stringhe differenti. È questo che rende le espressioni regolari così potenti – sono un modo conciso per descrivere un numero potenzialmente alto di variazioni.

Nota: Qui uso stringa con il significato usato nei linguaggi di programmazione: una stringa di uno o più caratteri, che includono caratteri effettivi, numeri, punteggiatura e i cosiddetti «whitespace» (testo a capo, tabulazioni, ecc.). Ricorda che generalmente caratteri con maiuscole differenti non sono considerati uguali, ne deriva perciò che «a» è un carattere di verso da «A», e così via. In calibre, le espressioni regolari non fanno distinzione di maiuscole nella Barra di Ricerca, ma la fanno nelle opzioni di conversione. Esiste un modo per fare in modo che nessuna espressione regolare distingua le maiuscole, ma ne parleremo più avanti. Diventa complicato perché le espressioni regolari permettono varianti nelle stringhe che trovano, e quindi a un'espressione può corrispondere più di una stringa, che è poi la ragione per cui la gente si prende il disturbo di usarle. Più informazioni fra un attimo.

10.4.4 Potresti spiegare meglio?

Siamo qui per questo. Primo, e questo è il concetto più importante delle espressioni regolari: *Una stringa presa da sola è un'espressione regolare che trova se stessa*. Ovverosia, se volessi trovare la stringa "Ciao, Mondo!" usando un'espressione regolare, l'espressione regolare da usare sarebbe *Ciao, Mondo!*. E sì, è proprio così semplice. Ti sarai accorto, tuttavia, che questa espressione trova *solo* la stringa esatta "Ciao, Mondo!", non p.es. "Ciao, mOndo!" o "ciao, mondo!" o altre varianti del genere.

10.4.5 Non suona affatto male. E poi?

Poi inizia la parte davvero interessante. Ricordi quando ti ho detto che le espressioni regolari possono trovare più di una stringa? È qui che le cose si complicano. Poniamo, come esercizio un po' più concreto, che l'e-book che volevi convertire avesse un fastidioso piè di pagina con la conta delle pagine, come «Pagina 5 di 423». Ovviamente il numero di pagina salirebbe da 1 a 423, per cui dovresti trovare 423 stringhe differenti, giusto? Sbagliato in realtà: le espressioni regolari ti permettono di definire set di caratteri da trovare: per definirne uno, metti tutti i caratteri che vuoi avere nel set dentro parentesi quadre. Quindi, ad esempio, il set [abc] troverebbe sia il carattere «a», sia «b» sia «c». *I set troveranno sempre solamente uno dei caratteri del set*. Essi «riconoscono» gli intervalli, ovvero, se volessi trovare tutti i caratteri minuscoli dovresti usare il set [a-z], per tutti i caratteri minuscoli e maiuscoli dovresti usare [a-zA-Z], e così via. Chiaro il concetto? Quindi, ovviamente, usando l'espressione *Pagina [0-9] di 423* saresti in grado di trovare le prime 9 pagine, riducendo quindi il numero di espressioni necessarie a tre: la seconda espressione *Pagina [0-9][0-9] di 423* troverebbe tutti i numeri di pagina di due cifre, e sono sicuro che tu possa immaginare l'aspetto della terza espressione. Esatto, vai pure. Scrivila.

10.4.6 Ehi, ben fatto! Ora inizia ad avere un senso!

Speravo proprio che lo dicessi. Ma aspetta, ora diventa ancora più bello! Abbiamo appena visto che usando gli insiemi possiamo trovare un carattere tra molti inclusi. Ma puoi anche ripetere un carattere o un insieme, e ridurre così il numero di espressioni necessarie per gestire l'esempio dei numeri di pagina di poco fa a uno. Esatto, UNO! Eccitato? Dovresti esserlo! Funziona così: Alcuni cosiddetti caratteri speciali, «+», «?» e «*», *ripetono l'elemento singolo che li precede*. Per elemento si intende un singolo carattere, un insieme di caratteri, una sequenza di escape o un gruppo (Di questi ultimi due parleremo più avanti) – in breve ogni entità singola in un'espressione regolare. Questi caratteri sono chiamati wildcard o quantificatori. Per essere più precisi, «?» trova *0 o 1* dell'elemento che lo precede, «*» trova *0 o più* dell'elemento che lo precede e «+» trova *1 o più* dell'elemento che lo precede. Alcuni esempi: l'espressione `a?` troverebbe sia «» (che è la stringa vuota, non proprio utile nel nostro caso) sia «a», l'espressione `a*` troverebbe «», «a», «aa» o un qualsiasi numero di a messe in fila e, infine, l'espressione `a+` troverebbe «a», «aa» o un qualsiasi numero di a messe in fila (Nota: non troverebbe la stringa vuota!). Lo stesso vale per gli insiemi: l'espressione `[0-9]+` troverebbe *ogni numero intero che c'è!* So cosa stai pensando, e hai ragione: se usassi questa espressione nel caso di prima dei numeri di pagina, non sarebbe questa l'espressione unica che trova tutti i numeri di pagina? Sì, l'espressione `Pagina [0-9]+ di 423` troverebbe tutti i numeri di pagina in quel libro!

Nota: Un appunto su questi quantificatori: di solito cercano di trovare la maggior quantità di testo possibile, quindi fai attenzione quando li usi. Questo è detto «comportamento vorace» - sono sicuro che capisci perché. Questo comportamento diventa problematico quando, poniamo, provi a trovare un tag. Considera per esempio la stringa `<p class="calibre2">Titolo qui</p>` e poniamo che tu voglia trovare il tag di apertura (la parte inclusa nella prima coppia di parentesi angolari, maggiori informazioni sui tag più avanti). Ti verrebbe da pensare che l'espressione `<p.*>` trovi quel tag, ma in realtà trova l'intera stringa! (Il carattere «.» è un altro carattere speciale. Trova ogni carattere tranne le interruzioni di riga quindi, in pratica, l'espressione `.*` troverebbe ogni singola linea immaginabile). Invece, prova a usare `<p.*?>` che rende il quantificatore «*» non vorace. Quell'espressione troverebbe solo il primo tag di apertura, come si voleva. C'è anche un altro modo per ottenere la stessa cosa, in realtà: l'espressione `<p[^>]*>` troverà lo stesso tag di apertura - capirai perché dopo la prossima sezione. Ricorda solo che c'è spesso più di un modo di scrivere un'espressione regolare.

10.4.7 Bene, questi caratteri speciali sono tutti molto graziosi, ma se volessi trovare un punto, o un punto di domanda?

Puoi farlo, naturalmente: Basta mettere una barra rovesciata prima di ogni carattere speciale e sarà interpretato come il carattere digitato, senza alcun significato speciale. Questa coppia di barra rovesciata seguita da un carattere singolo è chiamata sequenza di escape, e l'atto di mettere una barra rovesciata prima di un carattere speciale è detto (in inglese) «escaping the character». Una sequenza di escape è interpretata come un singolo elemento. Naturalmente ci sono sequenze di escape che fanno di più che rappresentare semplicemente i caratteri speciali, per esempio `"\t"` indica una tabulazione. Parleremo di alcune di queste sequenze di escape più avanti. Ah e comunque, riguardo a quei caratteri speciali: Considera che ognuno dei caratteri di cui discutiamo in questa introduzione perché hanno una certa funzione sia speciale, e che quindi abbia bisogno della barra di escape se vuoi il carattere digitato.

10.4.8 Quindi, quali sono i set più utili?

Sapevo che l'avresti chiesto. Alcuni utili set sono `[0-9]` che trova una singola cifra, `[a-z]` che trova una singola lettera minuscola, `[A-Z]` che trova una singola lettera maiuscola, `[a-zA-Z]` che trova una singola lettera e `[a-zA-Z0-9]` che trova una singola lettera o numero. Puoi anche usare una sequenza di escape per fare più in fretta:

`\d`
è equivalente a `[0-9]`

`\w`
è equivalente a `[a-zA-Z0-9_]`

`\s`
è equivalente a un qualunque spazio

Nota: Con «spazio» si intende qualsiasi cosa non venga stampata. Questi caratteri includono il normale spazio, il rientro (tabulazione), testo a capo, interruzione di pagina, ritorno a inizio riga, spazio unificatore, ecc.

Nota: I set di maiuscole e minuscole possono corrispondere sia alle maiuscole che alle minuscole se è abilitata l'impostazione per rendere le ricerche senza distinzione tra maiuscole e minuscole. Tali impostazioni si trovano, ad esempio, in Preferenze->Ricerca in calibre stesso e nel pannello di ricerca in calibre *Visualizzatore di e-book* così come nello strumento calibre *Modifica libro*.

Per concludere con i set, puoi anche definire un set con tutti i caratteri *esclusi* quelli nel set. Lo puoi fare includendo il carattere `^` come *primo carattere del set*. Così, `[^a]` troverebbe ogni carattere eccetto «a». Questo è detto integrare il set. Anche le sequenze di escape abbreviate che abbiamo visto in precedenza possono essere integrate: `^\d` significa qualunque carattere non numerico, essendo perciò equivalente a `[^0-9]`. Le altre sequenze abbreviate possono essere integrate usando, hai indovinato, le rispettive lettere maiuscole invece di quelle minuscole. Quindi, tornando all'esempio `<p [^>] *` della sezione precedente, ora puoi vedere che il set di caratteri utilizzato prova a trovare un qualsiasi carattere eccetto una parentesi angolare di chiusura.

10.4.9 Ma se avessi alcune stringhe diverse tra loro da trovare, diventerebbe complicato?

Niente paura, non c'è niente di più semplice. Considera questo esempio: Il libro che stai convertendo ha scritto «Titolo» su tutte le pagine dispari e «Autore» su tutte le pagine pari. Molto carino su carta, vero? Ma in un e-book, è fastidioso. Puoi raggruppare intere espressioni tra normali parentesi tonde, e il carattere `|` ti farà trovare *o* l'espressione alla sua destra *o* quella alla sua sinistra. Combinale ed è fatta. Ho spiegato troppo in fretta? Va bene, per prima cosa raggruppiamo le espressioni per le pagine pari e dispari, ottenendo così le due espressioni che ci servono `(Titolo) (Autore)`. Ora rendiamo le cose più semplici usando la barra verticale (`|` è detto carattere di barra verticale): Se usi l'espressione `(Titolo) | (Autore)` troverai o un risultato per «Titolo» (nelle pagine dispari) o un risultato per «Autore» (nelle pagine pari). Abbastanza semplice, no?

Naturalmente puoi usare la barra verticale anche senza le parentesi di raggruppamento. Ricordi quando ho detto che i quantificatori ripetono l'elemento che li precede? Ecco, la barra verticale funziona in un modo leggermente diverso: Anche l'espressione «TitoloAutore» troverà o la stringa «Titolo» o la stringa «Autore», esattamente come l'esempio precedente con il raggruppamento. *La barra verticale sceglie tra l'intera espressione che la precede e quella che la segue.* Quindi, se volessi trovare le stringhe «Calibre» e «calibre» e volessi distinguere solo tra la «c» maiuscola e minuscola, dovresti usare l'espressione `(c|C)alibre`, in cui il raggruppamento fa in modo che solo la «c» venga selezionata. Se invece avessi usato `c|Calibre`, avresti ottenuto una corrispondenza con la stringa «c» o con la stringa «Calibre», che non è quello che volevamo. In breve: Nel dubbio, usa il raggruppamento insieme alla barra verticale.

10.4.10 Ti sei dimenticato...

... fermo fermo fermo, c'è un'ultima cosa molto utile che puoi fare con i gruppi. Se hai catturato un gruppo in precedenza, puoi riferirti a quel gruppo più avanti nell'espressione stessa: I gruppi sono numerati a partire da 1, e puoi riferirti ad essi con il loro numero. Il quinto gruppo, per esempio, sarebbe richiamato da \5. Quindi, se cercassi `([^]+) \1` nella stringa «Test Test», troveresti l'intera stringa!

10.4.11 All'inizio hai detto che c'era un modo di creare un'espressione regolare che non distingue le maiuscole?

Sì, l'ho fatto, grazie per aver prestato attenzione e avermelo ricordato. Puoi dire a Calibre come vuoi che certe cose vengano gestite usando qualcosa chiamato flag. Puoi includere i flag nella tua espressione utilizzando il costrutto speciale `(?flags go here)` dove, ovviamente, sostituirai «flags go here» con i flag specifici che desideri. Per ignorare le maiuscole e minuscole, il flag è `i`, quindi includi `(?i)` nella tua espressione. Pertanto, `(?i)test` corrisponderà a «Test», «tEst», «TEst» e qualsiasi variazione di maiuscole e minuscole a cui potresti pensare.

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 Penso di stare iniziando a capire queste espressioni regolari... Come le posso usare in calibre?

Conversioni

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](#)⁹¹). You'd have to remove some of the tags as well. In this example, I'd recommend beginning with the tag `<b class="calibre2">`, now you have to end with the corresponding closing tag (opening tags are `<tag>`, closing tags are `</tag>`), which is simply the next `` in this case. (Refer to a good HTML manual or ask in the forum if you are unclear on this point). The opening tag can be described using `<b.*?>`, the closing tag using ``, thus we could remove everything between those tags using `<b.*?>.*?`. But using this expression would be a bad idea, because it removes everything enclosed by ``-tags (which, by the way, render the enclosed text in bold print), and it's a fair bet that we'll remove portions of the book in this way. Instead, include the beginning of the enclosed string as well, making the regular expression `<b.*?>\s*Generated\s+by\s+ABC\s+Amber\s+LIT.*?`. The `\s` with quantifiers are included here instead of explicitly using the spaces as seen in the string to catch any variations of the string that might occur. Remember to check what calibre will remove to make sure you don't remove any portions you

⁹¹ <https://www.mobileread.com/forums/showthread.php?t=75594>

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.

Aggiunta di libri

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.

Modifica di gruppo dei metadati

L'ultima parte è l'espressione regolare *Cerca e sostituisci* nei campi dei metadati. Puoi accedervi selezionando più libri nella libreria e utilizzando la modifica collettiva dei metadati. Fai molta attenzione quando usi quest'ultima funzionalità, poiché può causare **cose molto brutte** alla tua libreria! Controlla attentamente che le tue espressioni facciano ciò che desideri utilizzando i campi di test e contrassegna solo i libri che desideri veramente modificare! Nella modalità di ricerca delle espressioni regolari, puoi cercare in un campo, sostituire il testo con qualcosa e persino scrivere il risultato in un altro campo. Un esempio pratico: supponiamo che la tua biblioteca contenga i libri della serie Dune di Frank Herbert, dal nome della moda «Dune 1 - Dune», «Dune 2 - Dune Messiah» e così via. Ora vuoi inserire `Dune` nel campo della serie. Puoi farlo cercando `(.*?) \d - .*` nel campo del titolo e sostituendolo con `\1` nel campo della serie. Vedi cosa ho fatto lì? Questo è un riferimento al primo gruppo con cui stai sostituendo il campo della serie. Ora che hai impostato la serie, devi solo fare un'altra ricerca per `.*?` - nel campo del titolo e sostituiscilo con `" "` (una stringa vuota), sempre nel campo del titolo, e i tuoi metadati saranno tutti puliti e ordinati. Non è fantastico? A proposito, invece di sostituire l'intero campo, puoi anche aggiungere o anteporre al campo, quindi, se *volevi* che il titolo del libro fosse anteposto alle informazioni sulla serie, potresti farlo anche tu. Come avrai sicuramente notato, c'è una casella etichettata *Maiuscole e minuscole*, quindi non dovrai usare i flag per selezionare il comportamento qui.

Bene, con ciò abbiamo quasi finito questa breve introduzione alle espressioni regolari. Con un po' di fortuna ti avrò mostrato abbastanza per farti almeno partire e permetterti di continuare a imparare per conto tuo - un buon posto dove guardare sarebbe [Python documentation for regexps](https://docs.python.org/library/re.html)⁹².

Un ultimo avviso, però: nonostante le espressioni regolari siano molto potenti, è anche molto facile sbagliare nel crearle. calibre fornisce ottime funzioni di prova per controllare se le tue espressioni funzionano come ti aspetti. Usale. Cerca di evitare di spararti nel piede. (Cielo, quanto mi piace questa espressione...). Ma anche se dovessi, nonostante i preavvisi, ferirti al piede (o in altre parti anatomiche), prova ad imparare da quanto successo.

⁹² <https://docs.python.org/library/re.html>

10.4.13 Riferimenti rapidi

Riferimenti rapidi sulla sintassi regex

Questo elenco riassume le parti più comunemente usate/più difficili da ricordare del motore regex disponibile in molte delle parti di calibre.

Contenuti

- *Classi di caratteri* (pagina 211)
- *Classi di caratteri abbreviate* (pagina 212)
- *I quantificatori* (pagina 212)
- *Voracità* (pagina 212)
- *Alternative* (pagina 212)
- *Esclusione* (pagina 213)
- *Ancoraggi* (pagina 213)
- *Gruppi* (pagina 213)
- *Lookaround* (pagina 214)
- *Ricorsività* (pagina 214)
- *Caratteri speciali* (pagina 215)
- *Metacaratteri* (pagina 215)
- *Modalità* (pagina 215)

Classi di caratteri

Le classi di caratteri sono utili per rappresentare gruppi diversi di caratteri in modo conciso.

Esempi:

Rappresen- tazione	Classe
[a-z]	Lettere minuscole. Non include caratteri accentati, segnati e legature
[a-z0-9]	Lettere minuscole dalla a alla z o numeri da 0 a 9
[A-Za-z-]	Lettere maiuscole, minuscole o un trattino. Per includere il trattino in una classe devi metterlo all'inizio oppure alla fine, per non confonderlo con quello che specifica un intervallo di caratteri.
[^0-9]	Qualsiasi carattere tranne una cifra. Il segno di omissione (^) all'inizio della classe esclude i caratteri della classe (classe complementare)
[[a-z] -- [a	Consonanti minuscole. Una classe può essere inclusa in una classe. I caratteri – escludono ciò che li segue
[\w -- [\d_]]	Tutte le lettere (inclusi i caratteri accentati stranieri). Le classi abbreviate possono essere utilizzate all'interno di una classe

Esempio:


```
<[<>]+> to select an HTML tag
```

Classi di caratteri abbreviate

Rappresen- tazione	Classe
\d	Una cifra (uguale a [0-9])
\D	Ogni carattere non numerico (uguale a [^0-9])
\w	Un carattere alfanumerico ([a-zA-Z0-9]) inclusi caratteri con accenti, segni e legature.
\W	Qualsiasi carattere «non-parola»
\s	Spazio, spazio unificatore, rientro (tabulazione), testo a capo
\S	Qualsiasi carattere «non-vuoto»
.	Qualsiasi carattere eccetto newline. Usa la casella «Il punto vale tutto» o il modificatore regex (?s) per includere il carattere newline.

I quantificatori

Quantificatore	Numero di occorrenze dell'espressione che precede il quantificatore
?	0 o 1 occorrenze dell'espressione. Uguale a {0, 1}
+	1 o più occorrenze dell'espressione. Uguale a {1, }
*	0, 1 o più occorrenze dell'espressione. Uguale a {0, }
{n}	Esattamente n occorrenze dell'espressione
{min,max}	Un numero di occorrenze compreso tra i valori minimo e massimo inclusi
{min, }	Un numero di occorrenze compreso tra il valore minimo e l'infinito
{, max}	Un numero di occorrenze compreso tra 0 e il valore massimo incluso

Voracità

Per impostazione predefinita, con i quantificatori, il motore delle espressioni regolari è vorace (greedy): estende la selezione il più possibile. Questo causa delle sorprese all'inizio. ? segue un quantificatore per renderlo pigro (lazy). Evita di metterne due nella stessa espressione, i risultati possono essere imprevedibili.

Fai attenzione nel nidificare i quantificatori, come ad esempio nell'espressione (a*)*, perché fare ciò aumenta esponenzialmente il tempo di elaborazione.

Alternative

Il carattere | in un'espressione regolare è un OR logico. Significa che verrà cercata una corrispondenza o per l'espressione precedente o per quella seguente.

Esclusione

Metodo 1

```
schema_da_escludere (*SKIP) (*FAIL) | schema_da_selezionare
```

Esempio:

```
"Blabla" (*SKIP) (*FAIL) | Blabla
```

seleziona Blabla, nelle stringhe Blabla o «Blabla or Blabla», ma non in «Blabla».

Metodo 2

```
schema_da_escludere \K| (schema_da_selezionare)
```

```
"Blabla" \K| (Blabla)
```

seleziona Blabla, nelle stringhe Blabla o «Blabla or Blabla», ma non in «Blabla».

Ancoraggi

Un ancoraggio è un modo di trovare una posizione logica in una stringa, invece di un carattere. Gli ancoraggi più utili per l'elaborazione del testo sono:

\b

Indica il confine di una parola, ovvero una transizione da un carattere spazio a uno non-spazio. Per esempio, puoi usare `\bcerto` per trovare `di certo` ma non `concerto`.

^

Trova l'inizio di una riga (in modalità multilinea, che è quella predefinita)

\$

Trova la fine di una riga (in modalità multilinea, che è quella predefinita)

\K

Ripristina la posizione di partenza della selezione alla sua posizione nello schema. Alcuni motori di regex (ma non quello di calibre) non permettono lookbehind di lunghezza variabile, specialmente con i quantificatori. Se puoi utilizzare `\K` con questi motori, puoi anche aggirare questa limitazione scrivendo l'equivalente di un lookbehind positivo di lunghezza variabile.

Gruppi

(espressione)

Gruppo di cattura, che conserva la selezione e può essere richiamato in seguito nelle espressioni *cerca* e *sostituisci* usando `\n`, dove `n` è il numero di sequenza del gruppo di cattura (a partire da 1 in ordine di lettura)

(?:espressione)

Gruppo che non cattura la selezione

(?>espressione)

Gruppo Atomico: Non appena l'espressione è soddisfatta, il motore regex prosegue, e se il resto dello schema fallisce, non tornerà indietro per provare altre combinazioni con l'espressione. I gruppi atomici non sono di cattura.

(?|espressione)

Gruppo branch reset: i rami (branches) delle alternative incluse nell'espressione condividono gli stessi numeri di gruppo

(?<nome>espressione)

Gruppo chiamato “nome”. La selezione può essere richiamata in seguito nell’espressione *cerca* con (?P=nome) e in quella *sostituisci* con \g<nome>. Due gruppi diversi possono usare lo stesso nome.

Lookaround

Lookaround	Significato
?=	Lookahead positivo (da inserire dopo la selezione)
?!	Lookahead negativo (da inserire dopo la selezione)
?<=	Lookbehind positivo (da inserire prima della selezione)
?<!	Lookbehind negativo (da inserire prima della selezione)

I lookahead e i lookbehind non consumano caratteri, hanno lunghezza zero e non catturano. Sono gruppi atomici: non appena l’asserzione è soddisfatta, il motore regex prosegue, e se il resto dello schema fallisce, non tornerà indietro all’interno del lookaround per provare altre combinazioni.

Quando si cercano più corrispondenze in una stringa, alla posizione di partenza di ogni tentativo di corrispondenza, un lookbehind può esaminare i caratteri precedenti la posizione corrente. Di conseguenza, nella stringa 123, lo schema (?<=\d)\d (una cifra preceduta da una cifra) dovrebbe in teoria selezionare 2 e 3. D’altra parte \d\K\d può solo selezionare 2, perché la posizione di partenza dopo la prima selezione si trova immediatamente prima del 3, e non ci sono abbastanza cifre per una seconda corrispondenza. Similmente, \d(\d) cattura solo 2. Nella pratica del motore regex di calibre, il lookbehind positivo si comporta allo stesso modo, e seleziona solo 2, contrariamente alla teoria.

I gruppi possono essere inseriti all’interno dei lookaround, ma la cattura è raramente utile. Cionondimeno, se necessaria, bisognerà fare molta attenzione nell’uso di un quantificatore in un lookbehind: la voracità associata al non tornare indietro può dare una cattura sorprendente. Per questa ragione, usa \K piuttosto che un lookbehind positivo quando hai un quantificatore (o peggio, più di uno) in un gruppo di cattura del lookbehind positivo.

Esempio di lookahead negativo:

```
(?! [^<>{}]* [>{}])
```

Inserito alla fine dell’espressione impedisce la selezione dentro un tag o uno stile incorporato nel file.

Quando possibile, è sempre meglio «ancorare» i lookaround, per ridurre il numero di passi necessari ad ottenere il risultato.

Ricorsività

Rappresentazione	Significato
(?R)	Ricorsività dell’intero schema
(?1)	Ricorsività solamente dello schema del gruppo di cattura numerato, qui il gruppo 1

Ricorsività qui significa invocare se stessi. Ciò è utile per richieste equilibrate, come le stringhe tra virgolette, che possono avere incorporate stringhe tra virgolette. Quindi, se durante l’elaborazione di una stringa tra virgolette doppie incontriamo l’inizio di una nuova stringa compresa tra due virgolette doppie, sappiamo cosa fare, e invochiamo noi stessi. E abbiamo uno schema di questo tipo:

```
start-pattern(?>atomic sub-pattern|(?R))*end-pattern
```

Per selezionare una stringa compresa tra due virgolette doppie senza fermarci su una stringa incorporata:

```
" ( (?>[^\"]+|(?R)) * [^\"]+ ) "
```

Questo modello può anche essere usato per modificare coppie di tag che possono essere incorporati, come i tag `<div>`.

Caratteri speciali

Rappresentazione	Carattere
<code>\t</code>	rientro (tabulazione)
<code>\n</code>	interruzione di linea
<code>\x20</code>	spazio (divisibile)
<code>\xa0</code>	spazio unificatore (nbsp)

Metacaratteri

I metacaratteri sono quelli che hanno un significato speciale per il motore di regex. Di questi, dodici devono essere preceduti da un carattere di escape, la barra rovesciata (`\`), per perdere il loro significato speciale e ridiventare caratteri normali:

```
^ . [ ] $ ( ) * + ? | \
```

Sette altri metacaratteri non hanno bisogno di essere preceduti da una barra rovesciata (ma lo possono essere senza altre conseguenze):

```
{ } ! < > = :
```

I caratteri speciali perdono il loro stato se vengono usati dentro una classe (tra le parentesi `[]`). La parentesi di chiusura e il trattino hanno uno stato speciale dentro una classe. Fuori dalla classe, il trattino è una semplice lettera, mentre la parentesi di chiusura rimane un metacarattere.

La barra (`/`) e il cancelletto (o hashtag) (`#`) non sono metacaratteri, non hanno bisogno del carattere di escape.

In alcuni strumenti, come `regex101.com` con il motore Python, le virgolette doppie hanno lo stato speciale di separatore e devono essere precedute dal carattere di escape, oppure vanno cambiate le opzioni. Così non è nell'editor di calibre.

Modalità

(?s)

Fa in modo che il punto (`.`) trovi anche i caratteri newline

(?m)

Fa in modo che gli ancoraggi `^` e `$` trovino l'inizio e la fine delle righe invece dell'inizio e della fine dell'intera stringa.

10.4.14 Riconoscimenti

Grazie per aver contribuito con consigli, correzioni e altro:

- Idolse
- kovidgoyal
- chaley
- dwanthny
- kacir
- Starson17
- Orpheu

Per più informazioni sulle regex vedi [The Python User Manual](#)⁹³. In realtà la libreria usata da calibre per le espressioni regolari è [regex](#)⁹⁴, che supporta vari utili miglioramenti rispetto alla libreria standard di Python.

10.5 Scrivere i propri plugin per estendere le funzionalità di calibre

calibre ha una struttura estremamente modulare. Quasi tutte le funzionalità di calibre agiscono sotto forma di plugin. Dei plugin sono usati per le conversioni, per scaricare le notizie (sebbene questi siano chiamati ricette), per molte componenti dell'interfaccia utente, per connettersi a svariati dispositivi, per elaborare i file aggiunti a calibre e così via. Puoi ottenere una lista completa dei plugin inclusi in calibre andando in *Preferenze* → *Avanzate* → *Plugin*.

Qui, ti insegneremo come creare i tuoi plugin per aggiungere nuove funzioni a calibre.

Contenuti

- *Anatomia di un plugin di calibre* (pagina 217)
- *Un plugin dell'interfaccia utente* (pagina 218)
 - *__init__.py* (pagina 219)
 - *ui.py* (pagina 220)
 - *main.py* (pagina 221)
 - *Getting resources from the plugin ZIP file* (pagina 224)
 - *Enabling user configuration of your plugin* (pagina 224)
- *Edit book plugins* (pagina 226)
 - *main.py* (pagina 227)
- *Aggiungi traduzione al tuo plugin* (pagina 229)
- *The plugin API* (pagina 230)
- *Debugging plugins* (pagina 230)
- *Più esempi di plugin* (pagina 231)
- *Condividi i tuoi plugin con gli altri* (pagina 231)

⁹³ <https://docs.python.org/library/re.html>

⁹⁴ <https://bitbucket.org/mrabarnett/mrab-regex/src/hg/>

Nota: Si applica solo alle versioni di calibre $\geq 0.8.60$

10.5.1 Anatomia di un plugin di calibre

Un plugin di calibre è molto semplice, è solo un file ZIP che contiene del codice in Python e qualsiasi altra risorsa necessaria al plugin, come le immagini. Senza indugiare oltre, vediamo un semplice esempio.

Immagina di avere un'installazione di calibre che usi per l'autopubblicazione di vari documenti elettronici nei formati EPUB e MOBI. Vorresti fare in modo che tutti i file generati da calibre avessero «Hello world» impostato come editore, ecco il modo di farlo. Crea un file con il nome `__init__.py` (questo è un nome speciale che deve sempre essere usato per il file principale del tuo plugin) e inseriscivi il seguente codice Python:

```
from calibre.customize import FileTypePlugin

class HelloWorld(FileTypePlugin):

    name = 'Hello World Plugin' # Name of the plugin
    description = 'Set the publisher to Hello World for all new conversions'
    supported_platforms = ['windows', 'osx', 'linux'] # Platforms this plugin will
    ↪run on
    author = 'Acme Inc.' # The author of this plugin
    version = (1, 0, 0) # The version number of this plugin
    file_types = {'epub', 'mobi'} # The file types that this plugin will be
    ↪applied to
    on_postprocess = True # Run this plugin after conversion is complete
    minimum_calibre_version = (0, 7, 53)

    def run(self, path_to_ebook):
        from calibre.ebooks.metadata.meta import get_metadata, set_metadata
        with open(path_to_ebook, 'r+b') as file:
            ext = os.path.splitext(path_to_ebook)[-1][1:].lower()
            mi = get_metadata(file, ext)
            mi.publisher = 'Hello World'
            set_metadata(file, mi, ext)
        return path_to_ebook
```

Questo è tutto. Per aggiungere questo codice a calibre come plugin, devi solo eseguire il comando che segue nella cartella in cui hai creato `__init__.py`:

```
calibre-customize -b .
```

Nota: In macOS, gli strumenti da linea di comando si trovano dentro il bundle di calibre, per esempio, se hai installato calibre in `/Applications` gli strumenti si trovano in `/Applications/calibre.app/Contents/MacOS/`.

Puoi scaricare il plugin Hello World da [helloworld_plugin.zip](https://calibre-ebook.com/downloads/helloworld_plugin.zip)⁹⁵.

Ogni volta che usi calibre per convertire un libro il metodo `run()` del plugin sarà chiamato, e il libro convertito vedrà il suo editore impostato a «Hello World». Questo è un plugin molto semplice, passiamo a un esempio più complesso che aggiunge realmente un componente all'interfaccia utente.

⁹⁵ https://calibre-ebook.com/downloads/helloworld_plugin.zip

10.5.2 Un plugin dell'interfaccia utente

This plugin will be spread over a few files (to keep the code clean). It will show you how to get resources (images or data files) from the plugin ZIP file, allow users to configure your plugin, how to create elements in the calibre user interface and how to access and query the books database in calibre.

You can download this plugin from [interface_demo_plugin.zip](https://calibre-ebook.com/downloads/interface_demo_plugin.zip)⁹⁶

The first thing to note is that this ZIP file has a lot more files in it, explained below, pay particular attention to `plugin-import-name-interface_demo.txt`.

plugin-import-name-interface_demo.txt

An empty text file used to enable the multi-file plugin magic. This file must be present in all plugins that use more than one .py file. It should be empty and its filename must be of the form: `plugin-import-name-some_name.txt`. The presence of this file allows you to import code from the .py files present inside the ZIP file, using a statement like:

```
from calibre_plugins.some_name.some_module import some_object
```

The prefix `calibre_plugins` must always be present. `some_name` comes from the filename of the empty text file. `some_module` refers to `some_module.py` file inside the ZIP file. Note that this importing is just as powerful as regular Python imports. You can create packages and subpackages of .py modules inside the ZIP file, just like you would normally (by defining `__init__.py` in each sub-folder), and everything should «just work».

The name you use for `some_name` enters a global namespace shared by all plugins, **so make it as unique as possible**. But remember that it must be a valid Python identifier (only alphabets, numbers and the underscore).

__init__.py

As before, the file that defines the plugin class

main.py

This file contains the actual code that does something useful

ui.py

Questo file definisce la parte dell'interfaccia del plugin

images/icon.png

L'icona per questo plugin

about.txt

File di testo con informazioni sul plugin

traduzioni

La cartella contiene file .mo con l'interfaccia utente del tuo plugin tradotta in diverse lingue. Guarda più in basso per i dettagli.

Ora diamo un'occhiata al codice.

⁹⁶ https://calibre-ebook.com/downloads/interface_demo_plugin.zip

`__init__.py`

First, the obligatory `__init__.py` to define the plugin metadata:

```

from calibre.customize import InterfaceActionBase

class InterfacePluginDemo(InterfaceActionBase):
    '''
    This class is a simple wrapper that provides information about the actual
    plugin class. The actual interface plugin class is called InterfacePlugin
    and is defined in the ui.py file, as specified in the actual_plugin field
    below.

    The reason for having two classes is that it allows the command line
    calibre utilities to run without needing to load the GUI libraries.
    '''
    name = 'Interface Plugin Demo'
    description = 'An advanced plugin demo'
    supported_platforms = ['windows', 'osx', 'linux']
    author = 'Kovid Goyal'
    version = (1, 0, 0)
    minimum_calibre_version = (0, 7, 53)

    #: This field defines the GUI plugin class that contains all the code
    #: that actually does something. Its format is module_path:class_name
    #: The specified class must be defined in the specified module.
    actual_plugin = 'calibre_plugins.interface_demo.ui:InterfacePlugin'

    def is_customizable(self):
        '''
        This method must return True to enable customization via
        Preferences->Plugins
        '''
        return True

    def config_widget(self):
        '''
        Implement this method and :meth:`save_settings` in your plugin to
        use a custom configuration dialog.

        This method, if implemented, must return a QWidget. The widget can have
        an optional method validate() that takes no arguments and is called
        immediately after the user clicks OK. Changes are applied if and only
        if the method returns True.

        If for some reason you cannot perform the configuration at this time,
        return a tuple of two strings (message, details), these will be
        displayed as a warning dialog to the user and the process will be
        aborted.

        The base class implementation of this method raises NotImplementedError
        so by default no user configuration is possible.
        '''
        # It is important to put this import statement here rather than at the
        # top of the module as importing the config class will also cause the
        # GUI libraries to be loaded, which we do not want when using calibre
        # from the command line
        from calibre_plugins.interface_demo.config import ConfigWidget

```

(continues on next page)

(continua dalla pagina precedente)

```

    return ConfigWidget()

def save_settings(self, config_widget):
    '''
    Save the settings specified by the user with config_widget.

    :param config_widget: The widget returned by :meth:`config_widget`.
    '''
    config_widget.save_settings()

    # Apply the changes
    ac = self.actual_plugin_
    if ac is not None:
        ac.apply_settings()

```

The only noteworthy feature is the field `actual_plugin`. Since calibre has both command line and GUI interfaces, GUI plugins like this one should not load any GUI libraries in `__init__.py`. The `actual_plugin` field does this for you, by telling calibre that the actual plugin is to be found in another file inside your ZIP archive, which will only be loaded in a GUI context.

Remember that for this to work, you must have a `plugin-import-name-some_name.txt` file in your plugin ZIP file, as discussed above.

Also there are a couple of methods for enabling user configuration of the plugin. These are discussed below.

ui.py

Now let's look at `ui.py` which defines the actual GUI plugin. The source code is heavily commented and should be self explanatory:

```

from calibre.gui2.actions import InterfaceAction
from calibre_plugins.interface_demo.main import DemoDialog

class InterfacePlugin(InterfaceAction):

    name = 'Interface Plugin Demo'

    # Declare the main action associated with this plugin
    # The keyboard shortcut can be None if you dont want to use a keyboard
    # shortcut. Remember that currently calibre has no central management for
    # keyboard shortcuts, so try to use an unusual/unused shortcut.
    action_spec = ('Interface Plugin Demo', None,
                  'Run the Interface Plugin Demo', 'Ctrl+Shift+F1')

    def genesis(self):
        # This method is called once per plugin, do initial setup here

        # Set the icon for this interface action
        # The get_icons function is a builtin function defined for all your
        # plugin code. It loads icons from the plugin zip file. It returns
        # QIcon objects, if you want the actual data, use the analogous
        # get_resources builtin function.
        #

```

(continues on next page)

(continua dalla pagina precedente)

```

# Note that if you are loading more than one icon, for performance, you
# should pass a list of names to get_icons. In this case, get_icons
# will return a dictionary mapping names to QIcons. Names that
# are not found in the zip file will result in null QIcons.
icon = get_icons('images/icon.png', 'Interface Demo Plugin')

# The qaction is automatically created from the action_spec defined
# above
self.qaction.setIcon(icon)
self.qaction.triggered.connect(self.show_dialog)

def show_dialog(self):
    # The base plugin object defined in __init__.py
    base_plugin_object = self.interface_action_base_plugin
    # Show the config dialog
    # The config dialog can also be shown from within
    # Preferences->Plugins, which is why the do_user_config
    # method is defined on the base plugin class
    do_user_config = base_plugin_object.do_user_config

    # self.gui is the main calibre GUI. It acts as the gateway to access
    # all the elements of the calibre user interface, it should also be the
    # parent of the dialog
    d = DemoDialog(self.gui, self.qaction.icon(), do_user_config)
    d.show()

def apply_settings(self):
    from calibre_plugins.interface_demo.config import prefs
    # In an actual non trivial plugin, you would probably need to
    # do something based on the settings in prefs
    prefs

```

main.py

The actual logic to implement the Interface Plugin Demo dialog.

```

from 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()

```

(continues on next page)

(continua dalla pagina precedente)

```

self.setLayout(self.l)

self.label = QLabel(prefs['hello_world_msg'])
self.l.addWidget(self.label)

self.setWindowTitle('Interface Plugin Demo')
self.setWindowIcon(icon)

self.about_button = QPushButton('About', self)
self.about_button.clicked.connect(self.about)
self.l.addWidget(self.about_button)

self.marked_button = QPushButton(
    'Show books with only one format in the calibre GUI', self)
self.marked_button.clicked.connect(self.marked)
self.l.addWidget(self.marked_button)

self.view_button = QPushButton(
    'View the most recently added book', self)
self.view_button.clicked.connect(self.view)
self.l.addWidget(self.view_button)

self.update_metadata_button = QPushButton(
    'Update metadata in a book\'s files', self)
self.update_metadata_button.clicked.connect(self.update_metadata)
self.l.addWidget(self.update_metadata_button)

self.conf_button = QPushButton(
    'Configure this plugin', self)
self.conf_button.clicked.connect(self.config)
self.l.addWidget(self.conf_button)

self.resize(self.sizeHint())

def about(self):
    # Get the about text from a file inside the plugin zip file
    # The get_resources function is a builtin function defined for all your
    # plugin code. It loads files from the plugin zip file. It returns
    # the bytes from the specified file.
    #
    # Note that if you are loading more than one file, for performance, you
    # should pass a list of names to get_resources. In this case,
    # get_resources will return a dictionary mapping names to bytes. Names that
    # are not found in the zip file will not be in the returned dictionary.
    text = get_resources('about.txt')
    QMessageBox.about(self, 'About the Interface Plugin Demo',
        text.decode('utf-8'))

def marked(self):
    ''' Show books with only one format '''
    db = self.db.new_api
    matched_ids = {book_id for book_id in db.all_book_ids() if len(db.
→formats(book_id)) == 1}
    # Mark the records with the matching ids
    # new_api does not know anything about marked books, so we use the full
    # db object
    self.db.set_marked_ids(matched_ids)

```

(continues on next page)

(continua dalla pagina precedente)

```

# Tell the GUI to search for all marked records
self.gui.search.setEditText('marked:true')
self.gui.search.do_search()

def view(self):
    ''' View the most recently added book '''
    most_recent = most_recent_id = None
    db = self.db.new_api
    for book_id, timestamp in db.all_field_for('timestamp', db.all_book_ids()).
→items():
        if most_recent is None or timestamp > most_recent:
            most_recent = timestamp
            most_recent_id = book_id

    if most_recent_id is not None:
        # Get a reference to the View plugin
        view_plugin = self.gui.iactions['View']
        # Ask the view plugin to launch the viewer for row_number
        view_plugin._view_calibre_books([most_recent_id])

def update_metadata(self):
    '''
    Set the metadata in the files in the selected book's record to
    match the current metadata in the database.
    '''
    from calibre.ebooks.metadata.meta import set_metadata
    from calibre.gui2 import error_dialog, info_dialog

    # Get currently selected books
    rows = self.gui.library_view.selectionModel().selectedRows()
    if not rows or len(rows) == 0:
        return error_dialog(self.gui, 'Cannot update metadata',
                            'No books selected', show=True)

    # Map the rows to book ids
    ids = list(map(self.gui.library_view.model().id, rows))
    db = self.db.new_api
    for book_id in ids:
        # Get the current metadata for this book from the db
        mi = db.get_metadata(book_id, get_cover=True, cover_as_data=True)
        fmts = db.formats(book_id)
        if not fmts:
            continue
        for fmt in fmts:
            fmt = fmt.lower()
            # Get a python file object for the format. This will be either
            # an in memory file or a temporary on disk file
            ffile = db.format(book_id, fmt, as_file=True)
            ffile.seek(0)
            # Set metadata in the format
            set_metadata(ffile, mi, fmt)
            ffile.seek(0)
            # Now replace the file in the calibre library with the updated
            # file. We dont use add_format_with_hooks as the hooks were
            # already run when the file was first added to calibre.
            db.add_format(book_id, fmt, ffile, run_hooks=False)

```

(continues on next page)

(continua dalla pagina precedente)

```
info_dialog(self, 'Updated files',
            'Updated the metadata in the files of %d book(s)' % len(ids),
            show=True)

def config(self):
    self.do_user_config(parent=self)
    # Apply the changes
    self.label.setText(prefs['hello_world_msg'])
```

Getting resources from the plugin ZIP file

calibre's plugin loading system defines a couple of built-in functions that allow you to conveniently get files from the plugin ZIP file.

`get_resources(name_or_list_of_names)`

This function should be called with a list of paths to files inside the ZIP file. For example to access the file `icon.png` in the folder `images` in the ZIP file, you would use: `images/icon.png`. Always use a forward slash as the path separator, even on Windows. When you pass in a single name, the function will return the raw bytes of that file or `None` if the name was not found in the ZIP file. If you pass in more than one name then it returns a dictionary mapping the names to bytes. If a name is not found, it will not be present in the returned dictionary.

`get_icons(nome_o_lista_di_nomi, nome_plugin='')`

A wrapper for `get_resources()` that creates `QIcon` objects from the raw bytes returned by `get_resources`. If a name is not found in the ZIP file the corresponding `QIcon` will be null. In order to support icon theme-ing, pass in the human friendly name of your plugin as `plugin_name`. If the user is using an icon theme with icons for your plugin, they will be loaded preferentially.

Enabling user configuration of your plugin

To allow users to configure your plugin, you must define three methods in your base plugin class, `is_customizable`, `config_widget` and `save_settings` as shown below:

```
def is_customizable(self):
    '''
    This method must return True to enable customization via
    Preferences->Plugins
    '''
    return True
```

```
def config_widget(self):
    '''
    Implement this method and :meth:`save_settings` in your plugin to
    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
```

(continues on next page)

(continua dalla pagina precedente)

`aborted.`*The base class implementation of this method raises NotImplementedError so by default no user configuration is possible.*`'''`*# It is important to put this import statement here rather than at the top of the module as importing the config class will also cause the GUI libraries to be loaded, which we do not want when using calibre from the command line*`from calibre_plugins.interface_demo.config import ConfigWidget
return ConfigWidget()``def save_settings(self, config_widget):``'''`*Save the settings specified by the user with config_widget.**:param config_widget: The widget returned by :meth:`config_widget`.*
`'''``config_widget.save_settings()`*# Apply the changes*`ac = self.actual_plugin_``if ac is not None:` `ac.apply_settings()`

calibre has many different ways to store configuration data (a legacy of its long history). The recommended way is to use the **JSONConfig** class, which stores your configuration information in a .json file.

The code to manage configuration data in the demo plugin is in config.py:

`from calibre.utils.config import JSONConfig`*# This is where all preferences for this plugin will be stored
Remember that this name (i.e. plugins/interface_demo) is also in a global namespace, so make it as unique as possible.
You should always prefix your config file name with plugins/, so as to ensure you dont accidentally clobber a calibre config file*
`prefs = JSONConfig('plugins/interface_demo')`*# Set defaults*`prefs.defaults['hello_world_msg'] = 'Hello, World!'``class ConfigWidget (QWidget):` `def __init__(self):` `QWidget.__init__(self)` `self.l = QHBoxLayout()` `self.setLayout(self.l)` `self.label = QLabel('Hello world &message:')` `self.l.addWidget(self.label)` `self.msg = QLineEdit(self)` `self.msg.setText(prefs['hello_world_msg'])` `self.l.addWidget(self.msg)`

(continues on next page)

(continua dalla pagina precedente)

```
self.label.setBuddy(self.msg)

def save_settings(self):
    prefs['hello_world_msg'] = self.msg.text()
```

The prefs object is now available throughout the plugin code by a simple:

```
from calibre_plugins.interface_demo.config import prefs
```

You can see the prefs object being used in main.py:

```
def config(self):
    self.do_user_config(parent=self)
    # Apply the changes
    self.label.setText(prefs['hello_world_msg'])
```

10.5.3 Edit book plugins

Now let's change gears for a bit and look at creating a plugin to add tools to the calibre book editor. The plugin is available here: [editor_demo_plugin.zip](#)⁹⁷.

The first step, as for all plugins is to create the import name empty txt file, as described [above](#) (pagina 218). We shall name the file plugin-import-name-editor_plugin_demo.txt.

Now we create the mandatory `__init__.py` file that contains metadata about the plugin – its name, author, version, etc.

```
from calibre.customize import EditBookToolPlugin

class DemoPlugin(EditBookToolPlugin):

    name = 'Edit Book plugin demo'
    version = (1, 0, 0)
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'osx', 'linux']
    description = 'A demonstration of the plugin interface for the ebook editor'
    minimum_calibre_version = (1, 46, 0)
```

A single editor plugin can provide multiple tools each tool corresponds to a single button in the toolbar and entry in the *Plugins* menu in the editor. These can have sub-menus in case the tool has multiple related actions.

The tools must all be defined in the file `main.py` in your plugin. Every tool is a class that inherits from the `calibre.gui2.tweak_book.plugin.Tool` (pagina 361) 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` (pagina 361) class for more details.

⁹⁷ https://calibre-ebook.com/downloads/editor_demo_plugin.zip

main.py

Here we will see the definition of a single tool that will multiply all font sizes in the book by a number provided by the user. This tool demonstrates various important concepts that you will need in developing your own plugins, so you should read the (heavily commented) source code carefully.

```
import re
from qt.core import QAction, QDialog
from css_parser.css import CSSRule

# The base class that all tools must inherit from
from calibre.gui2.tweak_book.plugin import Tool

from calibre import force_unicode
from calibre.gui2 import error_dialog
from calibre.ebooks.oeb.polish.container import OEB_DOCS, OEB_STYLES, serialize

class DemoTool(Tool):

    #: Set this to a unique name it will be used as a key
    name = 'demo-tool'

    #: If True the user can choose to place this tool in the plugins toolbar
    allowed_in_toolbar = True

    #: If True the user can choose to place this tool in the plugins menu
    allowed_in_menu = True

    def create_action(self, for_toolbar=True):
        # Create an action, this will be added to the plugins toolbar and
        # the plugins menu
        ac = QAction(get_icons('images/icon.png'), 'Magnify fonts', self.gui) # noqa
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
            # register it for the action created for the menu, not the toolbar,
            # to avoid a double trigger
            self.register_shortcut(ac, 'magnify-fonts-tool', default_keys=(
↳ 'Ctrl+Shift+Alt+D',))
        ac.triggered.connect(self.ask_user)
        return ac

    def ask_user(self):
        # Ask the user for a factor by which to multiply all font sizes
        factor, ok = QDialog.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
```

(continues on next page)

(continua dalla pagina precedente)

```

        error_dialog(self.gui, _('Failed to magnify fonts'), _('
            Failed to magnify fonts, click "Show details" for more info'),
            det_msg=traceback.format_exc(), show=True)
        # Revert to the saved restore point
        self.boss.revert_requested(self.boss.global_undo.previous_container)
    else:
        # Show the user what changes we have made, allowing her to
        # revert them if necessary
        self.boss.show_current_diff()
        # Update the editor UI to take into account all the changes we
        # have made
        self.boss.apply_container_update_to_gui()

    def magnify_fonts(self, factor):
        # Magnify all font sizes defined in the book by the specified factor
        # First we create a restore point so that the user can undo all changes
        # we make.
        self.boss.add_savepoint('Before: Magnify fonts')

        container = self.current_container # The book being edited as a container_
↪object

        # Iterate over all style declarations in the book, this means css
        # stylesheets, <style> tags and style="" attributes
        for name, media_type in container.mime_map.items():
            if media_type in OEB_STYLES:
                # A stylesheet. Parsed stylesheets are css_parser CSSStyleSheet
                # objects.
                self.magnify_stylesheet(container.parsed(name), factor)
                container.dirty(name) # Tell the container that we have changed the_
↪stylesheet
            elif media_type in OEB_DOCS:
                # A HTML file. Parsed HTML files are lxml elements

                for style_tag in container.parsed(name).xpath('//*[local-name="style"]
↪'):
                    if style_tag.text and style_tag.get('type', None) in {None, 'text/
↪css'}:
                        # We have an inline CSS <style> tag, parse it into a
                        # stylesheet object
                        sheet = container.parse_css(style_tag.text)
                        self.magnify_stylesheet(sheet, factor)
                        style_tag.text = serialize(sheet, 'text/css', pretty_
↪print=True)
                        container.dirty(name) # Tell the container that we have_
↪changed the stylesheet
                        for elem in container.parsed(name).xpath('//*[style]'):
                            # Process inline style attributes
                            block = container.parse_css(elem.get('style'), is_
↪declaration=True)
                            self.magnify_declaration(block, factor)
                            elem.set('style', force_unicode(block.getCssText(separator=' '),
↪'utf-8'))

    def magnify_stylesheet(self, sheet, factor):
        # Magnify all fonts in the specified stylesheet by the specified
        # factor.

```

(continues on next page)

(continua dalla pagina precedente)

```

for rule in sheet.cssRules.rulesOfType(CSSRule.STYLE_RULE):
    self.magnify_declaration(rule.style, factor)

def magnify_declaration(self, style, factor):
    # Magnify all fonts in the specified style declaration by the specified
    # factor
    val = style.getPropertyValue('font-size')
    if not val:
        return
    # see if the font-size contains a number
    num = re.search(r'[0-9.]', val)
    if num is not None:
        num = num.group()
        val = val.replace(num, '%f' % (float(num) * factor))
        style.setProperty('font-size', val)
    # We should also be dealing with the font shorthand property and
    # font sizes specified as non numbers, but those are left as exercises
    # for the reader

```

Let's break down `main.py`. We see that it defines a single tool, named *Magnify fonts*. This tool will ask the user for a number and multiply all font sizes in the book by that number.

The first important thing is the tool name which you must set to some relatively unique string as it will be used as the key for this tool.

The next important entry point is the `calibre.gui2.tweak_book.plugin.Tool.create_action()` (pagina 362). 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` (pagina 363) class.

Finally, there is `self.current_container` which is a reference to the book being edited as a `calibre.ebooks.oeb.polish.container.Container` (pagina 354) 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 *Documentazione API per gli strumenti di modifica di un ebook* (pagina 354).

10.5.4 Aggiungi traduzione al tuo plugin

You can have all the user interface strings in your plugin translated and displayed in whatever language is set for the main calibre user interface.

The first step is to go through your plugin's source code and mark all user visible strings as translatable, by surrounding them in `_()`. For example:

```

action_spec = (_('My plugin'), None, _('My plugin is cool'), None)

```

Then use some program to generate .po files from your plugin source code. There should be one .po file for every language you want to translate into. For example: `de.po` for German, `fr.po` for French and so on. You can use the [Poedit](https://poedit.net/)⁹⁸ program for this.

⁹⁸ <https://poedit.net/>

Send these .po files to your translators. Once you get them back, compile them into .mo files. You can again use Poedit for that, or just do:

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

Put the .mo files into the `translations` folder in your plugin.

The last step is to simply call the function `load_translations()` at the top of your plugin's .py files. For performance reasons you should only call this function in those .py files that actually have translatable strings. So in a typical User Interface plugin you would call it at the top of `ui.py` but not `__init__.py`.

You can test the translations of your plugins by changing the user interface language in calibre under *Preferences* → *Interface* → *Look & feel* or by running calibre with the `CALIBRE_OVERRIDE_LANG` environment variable set. For example:

```
CALIBRE_OVERRIDE_LANG=de
```

Replace `de` with the language code of the language you want to test.

For translations with plurals, use the `ngettext()` function instead of `_()`. For example:

```
ngettext('Delete a book', 'Delete {} books', num_books).format(num_books)
```

10.5.5 The plugin API

As you may have noticed above, a plugin in calibre is a class. There are different classes for the different types of plugins in calibre. Details on each class, including the base class of all plugins can be found in [Documentazione API per i plugin](#) (pagina 248).

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 [Vista codice](#) (pagina 338).

10.5.6 Debugging plugins

The first, most important step is to run calibre in debug mode. You can do this from the command line with:

```
calibre-debug -g
```

Or from within calibre by right-clicking the *Preferences* button or using the `Ctrl+Shift+R` keyboard shortcut.

When running from the command line, debug output will be printed to the console, when running from within calibre the output will go to a txt file.

You can insert print statements anywhere in your plugin code, they will be output in debug mode. Remember, this is Python, you really shouldn't need anything more than print statements to debug ;) I developed all of calibre using just this debugging technique.

You can quickly test changes to your plugin by using the following command line:

```
calibre-debug -s; calibre-customize -b /path/to/your/plugin/folder; calibre
```

This will shutdown a running calibre, wait for the shutdown to complete, then update your plugin in calibre and relaunch calibre.

10.5.7 Più esempi di plugin

You can find a list of many sophisticated calibre plugins [here](#)⁹⁹.

10.5.8 Condividi i tuoi plugin con gli altri

Se vuoi condividere i plugin che hai creato con altri utenti di Calibre, posta il tuo plugin in un nuovo thread nel [calibre plugins forum](#)¹⁰⁰.

10.6 Comporre formule matematiche negli e-book

Il Lettore di E-book di calibre è in grado di mostrare le formule matematiche contenute nei libri (in formato EPUB e HTML). Puoi comporre le formule direttamente con TeX, MathML o AsciiMath, il lettore di e-book di calibre usa l'eccellente libreria *MathJax* <<https://www.mathjax.org>> _ per mostrarle. Questa è una breve guida sulla creazione di e-book contenenti formule matematiche compatibili con il lettore di e-book di calibre.

10.6.1 Un semplice file HTML con le formule matematiche

Puoi scrivere le espressioni dentro ad un semplice file HTML, il lettore di Calibre le convertirà nelle formule corrette. Potrai usare i normali comandi Tex ma con un piccolo accorgimento: la «e commerciale» il «maggiore di» ed il «meno di» devono essere scritte rispettivamente con: &, < e >.

Il primo passo è quello di dire a calibre che il file conterrà delle formule matematiche, inserendo il seguente frammento di codice alla sezione <head> del file HTML:

```
<script type="text/x-mathjax-config"></script>
```

Tutto qua, ora puoi digitare le formule come vuoi in un file .tex. Eco, ad esempio, l'equazione di Lorentz:

```
<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} &= \sigma(y-x) \\
\dot{y} &= \rho x - y - xz \\
\dot{z} &= -\beta z + xy
\end{align}
</p>
```

Ecco come appare il frammento in una schermata del lettore di e-book di calibre.

Qui sotto è riportato il file HTML completo, con più equazioni ed espressioni matematiche. Puoi convertire il file HTML in EPUB con Calibre per concludere con un e-book da distribuire agli altri.

```
<!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" />
```

(continues on next page)

⁹⁹ <https://www.mobileread.com/forums/showthread.php?t=118764>

¹⁰⁰ <https://www.mobileread.com/forums/forumdisplay.php?f=237>

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

Fig. 1: Equazione di Lorentz

(continua dalla pagina precedente)

```

<!-- This script tag is needed to make calibre's ebook-viewer recognize that this_
↪file needs math typesetting -->
<script type="text/x-mathjax-config">
    // This line adds numbers to all equations automatically, unless explicitly_
↪suppressed.
    MathJax.tex = {tags: 'all'};
</script>

<style>
h1 {text-align:center}
h2 {
    font-weight: bold;
    background-color: #DDDDDD;
    padding: .2em .5em;
    margin-top: 1.5em;
    border-top: 3px solid #666666;
    border-bottom: 2px solid #999999;
}
</style>
</head>
<body>

<h1>Sample Equations</h1>

<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} \text{ \& } &= \sigma(y-x) \text{ \label{lorenz}} \\
\dot{y} \text{ \& } &= \rho x - y - xz \\
\dot{z} \text{ \& } &= -\beta z + xy
\end{align}
</p>

<h2>The Cauchy-Schwarz Inequality</h2>

<p>\[
\left( \sum_{k=1}^n a_k b_k \right)^{\{\!\!\!2\}} \leq
\left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
\]</p>

<h2>A Cross Product Formula</h2>

```

(continues on next page)

(continua dalla pagina precedente)

```

<p>\[
  \mathbf{V}_1 \times \mathbf{V}_2 =
  \begin{vmatrix}
    \mathbf{i} & \mathbf{j} & \mathbf{k} \\
    \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\
    \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0
  \end{vmatrix}
\]
```

The probability of getting k heads when flipping n coins is:

```

<p>\[P(E) = \sum_{k=0}^n \binom{n}{k} p^k (1-p)^{n-k} \]
```

An Identity of Ramanujan

```

<p>\[
  \frac{1}{1 + \frac{e^{-2\pi}}{1 + \frac{e^{-4\pi}}{1 + \frac{e^{-6\pi}}{1 + \frac{e^{-8\pi}}{\ddots}}}}} =
\]
```

A Rogers-Ramanujan Identity

```

<p>\[
  1 + \frac{q^2}{(1-q)} + \frac{q^6}{(1-q)(1-q^2)} + \cdots =
  \prod_{j=0}^{\infty} \frac{1}{(1-q^{5j+2})(1-q^{5j+3})},
  \quad \text{for } |q| < 1.
\]
```

Maxwell's Equations

```

<p>
\begin{align}
  \nabla \times \vec{\mathbf{B}} &= \frac{1}{c} \frac{\partial \vec{\mathbf{E}}}{\partial t} \\
  \nabla \cdot \vec{\mathbf{E}} &= 4\pi \rho \\
  \nabla \times \vec{\mathbf{E}} &= -\frac{1}{c} \frac{\partial \vec{\mathbf{B}}}{\partial t} \\
  \nabla \cdot \vec{\mathbf{B}} &= 0
\end{align}
</p>
<h2>In-line Mathematics</h2>
<p>While display equations look good for a page of samples, the
ability to mix math and text in a paragraph is also important. This
expression  $\sqrt{3x-1} + (1+x)^2$  is an example of an inline equation. As
you see, equations can be used this way as well, without unduly
disturbing the spacing between lines.
</p>
<h2>References to equations</h2>
<p>Here is a reference to the Lorenz Equations (\ref{lorenz}). Clicking on the
equation number will take you back to the equation.
</p>
</body>

```

(continues on next page)

</html>

10.6.2 Ulteriori informazioni

Dato che il lettore di e-book di calibre usa la libreria MathJax per mostrare le formule matematiche, il miglior luogo per approfondire l'argomento delle formule matematiche negli e-book e per ottenere aiuto è il [sito web di MathJax](https://www.mathjax.org)¹⁰¹.

10.7 Creazione di cataloghi AZW3 • EPUB • MOBI

L'opzione Crea catalogo di calibre ti permette di creare un catalogo della tua libreria in una serie di formati diversi. Questo file di aiuto descrive le opzioni di catalogazione quando si genera un catalogo in formato AZW3, EPUB e MOBI.

- *Selezionare i libri da catalogare* (pagina 234)
- *Sezioni incluse* (pagina 235)
- *Prefissi* (pagina 236)
- *Libri esclusi* (pagina 236)
- *Generi esclusi* (pagina 237)
- *Altre opzioni* (pagina 237)
- *Copertine di catalogo personalizzate* (pagina 238)
- *Risorse aggiuntive di aiuto* (pagina 238)

10.7.1 Selezionare i libri da catalogare

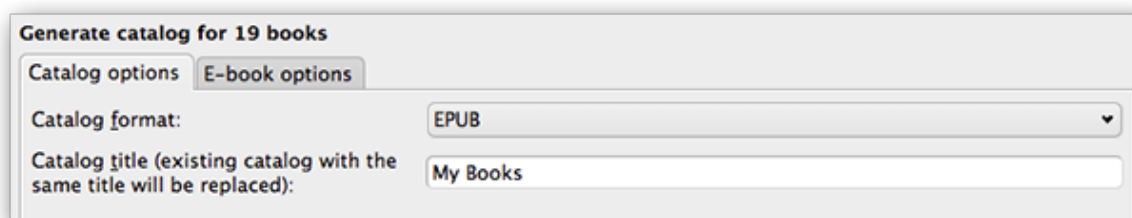
Se vuoi che *tutta* la tua biblioteca sia catalogata, rimuovi ogni criterio di ricerca o di filtraggio nella schermata principale. Con un singolo libro selezionato, tutti i libri nella tua biblioteca saranno candidati per l'inclusione nel catalogo generato. Singoli libri possono essere esclusi con vari criteri; vedi la sezione sottostante *Generi esclusi* (pagina 237) per maggiori informazioni.

Se si desidera che solo *alcuni* libri siano catalogati, esistono due opzioni:

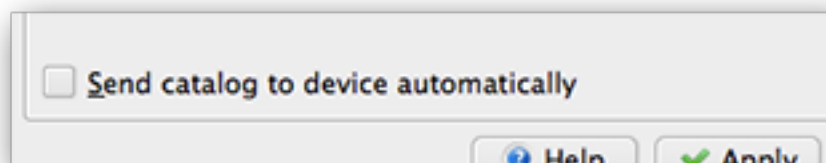
- Crea una selezione multipla dei libri che vuoi catalogare. Con più di un libro selezionato nella schermata principale di calibre, solo i libri selezionati saranno catalogati.
- Usa il campo Cerca o il navigatore di tag per filtrare i libri visualizzati. Solo i libri visualizzati saranno catalogati.

Per iniziare la creazione del catalogo, seleziona la voce del menu *Converti libri > Crea un catalogo di libri nella libreria di calibre*. È possibile anche aggiungere un pulsante *Crea catalogo* ad una barra degli strumenti in *Preferenze > Interfaccia > Barre degli strumenti e menu* per un accesso più facile alla finestra Genera catalogo.

¹⁰¹ <https://www.mathjax.org>

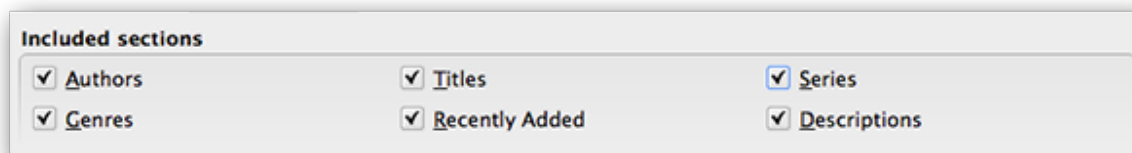


In *Opzioni catalogo*, seleziona **AZW3**, **EPUB** o **MOBI** come formato del catalogo. Nel campo *Titolo catalogo*, fornisci un nome che verrà usato per il catalogo generato. Se il catalogo con lo stesso nome e formato esiste già, sarà rimpiazzato dal nuovo catalogo generato.



Selezionare *Invia il catalogo al dispositivo automaticamente* scaricherà automaticamente il catalogo generato ad un dispositivo connesso al completamento.

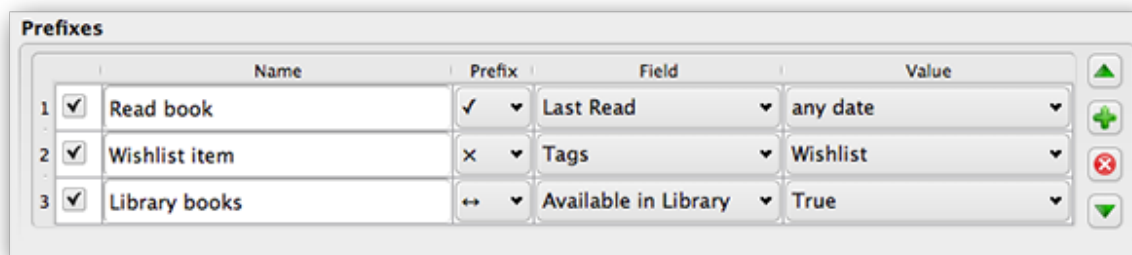
10.7.2 Sezioni incluse



Le sezioni abilitate tramite segno di spunta verranno inserite nel catalogo generato:

- *Authors* - tutti i libri, ordinati per autore, presentati in formato elenco. Libri non in serie vengono elencati prima dei libri in serie.
- *Titles* - Tutti i libri, ordinati per titolo, presentati nel formato elenco.
- *Serie* - tutti i tuoi libri che sono parte di una serie, ordinati per serie, presentati in un elenco.
- *Generi* - i generi individuali presentati in un elenco ordinati per Autore e Serie.
- *Aggiunti di recente* - tutti i libri, ordinati in ordine cronologico inverso. L'elenco include i libri aggiunti nei passati 30 giorni, poi elencando mese per mese l'elenco dei libri aggiunti.
- *Descrizioni* - pagina di descrizione dettagliata per ogni libro, inclusi la copertina ed i commenti. Ordinati per autore, con i libri non appartenenti a delle serie elencati prima di quelli appartenenti a delle serie.

10.7.3 Prefissi



Le regole di prefisso ti permettono di aggiungere un prefisso agli elenchi dei libri se certi criteri sono soddisfatti. Per esempio potresti voler segnare i libri che hai letto con una spunta, oppure i libri nella tua lista dei desideri con una X.

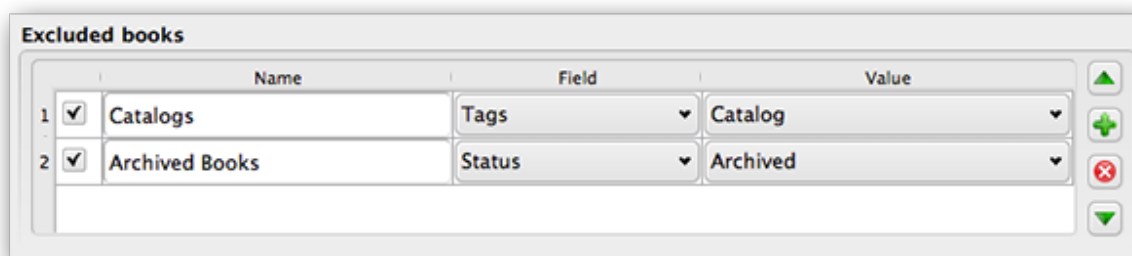
La casella nella prima colonna permette la regola. :guilabel:"Nome" è il nome di una regola che fornisci. :guilabel:"Campo" è :guilabel:"Tags" oppure una colonna personalizzata dalla tua libreria.

Nell'esempio precedente vengono specificate tre regole prefisso:

1. :guilabel:"Leggi libro" specifica che un libro con qualche data in una colonna personalizzata chiamata :guilabel:"Ultimo letto" sarà prefissato con il simbolo di una spunta.
2. La voce :guilabel:"Lista desideri" specifica che ogni libro con un tag *Wishlist* sarà prefissato con il simbolo di una X.
3. Il :guilabel:"Catalogo" di libri specifica che ogni libro con un valore di Vero (o Sì) in una colonna personalizzata :guilabel:"Disponibile in catalogo" sarà prefissato con il simbolo di una doppia freccia.

La prima regola che corrisponde fornisce il prefisso. Regole inattive o incomplete vengono ignorate.

10.7.4 Libri esclusi



Le regole di esclusione consentono di specificare quali libri non saranno catalogati.

La casella nella prima colonna permette la regola. :guilabel:"Nome" è il nome di una regola che fornisci. :guilabel:"Campo" è :guilabel:"Tags" oppure una colonna personalizzata dalla tua libreria. :guilabel:"Valore" è il contenuto che corrisponde a :guilabel:"Campo". Quando una regola di esclusione è soddisfatta, il libro viene escluso dal catalogo generato.

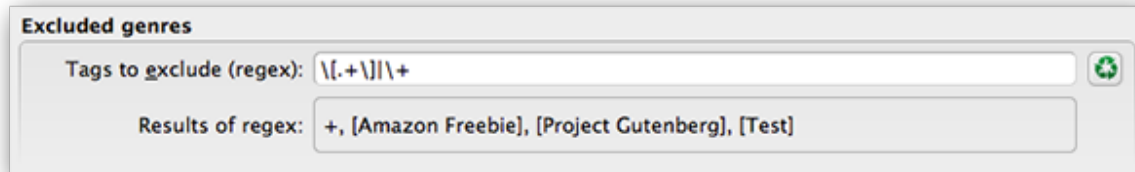
Due regole di esclusione sono state specificate nell'esempio di cui sopra:

1. La regola :guilabel:"Cataloghi" specifica che ogni libro con un tag :guilabel:"Catalogo" sarà escluso dal catalogo generato.

2. La regola dei Libri :guilabel:"Archiviati" specifica che ogni libro con un valore :guilabel:"Archiviato" nella colonna personalizzata :guilabel:"Stato" sarà escluso dal catalogo generato.

Tutte le regole vengono valutate per ogni libro. Regole inattive o incomplete vengono ignorate.

10.7.5 Generi esclusi



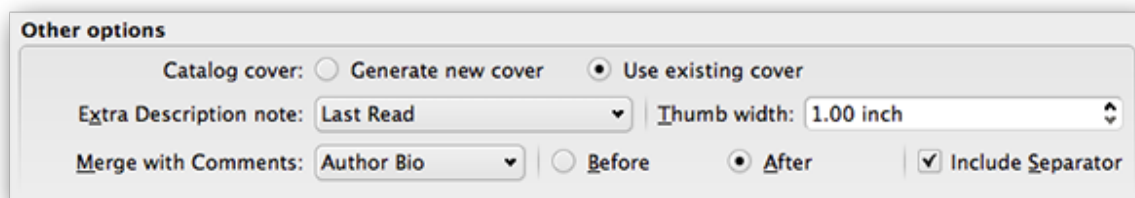
Quando il catalogo viene generato, i tags nel tuo database vengono usati come generi. Per esempio, puoi usare i tags «Narrativa» e «Saggistica». Questi tags diventano generi nel catalogo generato, con i libri elencati sotto la loro rispettive liste di genere basate sui loro tags assegnati. Un libro sarà elencato in ogni sezione di genere per cui ha un tag corrispondente.

Puoi usare alcuni tags per altri scopi, magari un + per indicare un libro letto oppure un tag fra parentesi come «[Amazon Freebie]» per indicare la fonte di un libro. Un'espressione regolare come :guilabel:"Generi esclusi" ti permette di specificare i tags che non vuoi usare come generi nel catalogo generato. Il modello predefinito dell'espressione regolare di esclusione «[.+]» esclude ogni tags dal modulo «[tag]», così come esclude «+», il tag predefinito per leggere libri, dall'essere usato come i generi nel catalogo generato.

Puoi anche usare un il nome di un tag preciso in un'espressione regolare. Per esempio, «[Amazon Freebie]» oppure «[Project Gutenberg]». Se vuoi elencare molteplici tag esatti per l'esclusione, metti un pipe (barra verticale) fra di essi «[AmazonFreebie]|[ProjectGutenberg]».

:guilabel:"Risultati espressione regolare" ti mostrano quali tags saranno esclusi quando il catalogo sarà fatto, basato sui tags nel tuo database e sul modello dell'espressione che inserisci. I risultati vengono aggiornati appena modifichi il modello dell'espressione.

10.7.6 Altre opzioni



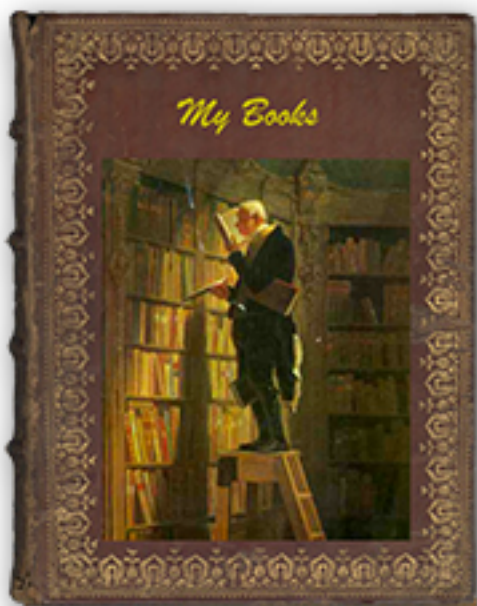
Copertina del catalogo Specifica se generare una nuova copertina o se usarne una esistente. È possibile creare una copertina personalizzata per i tuoi cataloghi - vedi *Copertine di catalogo personalizzate* (pagina 238) per maggiori informazioni. Se hai creato una copertina personalizzata che desideri riutilizzare, seleziona *Usa copertina esistente*, altrimenti seleziona *Genera nuova copertina*.

Nota descrizioni extra specifica i contenuti di una colonna personalizzata da inserire nella pagina di Descrizione, vicino all'anteprima della cover. Per esempio, puoi voler mostrare la data dell'ultima volta che hai letto un libro usando una colonna personalizzata :guilabel:"Ultima lettura". Per un uso avanzato della caratteristica della Nota descrittiva, vedi questo post nel forum calibre <<http://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395>>`_.

Sfoggia ampiezza specifica la preferenza di un'ampiezza per l'anteprima della cover, inclusa nelle pagine Descrittive. Le anteprime sono salvate nella cache per migliorare le performance. Per sperimentare con differenti ampiezze, prova generando un catalogo con soltanto pochi libri finché non hai deciso la tua ampiezza preferita, poi genera il tuo intero catalogo. La prima volta che un catalogo è generato con una nuova ampiezza di anteprima, la prestazione sarà più lenta, ma le strutture successive del catalogo saranno avvantaggiate dalla cache dell'anteprima.

Merge with Comments specifica una colonna personalizzata il cui contenuto sarà unito, in modo non distruttivo, con i metadata Commenti durante la creazione del catalogo. Ad esempio, si può avere una colonna personalizzata *Bio autore* che si vuole accodare ai metadata Commenti. Si può scegliere di inserire una colonna personalizzata «prima o dopo» la sezione commenti e separare (facoltativo) il contenuto accodato con un separatore orizzontale. I tipi di colonna personalizzati ammessi includono «testo, commenti e composito».

10.7.7 Copertine di catalogo personalizzate



Con il [Generatore di Cover plugin](#)¹⁰² installato, puoi creare cover personalizzate per il tuo catalogo. Per installare il plugin, vai su *Preferenze > Avanzate > Plugin > Ottieni nuovi plugin*.

10.7.8 Risorse aggiuntive di aiuto

Per ulteriori informazioni sulla funzionalità del Catalogo di calibre, consulta la discussione in evidenza sul forum MobileRead [Creating Catalogs - Start here](#)¹⁰³, dove sarà possibile trovare informazioni su come personalizzare i modelli di catalogo, e come segnalare un bug.

Per chiedere qualcosa o discutere sulle funzioni del Catalogo di calibre con altri utenti, visita il forum di MobileRead [Library Management](#)¹⁰⁴.

¹⁰² <https://www.mobileread.com/forums/showthread.php?t=124219>

¹⁰³ <https://www.mobileread.com/forums/showthread.php?t=118556>

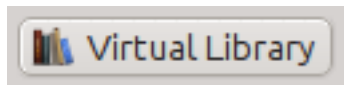
¹⁰⁴ <https://www.mobileread.com/forums/forumdisplay.php?f=236>

10.8 Biblioteche Virtuali

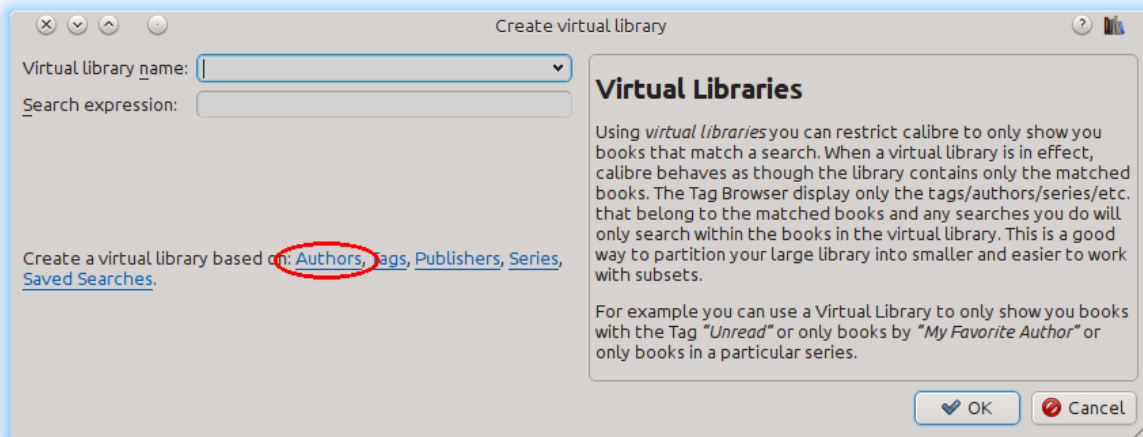
In Calibre, una Biblioteca Virtuale è un modo per dire a calibre di aprire solo un sottoinsieme di una biblioteca normale. Ad esempio, potresti voler lavorare solo con i libri di un determinato autore o con libri che hanno solo un determinato tag. L'uso delle Biblioteche Virtuali è il modo migliore di dividere la tua raccolta di libri di grandi dimensioni in raccolte secondarie più piccole. È superiore alla suddivisione della biblioteca in più biblioteche più piccole poiché, quando si desidera effettuare ricerche nell'intera raccolta, si può semplicemente tornare alla biblioteca completa. Non è possibile effettuare ricerche contemporanee in librerie diverse in calibre.

Una libreria Virtuale è diversa da una semplice ricerca. Una ricerca restringerà solo l'elenco dei libri visualizzati nell'elenco dei libri. Una libreria Virtuale lo fa e inoltre limita anche le voci mostrate nel :guilabel: `Visualizzatore di Etichette` a sinistra. Il Visualizzatore di Etichette mostrerà solo tag, autori, serie, editori, ecc. che provengono dai libri nella biblioteca Virtuale. Una libreria Virtuale si comporta quindi come se la libreria effettiva contenesse solo l'insieme di libri da essa delimitato.

10.8.1 Creazione di biblioteche virtuali



Per usare una biblioteca Virtuale fare clic sul pulsante *Biblioteca Virtuale* situato a sinistra della barra di ricerca e selezionare l'opzione *Crea biblioteca Virtuale*. Come primo esempio, creiamo una biblioteca Virtuale che ci mostra solo i libri di un determinato autore. Fare clic sul collegamento *Autori* come mostrato nell'immagine seguente e selezionare l'autore che si desidera utilizzare e fare clic su OK.



La finestra di dialogo Crea libreria Virtuale è stata compilata per te. Fai clic su OK e vedrai che è stata creata una nuova libreria Virtuale, a cui si è passati automaticamente, che mostra solo i libri dell'autore selezionato. Per come è pensato Calibre è come se la tua biblioteca contenga solo i libri dell'autore selezionato.

Puoi tornare alla libreria completa in qualsiasi momento facendo nuovamente clic su *Biblioteca Virtuale* e selezionando la voce denominata <None>.

Le librerie Virtuali si basano su *ricerche*. È possibile utilizzare qualsiasi ricerca come base di una libreria Virtuale. La biblioteca Virtuale conterrà solo i libri corrispondenti a quella ricerca. Innanzitutto, digitare la chiave di ricerca che si desidera utilizzare nella barra di ricerca o creare una ricerca usando *Visualizzatore Etichette*. Quando sei soddisfatto dei risultati restituiti, fai clic sul pulsante *Biblioteca virtuale*, scegli *Crea biblioteca* ed inserisci un nome per la nuova biblioteca Virtuale. La libreria Virtuale verrà quindi creata in base alla ricerca appena digitata. Le ricerche sono molto potenti, per esempio dei tipi di cose che puoi fare con loro, vedi *L'interfaccia di ricerca* (pagina 12).

Esempi di biblioteche virtuali utili

- **Libri aggiunti a calibre nell'ultimo giorno:**
data:>1giornifa
- **Libri aggiunti a calibre nell'ultimo mese:**
data:>30giornifa
- **Libri con valutazione di 5 stelle:**
valutazione:5
- **Libri con valutazione di 4 stelle:**
valutazione:>=4
- **Libri senza valutazione:**
valutazione:falso
- **Periodici scaricati dalla funzione Fetch News in Calibre**
tags:=News ed autore:=calibre
- **Libri senza tag:**
tags:false
- **Libri senza copertina:**
copertina:falso

10.8.2 Lavorare con le biblioteche virtuali

È possibile modificare una libreria Virtuale precedentemente creata o rimuoverla, facendo clic su *Biblioteca virtuale* e scegliendo l'azione appropriata.

Puoi dire a Calibre che vuoi sempre applicare una particolare libreria Virtuale quando viene aperta la libreria corrente, andando su *Preferenze* → *Interfaccia* → *Comportamento*.

Puoi usare rapidamente la ricerca corrente come libreria Virtuale temporanea facendo clic sul pulsante *Biblioteca Virtuale* e selezionando la voce **ricerca corrente*.

È possibile visualizzare tutte le Biblioteche Virtuali disponibili come schede sopra l'elenco dei libri. Ciò è particolarmente utile se ti piace passare da una Biblioteca Virtuale all'altra molto spesso. Fai clic sul pulsante *Biblioteca Virtuale* e seleziona *Mostra Biblioteche Virtuali come schede*. Puoi riorganizzare le schede trascinandole e chiudendo quelle che non vuoi vedere. Le schede chiuse possono essere ripristinate facendo clic con il tasto destro sulla barra delle schede.

10.8.3 Usare le biblioteche virtuali nelle ricerche

Puoi cercare libri che si trovano in una biblioteca Virtuale usando il prefisso `vl:`. Ad esempio, `vl:Letti` troverà tutti i libri nella libreria virtuale *Letti*. La ricerca ```vl:Letti e vl:>Science Fiction``` troverà tutti i libri che si trovano nelle librerie Virtuali *Letti* e *Science Fiction*.

Il valore seguente `vl:` deve essere il nome di una libreria Virtuale. Se il nome della libreria Virtuale contiene spazi, racchiudilo tra virgolette.

Uno degli usi di una ricerca nelle biblioteche virtuali è con il server dei contenuti. In *Preferenze* → *Condivisione via rete* → *Richiedi nome utente e password* puoi limitare le biblioteche di calibre visibili ad un utente. Per ogni biblioteca visibile puoi anche specificare un'espressione di ricerca che limiti ulteriormente quali libri vengono visualizzati. Usa `vl:"Nome della biblioteca virtuale"` per limitare i libri a quelli parte di una biblioteca virtuale.

10.8.4 Utilizzare restrizioni aggiuntive

Puoi limitare ulteriormente i libri mostrati in una biblioteca Virtuale usando *Restrizioni aggiuntive*. Un'ulteriore limitazione è la ricerca salvata precedentemente creata che può essere applicata alla libreria Virtuale corrente per limitare ulteriormente i libri mostrati in una libreria Virtuale. Ad esempio, supponiamo che tu abbia una biblioteca virtuale per i libri taggata come *Historical Fiction* e una ricerca salvata che ti mostra libri non letti, puoi fare clic sul pulsante *Virtual Library* e scegliere l'opzione :guilabel: *Additional restriction* per mostrare solo libri di narrativa storica non letti. Per informazioni sulle ricerche salvate, consultare [Salvare le ricerche](#) (pagina 18).

Lo schema degli URL calibre://

calibre si registra come programma di gestione per gli URL calibre://. Quindi puoi usarli per eseguire azioni come aprire libri, cercare libri, ecc. Da altri programmi/documenti o tramite la riga di comando. Ad esempio, eseguendo quanto segue dalla riga di comando:

```
calibre calibre://switch-library/Some_Library
```

Si aprirà calibre con la biblioteca denominata `Una biblioteca`. I nomi delle biblioteche sono il nome della cartella in cui si trova la biblioteca con gli spazi sostituiti da trattini bassi. Il valore speciale `_` indica la biblioteca corrente. I vari tipi di URL sono documentati di seguito.

Puoi persino inserire questi collegamenti all'interno di file HTML o documenti Word o simili ed il sistema operativo avvierà automaticamente calibre per eseguire l'azione specificata.

- *Passa ad una biblioteca specifica* (pagina 244)
- *Mostra un libro specifico in calibre* (pagina 244)
- *Apri un libro specifico nel Lettore e-book a una posizione specifica* (pagina 244)
- *Ricerca di libri* (pagina 245)
- *Apri una finestra con i dettagli del libro per un libro di una data biblioteca* (pagina 245)
- *Codifica esadecimale dei parametri dell'URL* (pagina 245)

11.1 Passa ad una biblioteca specifica

La sintassi dell'URL è:

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

I nomi delle biblioteche sono il nome della cartella della biblioteca con spazi sostituiti da trattini bassi. Il valore speciale `_` indica la biblioteca corrente. Si può anche usare la *codifica esadecimale* (pagina 245) per i nomi delle biblioteche, utile se i nomi delle biblioteche hanno dei caratteri speciali che possono richiama la codifica dell'URL. Le biblioteche con codifica esadecimale sono visualizzate come:

```
_hex_-AD23F4BC
```

Dove la parte dopo il prefisso `_hex_` è il nome della biblioteca codificato come UTF-8 ed ogni byte è rappresentato da due caratteri esadecimali.

11.2 Mostra un libro specifico in calibre

La sintassi dell'URL è:

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

Questo mostrerà il libro con `book_id` (un numero) in calibre. Gli id dei libri possono essere visualizzati nell'interfaccia di calibre passando con il mouse sul collegamento *Fai clic per aprire* nel pannello *Dettagli libro*, è il numero tra parentesi alla fine del percorso della cartella del libro.

Puoi copiare un collegamento al libro attualmente visualizzato in calibre facendo clic con il tasto destro sul pannello *Dettagli libro* e scegliendo *Copia collegamento al libro*.

Se c'è una ricerca attiva e il libro in questione non fa parte dei suoi risultati, la ricerca viene rimossa.

Se è selezionata una biblioteca virtuale, calibre la userà nel mostrare il libro. Se però il libro in questione non viene trovato in quella biblioteca virtuale, la biblioteca virtuale viene chiusa.

Se vuoi passare a una particolare biblioteca virtuale quando mostri il libro, usa:

```
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
```

sostituendo gli spazi nel nome della biblioteca virtuale con `%20`. Se il libro non viene trovato in quella biblioteca virtuale, la biblioteca virtuale viene ignorata.

11.3 Apri un libro specifico nel Lettore e-book a una posizione specifica

La sintassi dell'URL è:

```
calibre://view-book/Library_Name/book_id/book_format?open_at=location
```


Qui, `book_format` è il formato del libro, ad esempio, `EPUB` o `MOBI` e la `posizione` è una posizione opzionale all'interno del libro. Il modo più semplice per ottenere questi collegamenti è aprire un libro nel visualizzatore, quindi nei controlli del visualizzatore selezionare *Vai a → Posizione* e verrà fornito un collegamento che è possibile copiare/incollare altrove.

11.4 Ricerca di libri

La sintassi dell'URL è:

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

Qui la query è una qualunque *espressione di ricerca* (pagina 12) valida. Se l'espressione di ricerca è complessa, *codificala come una stringa esadecimale* (pagina 245) e usa `eq` al suo posto. Tralasciando la query, la ricerca corrente verrà cancellata.

Per impostazione predefinita, se una Biblioteca virtuale è selezionata, calibre la deselectonerà prima di effettuare la ricerca per assicurare che tutti i libri vengano trovati. Se vuoi mantenere la Biblioteca virtuale, usa:

```
calibre://search/Library_Name?q=query&virtual_library=_
```

Se vuoi passare a una specifica Biblioteca virtuale, usa:

```
calibre://search/Library_Name?virtual_library=Library%20Name
or
calibre://search/Library_Name?encoded_virtual_library=hex_encoded_virtual_library_name
```

sostituendo gli spazi nel nome della Biblioteca virtuale con `%20`.

Se effettui una ricerca in calibre e desideri generare un collegamento che la richiami, puoi farlo facendo clic con il tasto destro nella barra di ricerca e scegliendo *Copia ricerca come URL*.

11.5 Apri una finestra con i dettagli del libro per un libro di una data biblioteca

La sintassi dell'URL è:

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

In questo modo si apre una finestra con i dettagli del libro per il libro e la biblioteca specificati, senza cambiare la biblioteca corrente o il libro selezionato.

11.6 Codifica esadecimale dei parametri dell'URL

La codifica esadecimale dei parametri URL viene eseguita codificando prima il parametro come byte UTF-8, quindi sostituendo ogni byte con due caratteri esadecimali che rappresentano il byte. Ad esempio, la stringa `abc` è i byte `0x61 0x62 e 0x63` in UTF-8, quindi la versione codificata è la stringa `616263`.

Personalizzare calibre

calibre ha una struttura estremamente modulare. Molte delle sue parti possono essere personalizzate. Qui imparerai:

- come usare le variabili d'ambiente e le *ottimizzazioni* per personalizzare il comportamento di calibre,
- come impostare le tue personali risorse statiche, come icone e modelli, per sostituire quelle predefinite
- come usare i *plugin* per aggiungere funzionalità a calibre.
- come condividere temi di icone e plugin con gli altri utenti di calibre.
- per scoprire come creare le *ricette* per aggiungere nuove fonti di contenuto online a calibre, visita la Sezione *Aggiunta del tuo sito di notizie preferito* (pagina 29).

Nota: Tieni presente che sebbene i temi di icone e i plugin siano indicizzati e scaricabili dal gestore degli aggiornamenti incluso in calibre essi non ne sono parte integrante, e i luoghi in cui cercare supporto e codice sorgente sono i forum di Mobileread <<https://www.mobileread.com/forums/forumdisplay.php?f=166>> nei relativi thread di supporto.

- *Variabili d'ambiente* (pagina 280)
- *Ottimizzazioni* (pagina 281)
- *Sovrascrivere icone, modelli, eccetera* (pagina 292)
- *Creazione di un tema personale per le icone di calibre* (pagina 292)
- *Personalizzazione di calibre con i plugin* (pagina 293)

12.1 Documentazione API per i plugin

Definisce diverse classi astratte di base che possono essere utilizzate con sottoclassi per creare potenti plugin. Le classi utili sono:

- *Plugin* (pagina 248)
- *FileTypePlugin* (pagina 250)
- *Metadata plugins* (pagina 252)
- *Catalog plugins* (pagina 253)
- *Metadata download plugins* (pagina 254)
- *Conversion plugins* (pagina 258)
- *Device drivers* (pagina 260)
- *User interface actions* (pagina 274)
- *Preferences plugins* (pagina 278)

12.1.1 Plugin

class calibre.customize.**Plugin** (*plugin_path*)

Un plugin calibre. Membri utili:

- `self.installation_type`: conserva la modalità di installazione del plugin.
- **`self.plugin_path`**: Stores path to the ZIP file that contains questo plugin o Nessuno se è un plugin interno
- **`self.site_customization`**: Stores a customization string entered dall'utente.

Methods that should be overridden in sub classes:

- `initialize()` (pagina 249)
- `customization_help()` (pagina 250)

Metodi utili:

- `temporary_file()` (pagina 250)
- `__enter__()`
- `load_resources()` (pagina 249)

supported_platforms = []

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: ['windows', 'osx', 'linux']

name = 'Trivial Plugin'

The name of this plugin. You must set it something other than Trivial Plugin for it to work.

version = (1, 0, 0)

The version of this plugin as a 3-tuple (major, minor, revision)

description = 'Non fa assolutamente niente'

Una breve stringa che descrive cosa fa questo plugin

author = 'Sconosciuto'

L'autore di questo plugin

priority = 1

When more than one plugin exists for a filetype, the plugins are run in order of decreasing priority. Plugins with higher priority will be run first. The highest possible priority is `sys.maxsize`. Default priority is 1.

minimum_calibre_version = (0, 4, 118)

La versione piu' recente di calibre che questo plugin richiede

installation_type = None

La modalità di installazione di questo plugin

can_be_disabled = True

If False, the user will not be able to disable this plugin. Use with care.

type = 'Base'

The type of this plugin. Used for categorizing plugins in the GUI

initialize()

Called once when calibre plugins are initialized. Plugins are re-initialized every time a new plugin is added. Also note that if the plugin is run in a worker process, such as for adding books, then the plugin will be initialized for every new worker process.

Perform any plugin specific initialization here, such as extracting resources from the plugin ZIP file. The path to the ZIP file is available as `self.plugin_path`.

Note that `self.site_customization` is **not** available at this point.

config_widget()

Implement this method and `save_settings()` (pagina 249) in your plugin to use a custom configuration dialog, rather than relying on the simple string based default customization.

This method, if implemented, must return a QWidget. The widget can have an optional method `validate()` that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns True.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

save_settings(config_widget)

Save the settings specified by the user with `config_widget`.

Parametri

config_widget – The widget returned by `config_widget()` (pagina 249).

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.

For example to load an image:

```

pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png'])['images/icon.png
↪'])
icon = QIcon(pixmap)

```

Parametri

names – List of paths to resources in the ZIP file using / as separator

Ritorna

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.

Parametri

gui – If True return HTML help, otherwise return plain text help.

temporary_file (suffix)

Return a file-like object that is a temporary file on the file system. This file will remain available even after being closed and will only be removed on interpreter shutdown. Use the `name` member of the returned object to access the full path to the created temporary file.

Parametri

suffix – Il suffisso che avrà il file temporaneo.

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)

Basi: *Plugin* (pagina 248)

A plugin that is associated with a particular set of file types.

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

If True, this plugin is run when books are added to the database

on_postimport = False

If True, this plugin is run after books are added to the database. In this case the `postimport` and `postadd` methods of the plugin are called.

on_postconvert = False

If True, this plugin is run after a book is converted. In this case the postconvert method of the plugin is called.

on_postdelete = False

If True, this plugin is run after a book file is deleted from the database. In this case the postdelete method of the plugin is called.

on_preprocess = False

If True, this plugin is run just before a conversion

on_postprocess = False

If True, this plugin is run after conversion on the final file produced by the conversion output plugin.

type = 'Tipo di file'

The type of this plugin. Used for categorizing plugins in the GUI

run (*path_to_ebook*)

Run the plugin. Must be implemented in subclasses. It should perform whatever modifications are required on the e-book and return the absolute path to the modified e-book. If no modifications are needed, it should return the path to the original e-book. If an error is encountered it should raise an Exception. The default implementation simply return the path to the original e-book. Note that the path to the original file (before any file type plugins are run, is available as `self.original_path_to_file`).

The modified e-book file should be created with the `temporary_file()` method.

Parametri

path_to_ebook – Absolute path to the e-book.

Ritorna

Absolute path to the modified e-book.

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()` (pagina 252) 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.

Parametri

- **book_id** – Database id of the added book.
- **book_format** – Il tipo di file del libro che è stato aggiunto.
- **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.

Parametri

- **book_id** – Database id of the added book.
- **book_format** – Il tipo di file del libro che è stato aggiunto.
- **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.

Parametri

- **book_id** – Database id of the added book.
- **book_format** – Il tipo di file del libro che è stato aggiunto.
- **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()* (pagina 251), 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.

Parametri

- **book_id** – Database id of the added book.
- **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** – Library database

12.1.3 Metadata plugins

class calibre.customize.**MetadataReaderPlugin** (*args, **kwargs)

Basi: *Plugin* (pagina 248)

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']

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: ['windows', 'osx', 'linux']

version = (6, 27, 0)

The version of this plugin as a 3-tuple (major, minor, revision)

author = 'Kovid Goyal'

L'autore di questo plugin

type = 'Lettore metadati'

The type of this plugin. Used for categorizing plugins in the GUI

get_metadata (*stream, type*)

Return metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data.

Parametri

type – The type of file. Guaranteed to be one of the entries in *file_types* (pagina 252).

Ritorna

A *calibre.ebooks.metadata.book.Metadata* object


```
class calibre.customize.MetadataWriterPlugin(*args, **kwargs)
```

Basi: [Plugin](#) (pagina 248)

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']
```

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: `['windows', 'osx', 'linux']`

```
version = (6, 27, 0)
```

The version of this plugin as a 3-tuple (major, minor, revision)

```
author = 'Kovid Goyal'
```

L'autore di questo plugin

```
type = 'Scrittore metadati'
```

The type of this plugin. Used for categorizing plugins in the GUI

```
set_metadata(stream, mi, type)
```

Set metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data.

Parametri

- **type** – The type of file. Guaranteed to be one of the entries in [file_types](#) (pagina 253).
- **mi** – A `calibre.ebooks.metadata.book.Metadata` object

12.1.4 Catalog plugins

```
class calibre.customize.CatalogPlugin(plugin_path)
```

Basi: [Plugin](#) (pagina 248)

A plugin that implements a catalog generator.

```
file_types = {}
```

Output file type for which this plugin should be run. For example: “epub” or “xml”

```
type = 'Generatore di catalogo'
```

The type of this plugin. Used for categorizing plugins in the GUI

```
cli_options = []
```

CLI parser options specific to this plugin, declared as *namedtuple Option*:

```
from collections import namedtuple
Option = namedtuple("Option", "option, default, dest, help")
cli_options = [Option("--catalog-title", default = "My Catalog", dest = "catalog_title", help = _("Title of generated catalog. nDefault:")) + « "» + "%default" + «"»))]
cli_options parsed in calibre.db.cli.cmd_catalog.option_parser()
```

```
initialize()
```

If plugin is not a built-in, copy the plugin's .ui and .py files from the ZIP file to \$TMPDIR. Tab will be dynamically generated and added to the Catalog Options dialog in `calibre.gui2.dialogs.catalog.py:Catalog`

run (*path_to_output*, *opts*, *db*, *ids*, *notification=None*)

Run the plugin. Must be implemented in subclasses. It should generate the catalog in the format specified in *file_types*, returning the absolute path to the generated catalog file. If an error is encountered it should raise an Exception.

The generated catalog file should be created with the `temporary_file()` method.

Parametri

- **path_to_output** – Absolute path to the generated catalog file.
- **opts** – A dictionary of keyword arguments
- **db** – A `LibraryDatabase2` object

12.1.5 Metadata download plugins

class `calibre.ebooks.metadata.sources.base.Source` (**args*, ***kwargs*)

Basi: *Plugin* (pagina 248)

type = `'Sorgente metadati'`

The type of this plugin. Used for categorizing plugins in the GUI

author = `'Kovid Goyal'`

L'autore di questo plugin

supported_platforms = `['windows', 'osx', 'linux']`

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: `['windows', 'osx', 'linux']`

capabilities = `frozenset({})`

Set of capabilities supported by this plugin. Useful capabilities are: “identify”, “cover”

touched_fields = `frozenset({})`

List of metadata fields that can potentially be download by this plugin during the identify phase

has_html_comments = `False`

Set this to True if your plugin returns HTML formatted comments

supports_gzip_transfer_encoding = `False`

Setting this to True means that the browser object will indicate that it supports gzip transfer encoding. This can speedup downloads but make sure that the source actually supports gzip transfer encoding correctly first

ignore_ssl_errors = `False`

Set this to True to ignore HTTPS certificate errors when connecting to this source.

cached_cover_url_is_reliable = `True`

Cached cover URLs can sometimes be unreliable (i.e. the download could fail or the returned image could be bogus). If that is often the case with this source, set to False

options = `()`

A list of `Option` objects. They will be used to automatically construct the configuration widget for this plugin

config_help_message = `None`

A string that is displayed at the top of the config widget for this plugin

can_get_multiple_covers = False

If True this source can return multiple covers for a given query

auto_trim_covers = False

If set to True covers downloaded by this plugin are automatically trimmed.

prefer_results_with_isbn = True

If set to True, and this source returns multiple results for a query, some of which have ISBNs and some of which do not, the results without ISBNs will be ignored

is_configured()

Return False if your plugin needs to be configured before it can be used. For example, it might need a username/password/API key.

customization_help()

Return a string giving help on how to customize this plugin. By default raise a `NotImplementedError`, which indicates that the plugin does not require customization.

If you re-implement this method in your subclass, the user will be asked to enter a string as customization for this plugin. The customization string will be available as `self.site_customization`.

Site customization could be anything, for example, the path to a needed binary on the user's computer.

Parametri

gui – If True return HTML help, otherwise return plain text help.

config_widget()

Implement this method and `save_settings()` (pagina 255) in your plugin to use a custom configuration dialog, rather than relying on the simple string based default customization.

This method, if implemented, must return a `QWidget`. The widget can have an optional method `validate()` that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns True.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

save_settings(config_widget)

Save the settings specified by the user with `config_widget`.

Parametri

config_widget – The widget returned by `config_widget()` (pagina 255).

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()` (pagina 256).

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()` (pagina 256).

For details on the default algorithm see `InternalMetadataCompareKeyGen` (pagina 257). 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*)

Identify a book by its Title/Author/ISBN/etc.

If identifiers(s) are specified and no match is found and this metadata source does not store all related identifiers (for example, all ISBNs of a book), this method should retry with just the title and author (assuming they were specified).

If this metadata source also provides covers, the URL to the cover should be cached so that a subsequent call to the get covers API with the same ISBN/special identifier does not need to get the cover URL again. Use the caching API for this.

Every Metadata object put into result_queue by this method must have a *source_relevance* attribute that is an integer indicating the order in which the results were returned by the metadata source for this query. This integer will be used by `compare_identify_results()`. If the order is unimportant, set it to zero for every result.

Make sure that any cover/ISBN mapping information is cached before the Metadata object is put into result_queue.

Parametri

- **log** – A log object, use it to output debugging information/errors

- **result_queue** – A result Queue, results should be put into it. Each result is a Metadata object
- **abort** – If `abort.is_set()` returns True, abort further processing and return as soon as possible
- **title** – The title of the book, can be None
- **authors** – A list of authors of the book, can be None
- **identifiers** – A dictionary of other identifiers, most commonly {"isbn": "1234..."}
- **timeout** – Timeout in seconds, no network request should hang for longer than timeout.

Ritorna

None if no errors occurred, otherwise a unicode representation of the error suitable for showing to the user

download_cover (*log, result_queue, abort, title=None, authors=None, identifiers={}, timeout=30, get_best_cover=False*)

Download a cover and put it into result_queue. The parameters all have the same meaning as for *identify()* (pagina 256). Put (self, cover_data) into result_queue.

This method should use cached cover URLs for efficiency whenever possible. When cached data is not present, most plugins simply call identify and use its results.

If the parameter `get_best_cover` is True and this plugin can get multiple covers, it should only get the «best» one.

```
class calibre.ebooks.metadata.sources.base.InternalMetadataCompareKeyGen (mi,
                                                                    sour-
                                                                    ce_plugin,
                                                                    title,
                                                                    authors,
                                                                    identi-
                                                                    fiers)
```

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.

L'algoritmo è:

- Prefer results that have at least one identifier the same as for the query
- Prefer results with a cached cover URL
- Prefer results with all available fields filled in
- Prefer results with the same language as the current user interface language
- Prefer results that are an exact title match to the query
- Prefer results with longer comments (greater than 10% longer)
- **Use the relevance of the result as reported by the metadata source's search motore**

12.1.6 Conversion plugins

class calibre.customize.conversion.InputFormatPlugin(*args)

Basi: *Plugin* (pagina 248)

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()* (pagina 258).

type = 'Input per la conversione'

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

If False, the user will not be able to disable this plugin. Use with care.

supported_platforms = ['windows', 'osx', 'linux']

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: ['windows', 'osx', 'linux']

file_types = {}

Set of file types for which this plugin should be run For example: set(['azw', 'mobi', 'prc'])

is_image_collection = False

If True, this input plugin generates a collection of images, one per HTML file. This can be set dynamically, in the convert method if the input files can be both image collections and non-image collections. If you set this to True, you must implement the get_images() method that returns a list of images.

core_usage = 1

Number of CPU cores used by this plugin. A value of -1 means that it uses all available cores

for_viewer = False

If set to True, the input plugin will perform special processing to make its output suitable for viewing

output_encoding = 'utf-8'

The encoding that this input plugin creates files in. A value of None means that the encoding is undefined and must be detected individually

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Options shared by all Input format plugins. Do not override in sub-classes. Use *options* (pagina 258) 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 OEB-Book 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.

Parametri

- **stream** – A file like object that contains the input file.
- **options** – Options to customize the conversion process. Guaranteed to have attributes corresponding to all the options declared by this plugin. In addition, it will have a verbose attribute that takes integral values from zero upwards. Higher numbers mean be more verbose. Another useful attribute is `input_profile` that is an instance of `calibre.customize.profiles.InputProfile`.
- **file_ext** – The extension (without the `.`) of the input file. It is guaranteed to be one of the *file_types* supported by this plugin.
- **log** – A `calibre.utils.logging.Log` object. All output should use this object.
- **accelarators** – A dictionary of various information that the input plugin can get easily that would speed up the subsequent stages of the conversion.

postprocess_book (*oeb, opts, log*)

Called to allow the input plugin to perform postprocessing after the book has been parsed.

specialize (*oeb, opts, log, output_fmt*)

Called to allow the input plugin to specialize the parsed book for a particular output format. Called after `postprocess_book` and before any transforms are performed on the parsed book.

gui_configuration_widget (*parent, get_option_by_name, get_option_help, db, book_id=None*)

Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the `PluginWidget` class. See the builtin input plugins for examples.

class `calibre.customize.conversion.OutputFormatPlugin` (*args)

Basi: *Plugin* (pagina 248)

`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()` (pagina 260).

type = 'Output della conversione'

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

If False, the user will not be able to disable this plugin. Use with care.

supported_platforms = ['windows', 'osx', 'linux']

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: ['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* (pagina 259) 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 sys.getdefaultencoding(). 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.

Parametri

- **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.

Parametri

- **item** – L'elemento (file HTML) in corso di elaborazione
- **stylizer** – A Stylizer object containing the flattened styles for item. You can get the style for any element by stylizer.style(element).

gui_configuration_widget (parent, get_option_by_name, get_option_help, db, book_id=None)

Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin output plugins for examples.

12.1.7 Device drivers

The base class for all device drivers is *DevicePlugin* (pagina 260). However, if your device exposes itself as a USBMS drive to the operating system, you should use the USBMS class instead as it implements all the logic needed to support these kinds of devices.

class calibre.devices.interface.**DevicePlugin** (plugin_path)

Basi: *Plugin* (pagina 248)

Defines the interface that should be implemented by backends that communicate with an e-book reader.

type = 'Interfaccia dispositivo'

The type of this plugin. Used for categorizing plugins in the GUI

FORMATS = ['lrf', 'rtf', 'pdf', 'txt']

Ordered list of supported formats

VENDOR_ID = 0

VENDOR_ID can be either an integer, a list of integers or a dictionary. If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
  integer_vendor_id : { product_id : [list of BCDs], ... },
  ...
}
```

PRODUCT_ID = 0

An integer or a list of integers

BCD = None

BCD can be either None to not distinguish between devices based on BCD, or it can be a list of the BCD numbers of all devices supported by this driver.

THUMBNAIL_HEIGHT = 68

Height for thumbnails on the device

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'

L'icona per questo dispositivo

UserAnnotation

alias of Annotation

OPEN_FEEDBACK_MESSAGE = None

GUI displays this as a message if not None in the status bar. Useful if opening can take a long time

VIRTUAL_BOOK_EXTENSIONS = frozenset({})

Set of extensions that are «virtual books» on the device and therefore cannot be viewed/saved/added to library.
For example: `frozenset(['kobo'])`

VIRTUAL_BOOK_EXTENSION_MESSAGE = None

Message to display to user for virtual book extensions.

NUKE_COMMENTS = None

Whether to nuke comments in the copy of the book sent to the device. If not None this should be short string that the comments will be replaced by.

MANAGES_DEVICE_PRESENCE = False

If True indicates that this driver completely manages device detection, ejecting and so forth. If you set this to True, you *must* implement the `detect_managed_devices` and `debug_managed_device_detection` methods. A driver with this set to true is responsible for detection of devices, managing a blacklist of devices, a list of ejected devices and so forth. calibre will periodically call the `detect_managed_devices()` method and if it returns a detected device, calibre will call `open()`. `open()` will be called every time a device is returned even if previous calls to `open()` failed, therefore the driver must maintain its own blacklist of failed devices. Similarly, when ejecting, calibre will call `eject()` and then assuming the next call to `detect_managed_devices()` returns None, it will call `post_yank_cleanup()`.

SLOW_DRIVEINFO = False

If set the True, calibre will call the `get_driveinfo()` (pagina 264) 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()` (pagina 266) and `ignore_connected_device()` (pagina 266) and `get_user_blacklisted_devices()` (pagina 266) and `set_user_blacklisted_devices()` (pagina 266)

user_feedback_after_callback = None

Set this to a dictionary of the form `{“title”:title, “msg”:msg, “det_msg”:detailed_msg}` to have calibre popup a message to the user after some callbacks are run (currently only `upload_books`). Be careful to not spam the user with too many messages. This variable is checked after *every* callback, so only set it when you really need to.

classmethod get_open_popup_message ()

GUI displays this as a non-modal popup. Should be an instance of `OpenPopupMessage`

is_usb_connected (devices_on_system, debug=False, only_presence=False)

Return True, device_info if a device handled by this plugin is currently connected.

Parametri

devices_on_system – Elenco di dispositivi attualmente connessi

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.

Parametri

- **devices_on_system** – Set of USB devices found on the system.
- **force_refresh** – If True and the driver uses a cache to prevent repeated scanning, the cache must be flushed.

debug_managed_device_detection (devices_on_system, output)

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*)

Parametri

- **key** – Il tasto per sbloccare il 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\(\)](#) (pagina 263).

Note that for devices based on USBMS this method by default delegates to [can_handle\(\)](#) (pagina 263). So you only need to override [can_handle\(\)](#) (pagina 263) in your subclass of USBMS.

Parametri

usbdevice – A usbdevice as returned by `calibre.devices.winusb.scan_usb_devices()`

can_handle (*device_info, debug=False*)

Unix version of [can_handle_windows\(\)](#) (pagina 263).

Parametri

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()` (pagina 271) 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.

Parametri

- **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.

Parametri

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}.

Ritorna

(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. Memoria principale
2. Memory Card A
3. Memory Card B

Ritorna

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. Memoria principale
2. Scheda A
3. Scheda B

Ritorna

A 3 element list with free space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return -1.

books (*oncard=None, end_session=True*)

Return a list of e-books on the device.

Parametri

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.

Ritorna

A BookList.

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».

Parametri

- **files** – A list of paths
- **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).

Ritorna

A list of 3-element tuples. The list is meant to be passed to `add_books_to_metadata()` (pagina 265).

classmethod add_books_to_metadata (*locations, metadata, booklists*)

Add locations to the booklists. This function must not communicate with the device.

Parametri

- **locations** – Result of a call to `L{upload_books}`
- **metadata** – List of `Metadata` objects, same as for `upload_books()` (pagina 265).
- **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*)

Delete books at paths on device.

classmethod remove_books_from_metadata (*paths, booklists*)

Remove books from the metadata list. This function must not communicate with the device.

Parametri

- **paths** – paths to books on the device.
- **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*)

Update metadata on device.

Parametri

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`.

Parametri

outfile – file object like `sys.stdout` or the result of an `open()` (pagina 263) call.

classmethod config_widget ()

Should return a QWidget. The QWidget contains the settings for the device interface

classmethod save_settings (settings_widget)

Should save settings to disk. Takes the widget created in `config_widget ()` (pagina 265) and saves all settings to disk.

classmethod settings ()

Should return an opts object. The opts object should have at least one attribute *format_map* which is an ordered list of formats for the device.

set_plugboards (plugboards, pb_func)

provide the driver the current set of plugboards and a function to select a specific plugboard. This method is called immediately before `add_books` and `sync_booklists`.

pb_func is a callable with the following signature::

def pb_func(device_name, format, plugboards)

You give it the current device name (either the class name or `DEVICE_PLUGBOARD_NAME`), the format you are interested in (a “real” format or “device-db”), and the plugboards (you were given those by `set_plugboards`, the same place you got this method).

Ritorna

None or a single plugboard instance.

set_driveinfo_name (location_code, name)

Set the device name in the driveinfo file to “name”. This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the `get_device_information()` method.

prepare_addable_books (paths)

Given a list of paths, returns another list of paths. These paths point to addable versions of the books.

If there is an error preparing a book, then instead of a path, the position in the returned list for that book should be a three tuple: (original_path, the exception instance, traceback)

startup ()

Called when calibre is starting the device. Do any initialization required. Note that multiple instances of the class can be instantiated, and thus `__init__` can be called multiple times, but only one instance will have this method called. This method is called on the device thread, not the GUI thread.

shutdown ()

Called when calibre is shutting down, either for good or in preparation to restart. Do any cleanup required. This method is called on the device thread, not the GUI thread.

get_device_uid ()

Must return a unique id for the currently connected device (this is called immediately after a successful call to `open()`). You must implement this method if you set `ASK_TO_ALLOW_CONNECT = True`

ignore_connected_device (uid)

Should ignore the device identified by uid (the result of a call to `get_device_uid()`) in the future. You must implement this method if you set `ASK_TO_ALLOW_CONNECT = True`. Note that this function is called immediately after `open()`, so if `open()` caches some state, the driver should reset that state.

get_user_blacklisted_devices ()

Return map of device uid to friendly name for all devices that the user has asked to be ignored.

set_user_blacklisted_devices (*devices*)

Set the list of device uids that should be ignored by this driver.

specialize_global_preferences (*device_prefs*)

Implement this method if your device wants to override a particular preference. You must ensure that all call sites that want a preference that can be overridden use `device_prefs["something"]` instead of `prefs["something"]`. Your method should call `device_prefs.set_overrides(pref=val, pref=val, ...)`. Currently used for: metadata management (`prefs["manage_device_metadata"]`)

set_library_info (*library_name, library_uuid, field_metadata*)

Implement this method if you want information about the current calibre library. This method is called at startup and when the calibre library changes while connected.

is_dynamically_controllable ()

Called by the device manager when starting plugins. If this method returns a string, then a) it supports the device manager's dynamic control interface, and b) that name is to be used when talking to the plugin.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

start_plugin ()

This method is called to start the plugin. The plugin should begin to accept device connections however it does that. If the plugin is already accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

stop_plugin ()

This method is called to stop the plugin. The plugin should no longer accept connections, and should clean-up behind itself. It is likely that this method should call shutdown. If the plugin is already not accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

get_option (*opt_string, default=None*)

Return the value of the option indicated by `opt_string`. This method can be called when the plugin is not started. Return `None` if the option does not exist.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

set_option (*opt_string, opt_value*)

Set the value of the option indicated by `opt_string`. This method can be called when the plugin is not started.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

is_running ()

Return `True` if the plugin is started, otherwise `false`

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

synchronize_with_db (*db, book_id, book_metadata, first_call*)

Called during book matching when a book on the device is matched with a book in calibre's db. The method is responsible for synchronizing data from the device to calibre's db (if needed).

The method must return a two-value tuple. The first value is a set of calibre book ids changed if calibre's database was changed or `None` if the database was not changed. If the first value is an empty set then the metadata for the book on the device is updated with calibre's metadata and given back to the device, but no GUI refresh of that book is done. This is useful when the calibre data is correct but must be sent to the device.

The second value is itself a 2-value tuple. The first value in the tuple specifies whether a book format should be sent to the device. The intent is to permit verifying that the book on the device is the same as the book in calibre. This value must be `None` if no book is to be sent, otherwise return the base file name on the device (a

string like foobar.epub). Be sure to include the extension in the name. The device subsystem will construct a `send_books` job for all books with not- `None` returned values. Note: other than to later retrieve the extension, the name is ignored in cases where the device uses a template to generate the file name, which most do. The second value in the returned tuple indicated whether the format is future-dated. Return `True` if it is, otherwise return `False`. calibre will display a dialog to the user listing all future dated books.

Extremely important: this method is called on the GUI thread. It must be threadsafe with respect to the device manager's thread.

`book_id`: the calibre id for the book in the database. `book_metadata`: the `Metadata` object for the book coming from the device. `first_call`: `True` if this is the first call during a sync, `False` otherwise

class `calibre.devices.interface.BookList` (*oncard, prefix, settings*)

Basi: `list`

A list of books. Each `Book` object must have the fields

1. `titolo`
2. `authors`
3. `dimensioni` (dimensioni del file del libro)
4. `datetime` (a UTC time tuple)
5. `path` (path on the device to the book)
6. `thumbnail` (can be `None`) thumbnail is either a `str/bytes` object with the image data or it should have an attribute `image_path` that stores an absolute (platform native) path to the image
7. `tag` (una lista di stringhe, si può lasciare vuoto)

supports_collections ()

Restituisce `True` se il dispositivo supporta le collezioni per questa lista di libri.

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.

Parametri

collection_attributes – Una lista di attributi dell'oggetto Libro

Dispositivi basati su Memoria di Massa USB

The base class for such devices is `calibre.devices.usbms.driver.USBMS` (pagina 272). This class in turn inherits some of its functionality from its bases, documented below. A typical basic USBMS based driver looks like this:

```
from calibre.devices.usbms.driver import USBMS

class PDNOVEL(USBMS):
    name = 'Pandigital Novel device interface'
    gui_name = 'PD Novel'
    description = _('Communicate with the Pandigital Novel')
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'linux', 'osx']
    FORMATS = ['epub', 'pdf']

    VENDOR_ID = [0x18d1]
    PRODUCT_ID = [0xb004]
    BCD = [0x224]

    THUMBNAIL_HEIGHT = 144

    EBOOK_DIR_MAIN = 'eBooks'
    SUPPORTS_SUB_DIRS = False

    def upload_cover(self, path, filename, metadata):
        coverdata = getattr(metadata, 'thumbnail', None)
        if coverdata and coverdata[2]:
            with open('%s.jpg' % os.path.join(path, filename), 'wb') as coverfile:
                coverfile.write(coverdata[2])
```

class `calibre.devices.usbms.device.Device` (*plugin_path*)

Basi: `DeviceConfig`, `DevicePlugin` (pagina 260)

This class provides logic common to all drivers for devices that export themselves as USB Mass Storage devices. Provides implementations for mounting/ejecting of USBMS devices on all platforms.

VENDOR_ID = 0

VENDOR_ID can be either an integer, a list of integers or a dictionary. If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
    integer_vendor_id : { product_id : [list of BCDs], ... },
    ...
}
```

PRODUCT_ID = 0

An integer or a list of integers

BCD = None

BCD can be either None to not distinguish between devices based on BCD, or it can be a list of the BCD numbers of all devices supported by this driver.

WINDOWS_MAIN_MEM = None

String identifying the main memory of the device in the Windows PnP id strings. This can be None, string, list of strings or compiled regex.

WINDOWS_CARD_A_MEM = None

String identifying the first card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_B_MEM = None

String identifying the second card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

OSX_MAIN_MEM_VOL_PAT = None

Used by the new driver detection to disambiguate main memory from storage cards. Should be a regular expression that matches the main memory mount point assigned by macOS

BACKLOADING_ERROR_MESSAGE = None

MAX_PATH_LEN = 250

The maximum length of paths created on the device

NEWS_IN_FOLDER = True

Inserisci le news nella loro cartella

reset (*key='-1', log_packets=False, report_progress=None, detected_device=None*)

Parametri

- **key** – Il tasto per sbloccare il 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.

Parametri

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. Memoria principale
2. Memory Card A
3. Memory Card B

Ritorna

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. Memoria principale
2. Scheda A
3. Scheda B

Ritorna

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*)

Chiamata per eliminare l'ambiguità tra la memoria principale e la scheda di memoria in dispositivi che non le distinguono sulla base di *WINDOWS_CARD_NAME*. Per esempio: L'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.

Parametri

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()` (pagina 271) 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.

Parametri

- **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*)

Aggiungi annotazione alla libreria di calibre

class calibre.devices.usbms.cli.CLI

class calibre.devices.usbms.driver.USBMS (*plugin_path*)

Basi: [CLI](#) (pagina 272), [Device](#) (pagina 269)

La classe base per tutti i dispositivi USBMS. Implementa la logica per inviare/ricevere/aggiornare metadati/raccogliere metadati/etc.

description = 'Comunica con un lettore di e-book.'

Una breve stringa che descrive cosa fa questo plugin

author = 'John Schember'

L'autore di questo plugin

supported_platforms = ['windows', 'osx', 'linux']

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: ['windows', 'osx', 'linux']

booklist_class

alias of BookList

book_class

alias of Book

FORMATS = []

Ordered list of supported formats

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}`.

Ritorna

(device name, device version, software version on device, MIME type) The tuple can optionally have a fifth element, which is a drive information dictionary. See `usbms.driver` for an example.

set_driveinfo_name (*location_code, name*)

Set the device name in the driveinfo file to “name”. This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the `get_device_information()` method.

books (*oncard=None, end_session=True*)

Return a list of e-books on the device.

Parametri

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.

Ritorna

A BookList.

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».

Parametri

- **files** – A list of paths
- **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).

Ritorna

A list of 3-element tuples. The list is meant to be passed to `add_books_to_metadata()` (pagina 273).

upload_cover (*path, filename, metadata, filepath*)

Upload book cover to the device. Default implementation does nothing.

Parametri

- **path** – The full path to the folder where the associated book is located.
- **filename** – The name of the book file without the extension.
- **metadata** – metadata belonging to the book. Use `metadata.thumbnail` for cover
- **filepath** – The full path to the e-book file

add_books_to_metadata (*locations, metadata, booklists*)

Add locations to the booklists. This function must not communicate with the device.

Parametri

- **locations** – Result of a call to `L{upload_books}`
- **metadata** – List of `Metadata` objects, same as for `upload_books()` (pagina 273).
- **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*)

Delete books at paths on device.

remove_books_from_metadata (*paths, booklists*)

Remove books from the metadata list. This function must not communicate with the device.

Parametri

- **paths** – paths to books on the device.
- **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*)

Update metadata on device.

Parametri

booklists – A tuple containing the result of calls to `(books (oncard=None) (), books (oncard='carda') (), :meth`books(oncard="cardb"))`.

classmethod normalize_path (*path*)

Return path with platform native path separators

12.1.8 User interface actions

If you are adding your own plugin in a ZIP file, you should subclass both `InterfaceActionBase` and `InterfaceAction`. The `load_actual_plugin()` method of your `InterfaceActionBase` subclass must return an instantiated object of your `InterfaceBase` subclass.

class `calibre.gui2.actions.InterfaceAction` (*parent, site_customization*)

Basi: `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` (pagina 274) objects have the same name, the one with higher priority takes precedence.

Sub-classes should implement the `genesis()` (pagina 276), `library_changed()` (pagina 277), `location_selected()` (pagina 276), `shutting_down()` (pagina 277), `initialization_complete()` (pagina 277) and `tag_browser_context_action()` (pagina 277) 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` (pagina 275) is automatically create and made available as `self.qaction`.

name = 'Implement me'

The plugin name. If two plugins with the same name are present, the one with higher priority takes precedence.

priority = 1

The plugin priority. If two plugins with the same name are present, the one with higher priority takes precedence.

popup_type = 1

The menu popup type for when this plugin is added to a toolbar

auto_repeat = False

Whether this action should be auto repeated when its shortcut key is held down.

action_spec = ('text', 'icon', None, None)

Of the form: (text, icon_path, tooltip, keyboard shortcut). icon, tooltip and keyboard shortcut can be None. keyboard shortcut must be either a string, None or tuple of shortcuts. If None, a keyboard shortcut corresponding to the action is not registered. If you pass an empty tuple, then the shortcut is registered with no default key binding.

action_shortcut_name = None

If not None, used for the name displayed to the user when customizing the keyboard shortcuts for the above action spec instead of action_spec[0]

action_add_menu = False

If True, a menu is automatically created and added to self.qaction

action_menu_clone_qaction = False

If True, a clone of self.qaction is added to the menu of self.qaction. If you want the text of this action to be different from that of self.qaction, set this variable to the new text

dont_add_to = frozenset({})

Set of locations to which this action must not be added. See `all_locations` for a list of possible locations

dont_remove_from = frozenset({})

Set of locations from which this action must not be removed. See `all_locations` for a list of possible locations

action_type = 'global'

Type of action “current” means acts on the current view “global” means an action that does not act on the current view, but rather on calibre as a whole

accepts_drops = False

If True, then this InterfaceAction will have the opportunity to interact with drag and drop events. See the methods, `accept_enter_event()` (pagina 275), `meth`accept_drag_move_event``, `drop_event()` (pagina 275) 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.

Parametri

- **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** – The text of the action.
- **icon** – Either a QIcon or a file name. The file name is passed to the QIcon.ic() builtin, so you do not need to pass the full path to the images folder.
- **shortcut** – A string, a list of strings, None or False. If False, no keyboard shortcut is registered for this action. If None, a keyboard shortcut with no default keybinding is registered. String and list of strings register a shortcut with default keybinding as specified.
- **description** – A description for this action. Used to set tooltips.
- **triggered** – A callable which is connected to the triggered signal of the created action.
- **shortcut_name** – The text displayed to the user when customizing the keyboard shortcuts for this action. By default it is set to the value of `text`.
- **persist_shortcut** – Shortcuts for actions that don't always appear, or are library dependent, may disappear when other keyboard shortcuts are edited unless ``persist_shortcut`` is set True.

load_resources (*names*)

If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

For example to load an image:

```

pixmap = QPixmap()
pixmap.loadFromData(tuple(self.load_resources(['images/icon.png'])).
    ↪values())[0])
icon = QIcon(pixmap)

```

Parametri

names – List of paths to resources in the ZIP file using / as separator

Ritorna

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` (pagina 275) is available as `self.qaction`.

location_selected (*loc*)

Called whenever the book list being displayed in calibre changes. Currently values for `loc` are: `library`, `main`, `card` and `cardb`.

This method should enable/disable this action and its sub actions as appropriate for the location.

library_about_to_change (*olddb, db*)

Called whenever the current library is changed.

Parametri

- **olddb** – The LibraryDatabase corresponding to the previous library.
- **db** – The LibraryDatabase corresponding to the new library.

library_changed (*db*)

Called whenever the current library is changed.

Parametri

- 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()* (pagina 277).

initialization_complete ()

Called once per action when the initialization of the main GUI is completed.

tag_browser_context_action (*index*)

Called when displaying the context menu in the Tag browser. *index* is the QModelIndex that points to the Tag browser item that was right clicked. Test it for validity with *index.valid()* and get the underlying TagTreeWidgetItem 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)

Basi: *Plugin* (pagina 248)

supported_platforms = ['windows', 'osx', 'linux']

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: ['windows', 'osx', 'linux']

author = 'Kovid Goyal'

L'autore di questo plugin

type = "Azione dell'interfaccia utente"

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

If False, the user will not be able to disable this plugin. Use with care.

load_actual_plugin (*gui*)

This method must return the actual interface action plugin object.

12.1.9 Preferences plugins

class calibre.customize.PreferencesPlugin(*plugin_path*)

Basi: *Plugin* (pagina 248)

A plugin representing a widget displayed in the Preferences dialog.

This plugin has only one important method *create_widget()* (pagina 278). The various fields of the plugin control how it is categorized in the UI.

supported_platforms = ['windows', 'osx', 'linux']

Elenco delle piattaforme sulle quali questo plugin funziona. Ad esempio: ['windows', 'osx', 'linux']

author = 'Kovid Goyal'

L'autore di questo plugin

type = 'Preferenze'

The type of this plugin. Used for categorizing plugins in the GUI

can_be_disabled = False

If False, the user will not be able to disable this plugin. Use with care.

config_widget = None

Import path to module that contains a class named ConfigWidget which implements the ConfigWidgetInterface. Used by *create_widget()* (pagina 278).

category_order = 100

Where in the list of categories the *category* (pagina 278) of this plugin should be.

name_order = 100

Where in the list of names in a category, the *gui_name* (pagina 278) of this plugin should be

category = None

The category this plugin should be in

gui_category = None

The category name displayed to the user for this plugin

gui_name = None

The name displayed to the user for this plugin

icon = None

The icon for this plugin, should be an absolute path

description = None

The description used for tooltips and the like

create_widget (*parent=None*)

Create and return the actual Qt widget used for setting this group of preferences. The widget must implement the *calibre.gui2.preferences.ConfigWidgetInterface* (pagina 278).

The default implementation uses *config_widget* (pagina 278) to instantiate the widget.

class calibre.gui2.preferences.ConfigWidgetInterface

This class defines the interface that all widgets displayed in the Preferences dialog must implement. See *ConfigWidgetBase* (pagina 279) for a base class that implements this interface and defines various convenience methods as well.

changed_signal = None

This signal must be emitted whenever the user changes a value in this widget

supports_restoring_to_defaults = True

Set to True iff the `restore_to_defaults()` method is implemented.

restore_defaults_desc = 'Ripristina le impostazioni predefinite. È necessario fare clic su Applica per salvare le impostazioni predefinite.'

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()` (pagina 279) returns True.

genesis(gui)

Called once before the widget is displayed, should perform any necessary setup.

Parametri

gui – The main calibre graphical user interface

initialize()

Should set all config values to their initial values (the values stored in the config files). A «return» statement is optional. Return False if the dialog is not to be shown.

restore_defaults()

Should set all config values to their defaults.

commit()

Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an `AbortCommit` exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

refresh_gui(gui)

Called once after this widget is committed. Responsible for causing the gui to reread any changed settings. Note that by default the GUI re-initializes various elements anyway, so most widgets won't need to use this method.

initial_tab_changed()

Called if the initially displayed tab is changed before the widget is shown, but after it is initialized.

class calibre.gui2.preferences.ConfigWidgetBase (*parent=None*)

Base class that contains code to easily add standard config widgets like checkboxes, combo boxes, text fields and so on. See the `register()` (pagina 280) method.

This class automatically handles change notification, resetting to default, translation between gui objects and config objects, etc. for registered settings.

If your config widget inherits from this class but includes setting that are not registered, you should override the `ConfigWidgetInterface` (pagina 278) methods and call the base class methods inside the overrides.

changed_signal

This signal must be emitted whenever the user changes a value in this widget

supports_restoring_to_defaults = True

Set to True iff the `restore_to_defaults()` method is implemented.

restart_critical = False

If True the Preferences dialog will not allow the user to set any more preferences. Only has effect if `commit()` (pagina 280) 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.

Parametri

- **name** – The setting name
- **config** – The config object that reads/writes the setting
- **gui_name** – The name of the GUI object that presents an interface to change the setting. By default it is assumed to be 'opt_' + name.
- **choices** – If this setting is a multiple choice (combobox) based setting, the list of choices. The list is a list of two element tuples of the form: [(gui name, value), ...]
- **setting** – The class responsible for managing this setting. The default class handles almost all cases, so this param is rarely used.

initialize()

Should set all config values to their initial values (the values stored in the config files). A «return» statement is optional. Return False if the dialog is not to be shown.

commit (*args)

Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an `AbortCommit` exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

restore_defaults (*args)

Should set all config values to their defaults.

12.2 Variabili d'ambiente

- `CALIBRE_CONFIG_DIRECTORY` - Cartella nella quale i file di configurazione vengono salvati/letti
- `CALIBRE_TEMP_DIR` - Cartella dei file temporanei utilizzata da calibre
- `CALIBRE_CACHE_DIRECTORY` - cartella usata da calibre per tenere in cache i dati persistenti tra le sessioni
- `CALIBRE_OVERRIDE_DATABASE_PATH` - Ti permette di specificare il percorso completo di metadata.db. Utilizzando questa variabile puoi avere metadata.db in una posizione differente dalla cartella della libreria. Utile se la tua cartella della libreria si trova in una unità di rete che non supporta il blocco dei file
- `CALIBRE_DEVELOP_FROM` - utilizzato per eseguire calibre in un ambiente di sviluppo. consulta [Creazione di un ambiente di sviluppo di calibre](#) (pagina 337).
- `CALIBRE_OVERRIDE_LANG` - utilizzato per forzare la lingua utilizzata dall'interfaccia (codice lingua ISO 639)
- `CALIBRE_TEST_TRANSLATION` - utilizzato per provare un file translation.po (deve essere il percorso del file .po)
- `CALIBRE_NO_NATIVE_FILEDIALOGS` - fa sì che calibre non usi le finestre di dialogo native per selezionare file / directory.
- `CALIBRE_NO_NATIVE_MENUBAR` - fa in modo che calibre non crei un menù nativo (globale) in Ubuntu Unity e in ambienti desktop Linux simili. Il menù viene invece posizionato dentro la finestra, come da tradizione.

- `CALIBRE_USE_SYSTEM_THEME` - per impostazione predefinita, su Linux, calibre usa il proprio stile Qt interno. Questo per evitare problemi e blocchi causati da incompatibilità tra la versione di Qt su cui calibre è compilato e quella di sistema. Il lato negativo è che calibre potrebbe non adattarsi all'aspetto del sistema. Se imposti questa variabile d'ambiente su Linux, farà in modo che calibre usi il tema di sistema – preparati a problemi e blocchi.
- `CALIBRE_SHOW_DEPRECATION_WARNINGS` - fa in modo che calibre mostri avvisi di deprecazione in stdout. Utile per gli sviluppatori di calibre.
- `CALIBRE_NO_DEFAULT_PROGRAMS` - impedisce che calibre registri automaticamente i tipi di file che è in grado di gestire con Windows.
- `QT_QPA_PLATFORM` - Su Linux, imposta su `wayland` per forzare calibre a usare Wayland, e su `xcb` per forzare l'uso di X11.
- `SYSFS_PATH` - Usa se `sysfs` è montato da qualche altra parte che non è `/sys`
- `http_proxy`, `https_proxy` - utilizzato su Linux per specificare un proxy HTTP(S)

Vedi [How to set environment variables in Windows](#)¹⁰⁵. Se sei su macOS puoi impostare le variabili d'ambiente creando il `~/Library/Preferences/calibre/macOS-env.txt` e mettendoci una variabile d'ambiente per linea, per esempio:

```
CALIBRE_DEVELOP_FROM=$HOME/calibre-src/src
CALIBRE_NO_NATIVE_FILEDIALOGS=1
CALIBRE_CONFIG_DIRECTORY=~/.config/calibre
```

12.3 Ottimizzazioni

Le ottimizzazioni sono piccoli cambiamenti che puoi fare per controllare vari aspetti del comportamento di calibre. Puoi cambiarle andando in Preferenze->Avanzate->Ottimizzazioni. I valori predefiniti per le ottimizzazioni sono riprodotti qui sotto

```
#!/usr/bin/env python
# vim:fileencoding=UTF-8:ts=4:sw=4:sta:et:sts=4:ai
# License: GPLv3 Copyright: 2010, Kovid Goyal <kovid at kovidgoyal.net>

# Contains various tweaks that affect calibre behavior. Only edit this file if
# you know what you are doing. If you delete this file, it will be recreated from
# defaults.

#: Auto increment series index
# The algorithm used to assign a book added to an existing series a series number.
# New series numbers assigned using this tweak are always integer values, except
# if a constant non-integer is specified.
# Possible values are:
#   next - First available integer larger than the largest existing number
#   first_free - First available integer larger than 0
#   next_free - First available integer larger than the smallest existing number
#   last_free - First available integer smaller than the largest existing number.
#   ↳Return largest existing + 1 if no free number is found
#   const - Assign the number 1 always
#   no_change - Do not change the series index
#   a number - Assign that number always. The number is not in quotes. Note that 0.0
#   ↳can be used here.
```

(continues on next page)

¹⁰⁵ <https://www.computerhope.com/issues/ch000549.htm>

(continua dalla pagina precedente)

```

# Examples:
#   series_index_auto_increment = 'next'
#   series_index_auto_increment = 'next_free'
#   series_index_auto_increment = 16.5
#
# Set the use_series_auto_increment_tweak_when_importing tweak to True to
# use the above values when importing/adding books. If this tweak is set to
# False (the default) then the series number will be set to 1 if it is not
# explicitly set during the import. If set to True, then the
# series index will be set according to the series_index_auto_increment setting.
# Note that the use_series_auto_increment_tweak_when_importing tweak is used
# only when a value is not provided during import. If the importing regular
# expression produces a value for series_index, or if you are reading metadata
# from books and the import plugin produces a value, then that value will
# be used irrespective of the setting of the tweak.
series_index_auto_increment = 'next'
use_series_auto_increment_tweak_when_importing = False

#: Add separator after completing an author name
# Set this if the completion separator should be appended to the end of the
# completed text to automatically begin a new completion operation for authors.
# It can be either True or False
authors_completer_append_separator = False

#: Author sort name algorithm
# The algorithm used to copy author to author_sort.
# Possible values are:
#   invert: use "fn ln" -> "ln, fn"
#   copy   : copy author to author_sort without modification
#   comma  : use 'copy' if there is a ',' in the name, otherwise use 'invert'
#   nocomma : "fn ln" -> "ln fn" (without the comma)
# When this tweak is changed, the author_sort values stored with each author
# must be recomputed by right-clicking on an author in the left-hand tags
# panel, selecting 'Manage authors', and pressing
# 'Recalculate all author sort values'.
#
# The author_name_suffixes are words that are ignored when they occur at the
# end of an author name. The case of the suffix is ignored and trailing
# periods are automatically handled.
#
# The same is true for author_name_prefixes.
#
# The author_name_copywords are a set of words which, if they occur in an
# author name, cause the automatically generated author sort string to be
# identical to the author's name. This means that the sort for a string like
# "Acme Inc." will be "Acme Inc." instead of "Inc., Acme".
#
# If author_use_surname_prefixes is enabled, any of the words in
# author_surname_prefixes will be treated as a prefix to the surname, if they
# occur before the surname. So for example, "John von Neumann" would be sorted
# as "von Neumann, John" and not "Neumann, John von".
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 = (

```

(continues on next page)

(continua dalla pagina precedente)

```

    'Agency', 'Corporation', 'Company', 'Co.', 'Council',
    'Committee', 'Inc.', 'Institute', 'National', 'Society', 'Club', 'Team',
    'Software', 'Games', 'Entertainment', 'Media', 'Studios',
)
author_use_surname_prefixes = False
author_surname_prefixes = ('da', 'de', 'di', 'la', 'le', 'van', 'von')

#: Splitting multiple author names
# By default, calibre splits a string containing multiple author names on
# ampersands and the words "and" and "with". You can customize the splitting
# by changing the regular expression below. Strings are split on whatever the
# specified regular expression matches, in addition to ampersands.
# Default: r'(?i),?\s+(and|with)\s+'
authors_split_regex = r'(?i),?\s+(and|with)\s+'

#: Use author sort in Tag browser
# Set which author field to display in the Tag browser (the list of authors,
# series, publishers etc on the left hand side). The choices are author and
# author_sort. This tweak affects only what is displayed under the authors
# category in the Tag browser and Content server. Please note that if you set this
# to author_sort, it is very possible to see duplicate names in the list because
# although it is guaranteed that author names are unique, there is no such
# guarantee for author_sort values. Showing duplicates won't break anything, but
# it could lead to some confusion. When using 'author_sort', the tooltip will
# show the author's name.
# Examples:
#   categories_use_field_for_author_name = 'author'
#   categories_use_field_for_author_name = 'author_sort'
categories_use_field_for_author_name = 'author'

#: Control partitioning of Tag browser
# When partitioning the Tag browser, the format of the subcategory label is
# controlled by a template: categories_collapsed_name_template if sorting by
# name, categories_collapsed_rating_template if sorting by average rating, and
# categories_collapsed_popularity_template if sorting by popularity. There are
# two variables available to the template: first and last. The variable 'first'
# is the initial item in the subcategory, and the variable 'last' is the final
# item in the subcategory. Both variables are 'objects'; they each have multiple
# values that are obtained by using a suffix. For example, first.name for an
# author category will be the name of the author. The sub-values available are:
#   name: the printable name of the item
#   count: the number of books that references this item
#   avg_rating: the average rating of all the books referencing this item
#   sort: the sort value. For authors, this is the author_sort for that author
#   category: the category (e.g., authors, series) that the item is in.
# Note that the "r" in front of the { is necessary if there are backslashes
# (\ characters) in the template. It doesn't hurt anything to leave it there
# even if there aren't any backslashes.
categories_collapsed_name_template = r'{first.sort:shorten(4,,0)} - {last.
↪sort:shorten(4,,0)}'
categories_collapsed_rating_template = r'{first.avg_rating:4.2f:ifempty(0)} - {last.
↪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

```

(continues on next page)

(continua dalla pagina precedente)

```

# otherwise it is a list of column,order pairs. Column is the
# lookup/search name, found using the tooltip for the column
# Order is 0 for ascending, 1 for descending.
# For example, set it to [('authors',0),('title',0)] to sort by
# title within authors.
sort_columns_at_startup = None

#: Control how dates are displayed
# Format to be used for publication date and the timestamp (date).
# A string controlling how the publication date is displayed in the GUI
# d      the day as number without a leading zero (1 to 31)
# dd     the day as number with a leading zero (01 to 31)
# ddd    the abbreviated localized day name (e.g. 'Mon' to 'Sun').
# dddd   the long localized day name (e.g. 'Monday' to 'Sunday').
# M      the month as number without a leading zero (1-12)
# MM     the month as number with a leading zero (01-12)
# MMM    the abbreviated localized month name (e.g. 'Jan' to 'Dec').
# MMMM   the long localized month name (e.g. 'January' to 'December').
# yy     the year as two digit number (00-99)
# yyyy   the year as four digit number
# h      the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm) '
# hh     the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm) '
# m      the minutes without a leading 0 (0 to 59) '
# mm     the minutes with a leading 0 (00 to 59) '
# s      the seconds without a leading 0 (0 to 59) '
# ss     the seconds with a leading 0 (00 to 59) '
# ap     use a 12-hour clock instead of a 24-hour clock, with "ap" replaced by the_
↳ localized string for am or pm
# AP     use a 12-hour clock instead of a 24-hour clock, with "AP" replaced by the_
↳ localized string for AM or PM
# iso    the date with time and timezone. Must be the only format present
# For example, given the date of 9 Jan 2010, the following formats show
# MMM yyyy ==> Jan 2010      yyyy ==> 2010      dd MMM yyyy ==> 09 Jan 2010
# MM/yyyy ==> 01/2010      d/M/yy ==> 9/1/10      yy ==> 10
#
# publication default if not set: MMM yyyy
# timestamp default if not set: dd MMM yyyy
# last_modified_display_format if not set: dd MMM yyyy
gui_pubdate_display_format = 'MMM yyyy'
gui_timestamp_display_format = 'dd MMM yyyy'
gui_last_modified_display_format = 'dd MMM yyyy'

#: Control sorting of titles and series in the library display
# Control title and series sorting in the library view. If set to
# 'library_order', the title sort field will be used instead of the title.
# Unless you have manually edited the title sort field, leading articles such as
# The and A will be ignored. If set to 'strictly_alphabetic', the titles will be
# sorted as-is (sort by title instead of title sort). For example, with
# library_order, The Client will sort under 'C'. With strictly_alphabetic, the
# book will sort under 'T'.
# This flag affects calibre's library display. It has no effect on devices. In
# 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'

```

(continues on next page)

(continua dalla pagina precedente)

```

#: Control formatting of title and series when used in templates
# Control how title and series names are formatted when saving to disk/sending
# to device. The behavior depends on the field being processed. If processing
# title, then if this tweak is set to 'library_order', the title will be
# replaced with title_sort. If it is set to 'strictly_alphabetic', then the
# title will not be changed. If processing series, then if set to
# 'library_order', articles such as 'The' and 'An' will be moved to the end. If
# set to 'strictly_alphabetic', the series will be sent without change.
# For example, if the tweak is set to library_order, "The Lord of the Rings"
# will become "Lord of the Rings, The". If the tweak is set to
# strictly_alphabetic, it would remain "The Lord of the Rings". Note that the
# formatter function raw_field will return the base value for title and
# series regardless of the setting of this tweak.
save_template_title_series_sorting = 'library_order'

#: Set the list of words considered to be "articles" for sort strings
# Set the list of words that are to be considered 'articles' when computing the
# title sort strings. The articles differ by language. By default, calibre uses
# a combination of articles from English and whatever language the calibre user
# interface is set to. In addition, in some contexts where the book language is
# available, the language of the book is used. You can change the list of
# articles for a given language or add a new language by editing
# per_language_title_sort_articles. To tell calibre to use a language other
# than the user interface language, set, default_language_for_title_sort. For
# example, to use German, set it to 'deu'. A value of None means the user
# interface language is used. The setting title_sort_articles is ignored
# (present only for legacy reasons).
per_language_title_sort_articles = {
    # English
    'eng' : (r'A\s+', r'The\s+', r'An\s+'),
    # Esperanto
    'epo': (r'La\s+', r"L'", 'L'),
    # Spanish
    'spa' : (r'El\s+', r'La\s+', r'Lo\s+', r'Los\s+', r'Las\s+', r'Un\s+',
            r'Una\s+', r'Unos\s+', r'Unas\s+'),
    # French
    'fra' : (r'Le\s+', r'La\s+', r"L'", u'L', u'L', r'Les\s+', r'Un\s+', r'Une\
→s+',
            r'Des\s+', r'De\s+La\s+', r'De\s+', r"D'", r'D', r'D'),
    # Polish
    'pol': (),
    # Italian
    'ita': ('Lo\\s+', 'Il\\s+', "L'", 'L', 'La\\s+', 'Gli\\s+',
            'I\\s+', 'Le\\s+', 'Uno\\s+', 'Un\\s+', 'Una\\s+', "Un'",
            'Un', 'Dei\\s+', 'Degli\\s+', 'Delle\\s+', 'Del\\s+',
            'Della\\s+', 'Dello\\s+', "Dell'", 'Dell'),
    # Portuguese
    'por' : (r'A\s+', r'O\s+', r'Os\s+', r'As\s+', r'Um\s+', r'Uns\s+',
            r'Uma\s+', r'Umas\s+', ),
    # Romanian
    'ron' : (r'Un\s+', r'O\s+', r'Niște\s+', ),
    # German
    'deu' : (r'Der\s+', r'Die\s+', r'Das\s+', r'Den\s+', r'Ei\\s+',
            r'Eine\s+', r'Einen\s+', r'Dem\s+', r'Des\s+', r'Einem\s+',
            r'Eines\s+'),
    # Dutch

```

(continues on next page)

(continua dalla pagina precedente)

```

'nld' : (r'De\s+', r'Het\s+', r'Een\s+', r"'n\s+", r"'s\s+', r'Ene\s+',
        r'Ener\s+', r'Enes\s+', r'Den\s+', r'Der\s+', r'Des\s+',
        r"'t\s+"),
# Swedish
'swe' : (r'En\s+', r'Ett\s+', r'Det\s+', r'Den\s+', r'De\s+', ),
# Turkish
'tur' : (r'Bir\s+', ),
# Afrikaans
'afr' : (r"'n\s+", r'Die\s+', ),
# Greek
'ell' : (r'O\s+', r'I\s+', r'To\s+', r'Ta\s+', r'Tus\s+', r'Tis\s+',
        r"'Enas\s+", r"'Mia\s+", r"'Ena\s+", r"'Enan\s+", ),
# Hungarian
'hun' : (r'A\s+', r'Az\s+', r'Egy\s+',),
}
default_language_for_title_sort = None
title_sort_articles=r'^(A|The|An)\s+'

#: Specify a folder calibre should connect to at startup
# Specify a folder that calibre should connect to at startup using
# connect_to_folder. This must be a full path to the folder. If the folder does
# not exist when calibre starts, it is ignored.
# Example for Windows:
#     auto_connect_to_folder = 'C:/Users/someone/Desktop/testlib'
# Example for other operating systems:
#     auto_connect_to_folder = '/home/dropbox/My Dropbox/someone/library'
auto_connect_to_folder = ''

#: Specify renaming rules for SONY collections
# Specify renaming rules for SONY collections. This tweak is only applicable if
# metadata management is set to automatic. Collections on SONYs are named
# depending upon whether the field is standard or custom. A collection derived
# from a standard field is named for the value in that field.
#
# For example, if the standard 'series' column contains the value 'Darkover', then the
# collection name is 'Darkover'. A collection derived from a custom field will
# have the name of the field added to the value. For example, if a custom series
# column named 'My Series' contains the name 'Darkover', then the collection
# will by default be named 'Darkover (My Series)'. For purposes of this
# documentation, 'Darkover' is called the value and 'My Series' is called the
# category. If two books have fields that generate the same collection name,
# then both books will be in that collection.
#
# This set of tweaks lets you specify for a standard or custom field how
# the collections are to be named. You can use it to add a description to a
# standard field, for example 'Foo (Tag)' instead of the 'Foo'. You can also use
# it to force multiple fields to end up in the same collection.
#
# For example, you could force the values in 'series', '#my_series_1', and
# '#my_series_2' to appear in collections named 'some_value (Series)', thereby
# merging all of the fields into one set of collections.
#
# There are two related tweaks. The first determines the category name to use
# for a metadata field. The second is a template, used to determines how the
# value and category are combined to create the collection name.
# The syntax of the first tweak, sony_collection_renaming_rules, is:
# {'field_lookup_name': 'category_name_to_use', 'lookup_name': 'name', ...}

```

(continues on next page)

(continua dalla pagina precedente)

```

#
# 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.
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

```

(continues on next page)

(continua dalla pagina precedente)

```
#: Set the maximum number of sort 'levels'
# Set the maximum number of sort 'levels' that calibre will use to resort the
# library after certain operations such as searches or device insertion. Each
# sort level adds a performance penalty. If the database is large (thousands of
# books) the penalty might be noticeable. If you are not concerned about multi-
# level sorts, and if you are seeing a slowdown, reduce the value of this tweak.
maximum_resort_levels = 5

#: Choose whether dates are sorted using visible fields
# Date values contain both a date and a time. When sorted, all the fields are
# used, regardless of what is displayed. Set this tweak to True to use only
# the fields that are being displayed.
sort_dates_using_visible_fields = False

#: Fuzz value for trimming covers
# The value used for the fuzz distance when trimming a cover.
# Colors within this distance are considered equal.
# The distance is in absolute intensity units.
cover_trim_fuzz_value = 10

#: Control behavior of the book list
# You can control the behavior of double clicks and pressing Enter on the books
# list. Choices: open_viewer, do_nothing, show_book_details, edit_cell,
# edit_metadata. Selecting anything other than open_viewer or show_book_details
# has the side effect of disabling editing a field using a single click.
# Default: open_viewer.
# Example: doubleclick_on_library_view = 'do_nothing'
# You can also control whether the book list scrolls per item or
# per pixel. Default is per item.
doubleclick_on_library_view = 'open_viewer'
enter_key_behavior = 'do_nothing'
horizontal_scrolling_per_column = False
vertical_scrolling_per_row = False

#: Language to use when sorting
# Setting this tweak will force sorting to use the
# collating order for the specified language. This might be useful if you run
# calibre in English but want sorting to work in the language where you live.
# Set the tweak to the desired ISO 639-1 language code, in lower case.
# You can find the list of supported locales at
# https://en.wikipedia.org/wiki/List\_of\_ISO\_639-1\_codes
# Default: locale_for_sorting = '' -- use the language calibre displays in
# Example: locale_for_sorting = 'fr' -- sort using French rules.
# Example: locale_for_sorting = 'nb' -- sort using Norwegian rules.
locale_for_sorting = ''

#: The number of seconds to wait before sending emails
# The number of seconds to wait before sending emails when using a
# public email server like GMX/Hotmail/Gmail. Default is: 5 minutes
# Setting it to lower may cause the server's SPAM controls to kick in,
# making email sending fail. Changes will take effect only after a restart of
# calibre. You can also change the list of hosts that calibre considers
# to be public relays here. Any relay host ending with one of the suffixes
# in the list below will be considered a public email server.
public_smtp_relay_delay = 301
public_smtp_relay_host_suffixes = ['gmail.com', 'live.com', 'gmx.com', 'outlook.com']
```

(continues on next page)

(continua dalla pagina precedente)

```

#: The maximum width and height for covers saved in the calibre library
# All covers in the calibre library will be resized, preserving aspect ratio,
# to fit within this size. This is to prevent slowdowns caused by extremely
# large covers
maximum_cover_size = (1650, 2200)

#: Where to send downloaded news
# When automatically sending downloaded news to a connected device, calibre
# will by default send it to the main memory. By changing this tweak, you can
# control where it is sent. Valid values are "main", "carda", "cardb". Note
# that if there isn't enough free space available on the location you choose,
# the files will be sent to the location with the most free space.
send_news_to_device_location = "main"

#: Unified toolbar on macOS
# If you enable this option and restart calibre, the toolbar will be 'unified'
# with the titlebar as is normal for macOS applications. However, doing this has
# various bugs, for instance the minimum width of the toolbar becomes twice
# what it should be and it causes other random bugs on some systems, so turn it
# on at your own risk!
unified_title_toolbar_on_osx = False

#: Save original file when converting/polishing from same format to same format
# When calibre does a conversion from the same format to the same format, for
# example, from EPUB to EPUB, the original file is saved, so that in case the
# conversion is poor, you can tweak the settings and run it again. By setting
# this to False you can prevent calibre from saving the original file.
# Similarly, by setting save_original_format_when_polishing to False you can
# prevent calibre from saving the original file when polishing.
save_original_format = True
save_original_format_when_polishing = True

#: Number of recently viewed books to show
# Right-clicking the "View" button shows a list of recently viewed books. Control
# how many should be shown, here.
gui_view_history_size = 15

#: Change the font size of the Book details panel in the interface
# Change the font size at which book details are rendered in the side panel and
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0

#: What format to default to when using the "Unpack book" feature
# The "Unpack book" feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
# to that format rather than your output format preference.
# 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

```

(continues on next page)

(continua dalla pagina precedente)

```
#: Do not preselect a completion when editing authors/tags/series/etc.
# This means that you can make changes and press Enter and your changes will
# not be overwritten by a matching completion. However, if you wish to use the
# completions you will now have to press Tab to select one before pressing
# Enter. Which technique you prefer will depend on the state of metadata in
# your library and your personal editing style.
#
# If preselect_first_completion is False and you want Tab to accept what you
# typed instead of the first completion then set tab_accepts_uncompleted_text
# to True. If you do this then to select from the completions you must press
# the Down or Up arrow keys. The tweak tab_accepts_uncompleted_text is ignored
# if preselect_first_completion is True
preselect_first_completion = False
tab_accepts_uncompleted_text = False

#: Completion mode when editing authors/tags/series/etc.
# By default, when completing items, calibre will show you all the candidates
# that start with the text you have already typed. You can instead have it show
# all candidates that contain the text you have already typed. To do this, set
# completion_mode to 'contains'. For example, if you type asi it will match both
# Asimov and Quasimodo, whereas the default behavior would match only Asimov.
completion_mode = 'prefix'

#: Sort the list of libraries alphabetically
# The list of libraries in the Copy to library and Quick switch menus are
# normally sorted by most used. However, if there are more than a certain
# number of such libraries, the sorting becomes alphabetic. You can set that
# number here. The default is ten libraries.
many_libraries = 10

#: Choose available output formats for conversion
# Restrict the list of available output formats in the conversion dialogs.
# For example, if you only want to convert to EPUB and AZW3, change this to
# restrict_output_formats = ['EPUB', 'AZW3']. The default value of None causes
# all available output formats to be present.
restrict_output_formats = None

#: Set the thumbnail image quality used by the Content server
# The quality of a thumbnail is largely controlled by the compression quality
# used when creating it. Set this to a larger number to improve the quality.
# Note that the thumbnails get much larger with larger compression quality
# numbers.
# The value can be between 50 and 99
content_server_thumbnail_compression_quality = 75

#: Image file types to treat as e-books when dropping onto the "Book details" panel
# Normally, if you drop any image file in a format known to calibre onto the
# "Book details" panel, it will be used to set the cover. If you want to store
# some image types as e-books instead, you can set this tweak.
# 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,
```

(continues on next page)

(continua dalla pagina precedente)

```

# exclude_fields_on_paste = ['cover', 'timestamp', '#mycolumn']
# to prevent pasting of the cover, Date and custom column, mycolumn.
# You can also add a shortcut in Preferences->Shortcuts->Edit metadata
# to paste metadata ignoring this tweak.
exclude_fields_on_paste = []

#: Skip internet connected check
# Skip checking whether the internet is available before downloading news.
# Useful if for some reason your operating systems network checking
# facilities are not reliable (for example NetworkManager on Linux).
skip_network_check = False

#: Tab stop width in the template editor
# Sets the width of the tab stop in the template editor in "average characters".
# For example, a value of 1 results in a space with the width of one average_
↳character.
template_editor_tab_stop_width = 4

#: Value for undefined numbers when sorting
# Sets the value to use for undefined numbers when sorting.
# For example, the value -10 sorts undefined numbers as if they were set to -10.
# Use 'maximum' for the largest possible number. Use 'minimum' for the smallest
# possible number. Quotes are optional if entering a number.
# Examples:
#   value_for_undefined_numbers_when_sorting = -100
#   value_for_undefined_numbers_when_sorting = '2'
#   value_for_undefined_numbers_when_sorting = -0.01
#   value_for_undefined_numbers_when_sorting = 'minimum'
#   value_for_undefined_numbers_when_sorting = 'maximum'
value_for_undefined_numbers_when_sorting = 0

#: Allow template database functions in composite columns
# If True then the template database functions book_values() and book_count()
# can be used in composite custom columns. Note: setting this tweak to True and
# using these functions in composites can be very slow.
# Default: False
allow_template_database_functions_in_composites = False

#: Change the programs that are run when opening files/URLs
# By default, calibre passes URLs to the operating system to open using
# whatever default programs are configured there. Here you can override
# that by specifying the program to use, per URL type. For local files,
# the type is "file" and for web links it is "http*". For example:
# openers_by_scheme = { "http*": "firefox %u" } will make calibre run Firefox
# for https://whatever URLs. %u is replaced by the URL to be opened. The scheme
# takes a glob pattern allowing a single entry to match multiple URL types.
openers_by_scheme = {}

```

12.4 Sovrascrivere icone, modelli, eccetera

Nota: calibre supporta nativamente temi per le icone, ci sono molti temi disponibili per calibre che puoi utilizzare andando in *Preferenze* → *Interfaccia* → *Aspetto* → *Cambia il tema icone*. È preferibile usare i temi icone rispetto a sostituire le icone individualmente.

calibre ti permette di sostituire le risorse statiche come icone, JavaScript e modelli per la sovracopertina dei metadati, cataloghi, ecc. con versioni personalizzate di tuo gusto. Tutte le risorse statiche sono conservate nella sottocartella *resources* nella posizione di installazione di calibre. Su Windows, questa è solitamente `C:\Program Files\Calibre2\app\resources`. Su Linux, se stai usando il file di installazione binario del sito web di calibre, sarà `/opt/calibre/resources`. Questi percorsi possono cambiare a seconda di dove scegli di installare calibre.

Dovresti evitare di cambiare i file della cartella risorse, perché i tuoi cambiamenti verranno sovrascritti la prossima volta che aggiorni calibre. Piuttosto, vai in *Preferenze* → *Avanzate* → *Varie* e fai clic su *Apri cartella di configurazione di calibre*. In questa cartella di configurazione, crea una sottocartella chiamata *resources* e metti dentro i file che vuoi sostituire. Metti i file nelle sottocartelle appropriate, per esempio le immagini in *resources/images*, ecc. calibre preferirà automaticamente i tuoi file personalizzati ai file integrati al successivo avvio dell'applicazione.

Per esempio, se desiderassi cambiare l'icona per l'azione *Rimuovi libri*, guarderesti per prima cosa nella cartella *resources* integrata e vedresti che il file che ti interessa è *resources/images/remove_books.png*. Supponendo che tu abbia un'icona alternativa in formato PNG chiamata *mio_rimuovi_libri.png*, la dovresti salvare nella cartella di configurazione come *resources/images/remove_books.png*. Tutte le icone usate dall'interfaccia utente di calibre si trovano in *resources/images* e nelle sue sottocartelle. Un file messo in questa posizione ha una priorità ancora maggiore di quella di un tema icone personalizzato.

12.5 Creazione di un tema personale per le icone di calibre

Se hai creato una splendida collezione di icone e desideri condividerle con gli altri utenti di calibre attraverso il sistema dei temi icone integrato in calibre, puoi facilmente impacchettare le tue icone in un tema. Per fare ciò, vai in *Preferenze* → *Varie* → *Crea un tema icone* e seleziona la cartella in cui hai messo le icone. Quindi compila i metadati del tema e clicca su OK. Ciò creerà un file ZIP contenente le icone del tema. Puoi caricarlo nel forum dedicato a calibre su [Mobileread](https://www.mobileread.com/forums/forumdisplay.php?f=166)¹⁰⁶, dopodiché io renderò disponibile il tuo tema grazie al sistema dei temi icone integrato in calibre. Se vuoi collaudare il tuo tema, ricorda di rimuovere le immagini dalla cartella *resources/images*, in modo che vengano utilizzate le sue icone.

Con calibre 6, puoi avere icone personalizzate sia per il tema chiaro che per quello scuro. Basta creare due versioni dell'icona e aggiungere ai nomi del file il suffisso `-for-dark-theme` e `-for-light-theme`. Per esempio, `modified-for-dark-theme.png` e `modified-for-light-theme.png`. calibre userà automaticamente l'icona appropriata in base al tema in uso.

¹⁰⁶ <https://www.mobileread.com/forums/forumdisplay.php?f=166>

12.6 Personalizzazione di calibre con i plugin

calibre ha una struttura estremamente modulare. Quasi tutte le funzionalità di calibre agiscono sotto forma di plugin. Dei plugin sono usati per le conversioni, per scaricare le notizie (sebbene questi siano chiamati ricette), per molte componenti dell'interfaccia utente, per connettersi a svariati dispositivi, per elaborare i file aggiunti a calibre e così via. Puoi ottenere una lista completa dei plugin inclusi in calibre andando in *Preferenze* → *Avanzate* → *Plugin*.

Puoi scrivere i tuoi plugin per personalizzare ed estendere il comportamento di calibre. L'architettura dei plugin in calibre è molto semplice, vedi la guida *Scrivere i propri plugin per estendere le funzionalità di calibre* (pagina 216).

Una volta creato un plugin, puoi caricarlo nel forum dei plugin di calibre su [Mobileread](https://www.mobileread.com/forums/forumdisplay.php?f=237)¹⁰⁷ e sarà reso disponibile attraverso il gestore degli aggiornamenti incluso in calibre.

¹⁰⁷ <https://www.mobileread.com/forums/forumdisplay.php?f=237>

Interfaccia a riga di comando

```
kovid giskard ~/work/libprs500/src/libprs500/manual $
```

Nota: In macOS, gli strumenti da linea di comando si trovano dentro il bundle di calibre, per esempio, se hai installato calibre in /Applications gli strumenti si trovano in /Applications/calibre.app/Contents/MacOS/.

13.1 Comandi documentati

13.1.1 calibre

```
calibre [opzioni] [percorso_dell_ebook o url di calibre ...]
```

Avvia l'interfaccia grafica principale di **calibre** e facoltativamente aggiunge l'e-book in percorso_dell_ebook al database. Puoi anche inserire degli URL di **calibre** per eseguire varie azioni differenti dalla semplice aggiunta di libri. Per esempio:

calibre://view-book/biblioteca_di_prova/1842/epub

Aprirà il libro con id 1842 nel formato EPUB dalla biblioteca «biblioteca_di_prova» nel Lettore e-book di **calibre**. I nomi delle biblioteche sono i nomi delle cartelle delle biblioteche in cui gli spazi sono sostituiti da trattini bassi. Una descrizione completa delle varie azioni basate su URL si trova nel manuale utente.

Ogni qualvolta vengono passati argomenti con degli spazi a **calibre**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--detach

Scollega dal terminale di controllo, se presente (solo su Linux)

--help, -h

mostra questo messaggio di aiuto ed esci

--ignore-plugins

Ignora i plugin personalizzati, utile se hai installato un plugin che impedisce l'avvio di calibre

--no-update-check

Non controllare gli aggiornamenti

--shutdown-running-calibre, -s

A causa di un'istanza calibre in esecuzione, se del caso, da arrestare. Nota che se ci sono lavori in esecuzione, essi saranno interrotti silenziosamente, perciò usala con attenzione.

--start-in-tray

Avvia minimizzato nella area di notifica

--verbose, -v

Ignorato, non utilizzare. Presente solo per motivi di eredità da versioni precedenti del software.

--version

mostra la versione del programma ed esci

--with-library

Usa la biblioteca collocata nel percorso specificato.

13.1.2 calibre-customize

opzioni calibre-customize

Personalizza calibre con l'utilizzo di plugin esterni

Ogni qualvolta vengono passati argomenti con degli spazi a **calibre-customize**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--add-plugin, -a

Aggiungi un plugin specificando il percorso al file ZIP che lo contiene

--build-plugin, -b

Per gli sviluppatori di plugin: percorso della cartella dove state sviluppando il plugin. Questo comando creerà automaticamente un archivio ZIP del plugin e lo aggiornerà in calibre.

--customize-plugin

Personalizza un plugin. Specifica il nome del plugin e la stringa di personalizzazione separati da una virgola.

--disable-plugin

Disabilita il plugin nominato

--enable-plugin

Abilita il plugin nominato

--help, -h

mostra questo messaggio di aiuto ed esci

--list-plugins, -l

Elenca tutti i plugin installati

--remove-plugin, -r

Rimuovi un plugin scelto per nome. Non ha effetto sui plugin di sistema

--version

mostra la versione del programma ed esci

13.1.3 calibre-debug

```
calibre-debug [opzioni]
```

Diverse interfacce da linea di comando utili per il debug di calibre. Senza opzioni, questo comando avvia un interprete Python integrato. Puoi anche avviare l'interfaccia principale di calibre, il lettore di e-book di calibre e l'editor di calibre in modalità debug.

Contiene anche interfacce per diverse parti di calibre che non hanno strumenti da linea di comando specifici, come la riduzione dei caratteri, lo strumento di comparazione e-book, ecc.

Puoi anche usare **calibre-debug** per avviare script stand-alone. Per farlo segui queste istruzioni:

```
calibre-debug -e myscript.py -- --option1 --option2 file1 file2 ...
```

Tutto ciò che segue i `--` è passato allo script. Puoi anche usare **calibre-debug** come shabang negli script, in questo modo:

```
#!/usr/bin/env -S calibre-debug -e -- -
```

Ogni qualvolta vengono passati argomenti con degli spazi a **calibre-debug**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]**--add-simple-plugin**

Aggiungi un plugin semplice (es. un plugin composto dal solo file .py), specificando il percorso del file py contenente il codice del plugin.

--command, -c

Esegui codice Python.

--debug-device-driver, -d

Debug riconoscimento dispositivi

--default-programs

Imposta/rimuovi Calibre dai programmi predefiniti di Windows `--default-programs` (pagina 297) = (register|unregister)

--diff

Esegui lo strumento diff di calibre. Ad esempio: `calibre-debug --diff` (pagina 297) file1 file2

--edit-book

Avvia lo strumento "Modifica libro" in modalità debug

--exec-file, -e

Esegui codice Python nel file.

--explode-book, -x

Frantuma il libro nella cartella specificata. Utilizzo: -x file.epub cartella_di_output Esporta il libro come collezione di file HTML e metadati, che puoi modificare usando strumenti pensati per l'HTML. Funziona con i file EPUB, AZW3, HTMLZ e DOCX.

--export-all-calibre-data

Esporta tutti i dati di calibre (libri/impostazioni/plugin). Normalmente, ti verrà chiesta la cartella di esportazione e le librerie da esportare. Puoi anche specificarli come argomenti della riga di comando per saltare le domande. Utilizza percorsi assoluti per la cartella di esportazione e le librerie. La parola chiave speciale "all" può essere utilizzata per esportare tutte le librerie. Esempi: calibre-debug *--export-all-calibre-data* (pagina 298) # per uso interattivo calibre-debug *--export-all-calibre-data* (pagina 298) /percorso/per/vuoto/esporta/cartella/percorso/per/libreria/cartella1 /percorso/per/libreria2 calibre-debug *--export-all-calibre-data* (pagina 298) /export/folder all # esporta tutte le librerie conosciute

--fix-multiprocessing

Per uso interno.

--gui, -g

Avvia l'interfaccia con il debug abilitato. Il risultato del debug è stampato su stdout e stderr.

--gui-debug

Avvia l'interfaccia con una console di debug, registrando i log in un percorso specificato. Solo per uso interno, utilizza l'opzione -g per avviare l'interfaccia nella modalità debug

--help, -h

mostra questo messaggio di aiuto ed esci

--implode-book, -i

Ricomponi un libro frantumato precedentemente. Utilizzo: -i cartella_di_output file.epub Importa il libro dai file in cartella_di_output che devono essere stati creati da una precedente chiamata a *--explode-book* (pagina 298). Assicurati di specificare lo stesso tipo di file utilizzato per la frantumazione.

--import-calibre-data

Importa i dati esportati in precedenza

--inspect-mobi, -m

Analizza i file MOBI nel percorso specifico

--paths

Fornisce i percorsi necessari per configurare l'ambiente di calibre

--run-plugin, -r

Esegui un plugin con un'interfaccia a linea di comando. Per esempio: calibre-debug -r "Nome plugin" -- file1 --opzione1 Tutto ciò che segue il -- sarà passato al plugin come argomento.

--run-test, -t

Esegui i(l) test nominato(i). Usa il valore speciale "all" per eseguirli tutti. Se il nome del test inizia con un punto, è interpretato come nome di un modulo. Se il nome del test inizia con @, è interpretato come nome di una categoria.

--run-without-debug

Non eseguire con il flag DEBUG attivato

--shutdown-running-calibre, -s

A causa di un'istanza calibre in esecuzione, se del caso, da arrestare. Nota che se ci sono lavori in esecuzione, essi saranno interrotti silenziosamente, perciò usala con attenzione.

--subset-font, -f

Crea un sottoinsieme del carattere specificato. Usa -- dopo questa opzione per passare opzione al programma di riduzione caratteri.

--test-build

Prova moduli binari in costruzione

--version

mostra la versione del programma ed esci

--viewer, -w

Esegui lettore di e-book in modalità debug

13.1.4 calibre-server

```
calibre-server [opzioni] [percorso cartella della biblioteca...]
```

Avvia il server dei contenuti di calibre. Il server dei contenuti di calibre rende visibili le tue biblioteche di calibre su internet. Puoi specificare il percorso delle cartelle della biblioteca come argomenti di **calibre-server**. Se non specifichi alcun percorso, verranno usate tutte le biblioteche che il programma principale di calibre conosce.

Ogni qualvolta vengono passati argomenti con degli spazi a **calibre-server**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--access-log

Percorso del file di registro degli accessi. Questo registro contiene informazioni sui client che si connettono al server e sulle loro richieste. Per impostazione predefinita non viene tenuto alcun registro degli accessi.

--ajax-timeout

Tempo (in secondi) di attesa di risposta dal server dopo una richiesta.

--auth-mode

Scegli il tipo di autenticazione adoperato. Imposta la modalità di autenticazione HTTP usata dal server. Imposta su "basic" se terrai questo server dietro un proxy SSL. Altrimenti lascia su "auto", che utilizzerà "basic" se SSL è configurato e "digest" altrimenti.

--auto-reload

Ricarica automaticamente il server quando il codice sorgente è cambiato. Utile per lo sviluppo. Occorre anche specificare un valore piccolo per il timeout di spegnimento.

--ban-after

Numero di login falliti necessari per il ban. Il numero di login falliti dopo i quali un indirizzo IP viene bloccato

--ban-for

Blocca gli indirizzi IP che hanno sbagliato più volte l'accesso. Blocca temporaneamente l'accesso agli indirizzi IP che hanno fallito ripetutamente il login per il numero di minuti specificato. Utile per prevenire tentativi di indovinare la password. Se impostato su zero non viene eseguito alcun ban.

--book-list-mode

Scegli la modalità predefinita per l'elenco dei libri. Imposta la modalità predefinita per l'elenco dei libri che sarà usata per i nuovi utenti. I singoli utenti possono cambiare l'impostazione predefinita nelle loro impostazioni personali. Come impostazione predefinita è usata la griglia delle copertine.

--compress-min-size

Dimensione minima per la quale le risposte utilizzano la compressione dei dati (in byte).

--custom-list-template

Percorso di un file JSON contenente un modello per la modalità di elenco personalizzato dei libri. Il modo più semplice per creare un modello del genere è andare in Preferenze-> Condivisione via rete-> Modello elenco dei libri in calibre, creare il modello ed esportarlo.

--daemonize

Esegui processo in background come daemon (solo Linux).

--displayed-fields

Limita i campi definiti dall'utente mostrati. Lista separata da virgole di campi di metadati definiti dall'utente che saranno mostrati dal server dei contenuti nelle interfacce /opds e /mobile. Se questa opzione è specificata, ogni campo assente da questa lista non sarà mostrato. Per esempio: my_rating,my_tags

--enable-allow-socket-preallocation, --disable-allow-socket-preallocation

Preallocazione socket, per esempio, attraverso l'attivazione socket di systemd. Per impostazione predefinita, questa opzione è abilitata.

--enable-auth, --disable-auth

Autenticazione con password per accedere a questo server. Normalmente il server è privo di restrizioni e permette a tutti di accedervi. Con questa opzione puoi limitare l'accesso a utenti preimpostati. Per impostazione predefinita, questa opzione è disabilitata.

--enable-fallback-to-detected-interface,

--disable-fallback-to-detected-interface

Ripiega su interfaccia rilevata automaticamente. Se per qualche ragione il server non è in grado di collegarsi all'interfaccia specificata nell'opzione listen_on, proverà a rilevare un'interfaccia che si connette al mondo esterno e a collegarsi ad essa. Per impostazione predefinita, questa opzione è abilitata.

--enable-local-write, --disable-local-write

Consenti alle connessioni locali non autenticate di apportare modifiche. Di norma, se non abiliti l'autenticazione, il server opera in modalità di sola lettura, in modo da non permettere ad utenti anonimi di effettuare cambiamenti alle tue biblioteche di calibre. Questa opzione permette a tutti coloro che si connettono dallo stesso computer in cui si trova il server di effettuare modifiche. Utile se vuoi far operare il server senza autenticazione ma vuoi anche usare calibredb per effettuare cambiamenti alle tue biblioteche di calibre. Tieni presente che abilitando questa opzione tutti i programmi in esecuzione sul computer possono effettuare modifiche alle tue biblioteche di calibre. Per impostazione predefinita, questa opzione è disabilitata.

--enable-log-not-found, --disable-log-not-found

Registra le richieste HTTP 404 (Non Trovato). Di norma, il server registra tutte le richieste HTTP dirette a risorse non trovate. Questo può generare una quantità esagerata di avvisi se il tuo server è preso di mira da bot. Usa questa opzione per disattivarli. Per impostazione predefinita, questa opzione è abilitata.

--enable-use-bonjour, --disable-use-bonjour

Pubblicizza i feed OPDS con Bonjour. Pubblicizza il feed OPDS attraverso il servizio Bonjour, così da permettere alle app di lettura basate su OPDS di rilevare e di connettersi al server automaticamente. Per impostazione predefinita, questa opzione è abilitata.

--enable-use-sendfile, --disable-use-sendfile

Trasferimenti di file in zero copy per prestazioni migliorate. Questa opzione abilita i trasferimenti di file in zero copy nel kernel per inviare file via rete, migliorando le prestazioni. Questo, tuttavia, può essere causa di trasferimenti di file corrotti in alcuni file system malfunzionanti. Se avvengono trasferimenti corrotti, disattiva questa opzione. Per impostazione predefinita, questa opzione è abilitata.

--help, -h

mostra questo messaggio di aiuto ed esci

--ignored-fields

Campi di metadati dell'utente da ignorare. Lista separata da virgole di campi di metadati definiti dall'utente che non saranno mostrati dal server dei contenuti nelle interfacce /opds e /mobile. Per esempio: my_rating,my_tags

--listen-on

L'interfaccia su cui ascoltare in attesa di connessioni. L'impostazione predefinita è di ascoltare su tutte le interfacce IPv4 disponibili. Puoi cambiare questa opzione in, per esempio, "127.0.0.1" per ascoltare solo per connessioni dalla macchina locale, o in "::" per ascoltare tutte le connessioni IPv6 e IPv4 in ingresso.

--log

Percorso del file di registro del server. Questo registro contiene informazioni ed errori del server, non i registri di accesso. Per impostazione predefinita viene scritto in stdout.

--manage-users

Gestisci il database degli utenti abilitati a collegarsi a questo server. Puoi usarlo in modalità automatica aggiungendo un `-`. Vedi `calibre-server --manage-users` (pagina 301) `-- help` per dettagli. Vedi anche l'opzione `--userdb` (pagina 302).

--max-header-line-size

Dimensione massima del singolo header HTTP (in KB).

--max-job-time

Tempo massimo dei processi di lavoro. Tempo massimo di esecuzione per i processi di lavoro (in minuti). Imposta a 0 per non dare limiti.

--max-jobs

Numero massimo di processi di lavoro. I processi di lavoro vengono avviati in base alle necessità e utilizzati per lavori di grandi dimensioni come la preparazione di un libro per la visualizzazione, l'aggiunta di libri, la conversione, ecc. Normalmente, il numero massimo di tali processi si basa sul numero di core della CPU. È possibile controllarlo con questa impostazione.

--max-log-size

Dimensione massima file di log (in MB). La massima dimensione dei registri (log) generati dal server. Quando un file di registro supera questa dimensione, è automaticamente ruotato. Imposta a zero per disattivare la rotazione dei registri.

--max-opds-items

Numero massimo di libri nei feed OPDS. Il massimo numero di libri che restituirà il server in un singolo feed di acquisizione OPDS.

--max-opds-ungrouped-items

Numero massimo di elementi non raggruppati nei feed OPDS. Raggruppa gli elementi in categorie come autore/tag in base alla prima lettera quando il numero di elementi è maggiore di questo numero. Imposta a zero per disattivare.

--max-request-body-size

Dimensione massima consentita per file caricati sul server (in MB).

--num-per-page

Numero di libri da mostrare in una pagina singola. Il numero di libri da mostrare in una singola pagina del browser.

--pidfile

Scrivi il PID di processo nel file specificato

--port

La porta di ascolto per la connessione.

--search-the-net-urls

Percorso di un file JSON contenente degli URL per la funzione "Cerca in internet". Il modo più semplice per creare un file del genere è andare in Preferenze-> Condivisione via rete-> Cerca in internet in calibre, creare gli URL ed esportarli.

--shutdown-timeout

Tempo totale in secondi da aspettare per un arresto pulito.

--ssl-certfile

Percorso del file del certificato SSL.

--ssl-keyfile

Percorso del file della chiave privata SSL.

--timeout

Tempo (in secondi) dopo il quale una connessione inattiva viene chiusa.

--trusted-ips

Consenti a connessioni non autenticate da indirizzi IP specifici di apportare modifiche. Di norma, se non abiliti l'autenticazione, il server opera in modalità di sola lettura, in modo da non permettere ad utenti anonimi di effettuare cambiamenti alle tue biblioteche di calibre. Questa opzione permette a tutti coloro che si connettono dall'indirizzo IP specificato di effettuare modifiche. Deve essere un elenco separato da virgole di reti o di indirizzi specifici. Utile se vuoi far operare il server senza autenticazione ma vuoi anche usare calibredb per effettuare cambiamenti alle tue biblioteche di calibre. Tieni presente che abilitando questa opzione tutti coloro che si connettono dagli indirizzi IP specificati possono effettuare modifiche alle tue biblioteche di calibre.

--url-prefix

Un prefisso da anteporre a tutti gli URL. Utile se desideri far operare questo server dietro un proxy inverso. Per esempio, usa /calibre come prefisso dell'URL.

--userdb

Percorso del database degli utenti da usare per l'autenticazione. Il database è un file SQL. Per crearlo usa `--manage-users` (pagina 301). Puoi saperne di più sulla gestione degli utenti qui: <https://manual.calibre-ebook.com/it/server.html#managing-user-accounts-from-the-command-line-only>

--version

mostra la versione del programma ed esci

--worker-count

Numero di thread di lavoro utilizzati per elaborare le richieste.

13.1.5 calibre-smtp

```
calibre-smtp [opzioni] [da a testo]
```

Invia posta usando il protocollo SMTP. **calibre-smtp** ha due modalità di funzionamento. Nella modalità componi vanno specificati da, a e testo, usati per costruire e inviare un messaggio email. Nella modalità filtra, **calibre-smtp** legge un messaggio email completo da STDIN e lo invia.

testo è il corpo del messaggio email. Se testo non viene specificato, un messaggio email completo è letto da STDIN. da è l'indirizzo email del mittente, a l'indirizzo email del destinatario. Quando un'email completa viene letta da STDIN, da e a sono usati solo durante la negoziazione SMTP, le intestazioni del messaggio non sono modificate.

Ogni qualvolta vengono passati argomenti con degli spazi a **calibre-smtp**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--fork, -f

Separa il processo e consegna il messaggio in background. Se usi questa opzione dovresti usare anche **--outbox** (pagina 303) per gestire le consegne fallite.

--help, -h

mostra questo messaggio di aiuto ed esci

--localhost, -l

Nome host di localhost. Utilizzato in caso di connessione a server SMTP.

--outbox, -o

Percorso della cartella di posta dove archiviare le mail fallite.

--timeout, -t

Tempo massimo per la connessione

--verbose, -v

Aumenta la verbosità

--version

mostra la versione del programma ed esci

COMPOSE MAIL

Opzioni per comporre delle email. Ignorato se il testo non è specificato

--attachment, -a

File da allegare all' email

--subject, -s

Oggetto dell' email

SMTP RELAY

Opzione di usare un server SMTP per mandare la mail. calibre proverà ad inviare la mail direttamente a meno che `--relay` non sia specificato

--cafile

Percorso di un file contenente certificati CA concatenati in formato PEM, usati per verificare il certificato del server quando viene usato TLS. Per impostazione predefinita, vengono usati i certificati CA del sistema.

--dont-verify-server-certificate

Non verificare il certificato del server quando ti connetti usando TLS. Questo era il comportamento predefinito nelle versioni di calibre precedenti alla 3.27. Se usi un relay con un certificato autofirmato o comunque invalido, puoi usare questa opzione per ripristinare il comportamento pre-3.27

--encryption-method, -e

Metodi crittografici da utilizzare per la connessione al server di inoltro. Le scelte possibili sono TLS, SSL e NONE. Il valore predefinito è TLS. ATTENZIONE: Scegliere NONE è decisamente non sicuro

--password, -p

Password per l'inoltro

--port

Porta per collegarsi al server di inoltro. Il valore predefinito è 465, se il metodo di cifratura è SSL, altrimenti 25.

--relay, -r

Un server SMTP di inoltro da utilizzare per inviare posta.

--username, -u

Nome utente per l'inoltro

13.1.6 calibredb

```
calibredb command [opzioni] [argomenti]
```

calibredb è l'interfaccia da linea di comando che agisce sul database di calibre. Possiede molti sotto-comandi, descritti più sotto

calibredb può essere usato per manipolare un database di calibre identificato da un percorso oppure da un *Server dei contenuti* di calibre in esecuzione sulla macchina locale o su internet. Puoi avviare un *Server dei contenuti* di calibre con il comando **calibre-server** oppure facendo clic nell'interfaccia principale di calibre *Connetti/condividi* → *Avvia server dei contenuti*. Dato che **calibredb** può modificare le tue biblioteche di calibre, devi prima impostare un'autenticazione per il server. Ci sono due modi per farlo:

- Se intendi connetterti solamente a un server in esecuzione sul computer stesso,

puoi semplicemente usare l'opzione `--enable-local-write` del server dei contenuti, per permettere ad ogni programma, calibredb incluso, in esecuzione sul computer locale di modificare i tuoi dati di calibre. Se il server è avviato dall'interfaccia principale di calibre, questa opzione si trova in *Preferenze* → *Condivisione via rete* → *Avanzate*.

- Se vuoi invece abilitare l'accesso via internet, dovresti impostare

degli utenti per il server e usare le opzioni `--username` (pagina 306) e `--password` (pagina 306) con **calibredb** per permettergli di accedere. Puoi impostare l'autenticazione utente per **calibre-server** usando l'opzione `--enable-auth` e usando `--manage-users` per creare gli utenti. Se hai avviato il server dall'interfaccia principale di calibre, usa *Preferenze* → *Condivisione via rete* → *Richiedi nome utente/password*.

Per connetterti a un server dei contenuti in esecuzione, inserisci l'URL del server dopo l'opzione `--with-library` (pagina 305), vedi la documentazione dell'opzione per dettagli ed esempi.

- *OPZIONI GLOBALI* (pagina 305)
- *list* (pagina 306)
- *add* (pagina 307)
 - *AGGIUNTA DA CARTELLE* (pagina 308)
- *remove* (pagina 308)
- *add_format* (pagina 309)
- *remove_format* (pagina 309)
- *show_metadata* (pagina 309)
- *set_metadata* (pagina 309)
- *export* (pagina 310)
- *catalog* (pagina 311)
 - *EPUB OPZIONI* (pagina 311)
- *saved_searches* (pagina 313)
- *add_custom_column* (pagina 313)
- *custom_columns* (pagina 314)
- *remove_custom_column* (pagina 314)
- *set_custom* (pagina 314)
- *restore_database* (pagina 315)
- *check_library* (pagina 315)
- *list_categories* (pagina 315)
- *backup_metadata* (pagina 316)
- *clone* (pagina 316)
- *embed_metadata* (pagina 316)
- *search* (pagina 317)
- *fts_index* (pagina 317)
- *fts_search* (pagina 318)

OPZIONI GLOBALI

--help, -h

mostra questo messaggio di aiuto ed esci

--library-path, --with-library

Percorso della biblioteca di calibre. Per impostazione predefinita viene usato il percorso salvato nelle impostazioni. Puoi anche connetterti a un server dei contenuti di calibre ed eseguire azioni su biblioteche remote. Per fare ciò usa un URL di formato: http://nomehost:porta/#id_biblioteca per esempio <http://localhost:8080/#lamiabiblioteca>. `id_biblioteca` è l'id della biblioteca a cui ti vuoi connettere nel server dei contenuti. Puoi usare il valore speciale di `id_biblioteca` di `-` per ottenere un elenco di id biblioteca disponibili nel server. Per dettagli su come imposta-

re l'accesso attraverso un server dei contenuti, vedi <https://manual.calibre-ebook.com/it/generated/it/calibredb.html>.

--password

La password per connettersi a un server dei contenuti di calibre. Per leggere la password dallo standard input, usa il valore speciale: <stdin>. Per leggere la password da un file, usa <f:/path/to/file> (ovvero <f: followed by the full path to the file and a trailing >). Le parentesi angolari di cui sopra sono richieste, ricordati di anteporre un carattere di escape o di usare le virgolette nel tuo terminale.

--timeout

Il timeout, in secondi, della connessione a una biblioteca di calibre in rete. Il tempo predefinito è di due minuti.

--username

Nome utente per connettersi a un server di contenuto calibre

--version

mostra la versione del programma ed esci

list

```
calibredb list [opzioni]
```

Elenca i libri disponibili nel database di calibre.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--ascending

Ordina i risultati in ordine crescente

--fields, -f

I campi da mostrare quando sono elencati i libri del database. Dovrebbe essere una lista separata da virgole di campi. Campi disponibili: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid. Valore predefinito: %predefinito. Il campo speciale "all" può essere usato per selezionare tutti i campi. Oltre ai campi integrati di cui sopra, sono disponibili anche campi personalizzati scritti come *nome_campo; per esempio, per un campo personalizzato #valutazione, usa il nome: *rating

--for-machine

Genera output in formato JSON, più adatto all'elaborazione automatica. Le opzioni di larghezza e separazione di linea verranno ignorate.

--limit

Il numero massimo di risultati da visualizzare. Predefinito: tutti

--line-width, -w

La larghezza massima di una singola riga nell'output. L'impostazione predefinita è quella di rilevare la dimensione dello schermo.

--prefix

Il prefisso per tutti i percorsi. L'impostazione predefinita è il percorso alla cartella della biblioteca.

--search, -s

Filtra i risultati con una ricerca. Per il formato della ricerca, vedere la documentazione relativa alla ricerca nel manuale utente. L'opzione predefinita è quella di non filtrare.

--separator

La stringa usata per separare i campi. Predefinita: uno spazio.

--sort-by

Il campo da usare per l'ordinamento dei risultati. Puoi specificare più campi usando la virgola come separatore. Campi disponibili: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid Predefinito: id

--template

Il modello da eseguire se "template" è nell'elenco dei campi. Predefinito: Nessuno

--template_file, -t

Percorso di un file contenente il modello da eseguire se "template" è nell'elenco dei campi. Predefinito: Nessuno

--template_heading

Intestazione della colonna del modello. Predefinito: pre%definito. Questa opzione viene ignorata se l'opzione `--for-machine` (pagina 306) è impostata

add

```
calibredb add [opzioni] file1 file2 file3 ...
```

Aggiungi come libri i file specificati nel database. Puoi anche specificare delle cartelle, vedi le opzioni riferite alle cartelle qui sotto.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--authors, -a

Imposta gli autori dei libri aggiunti

--automerge, -m

Se vengono trovati libri con titoli e autori simili, unisci automaticamente i formati (file) in arrivo con voci di libro già esistenti. Un valore di "ignore" significa che i formati duplicati vengono scartati. Un valore di "overwrite" significa che i formati duplicati nella biblioteca vengono sovrascritti dai nuovi file aggiunti. Un valore di "new_record" significa che i formati duplicati sono inseriti in una nuova voce libro.

--cover, -c

Percorso della copertina da utilizzare per il libro aggiunto

--duplicates, -d

Aggiungi libri al database anche se esistono già. Il confronto è basato su titoli e autori dei libri. Nota che l'opzione `--automerge` (pagina 307) ha la precedenza.

--empty, -e

Aggiungi un libro vuoto (un libro senza formati)

--identifier, -I

Imposta gli identificatori per questo libro, ad esempio: -I asin:XXX -I isbn:YYY

--isbn, -i

Imposta l'ISBN dei libri aggiunti

--languages, -l

Un elenco separato da virgole di lingue (meglio utilizzare i codici di lingua ISO639 per evitare che alcuni nomi di lingue non siano riconosciuti)

--series, -s

Imposta le serie dei libri aggiunti

--series-index, -S

Imposta il numero della serie dei libri aggiunti

--tags, -T

Imposta i tag dei libri aggiunti

--title, -t

Imposta il titolo dei libri aggiunti

AGGIUNTA DA CARTELLE

Opzioni per controllare l'aggiunta di libri dalle cartelle. Per impostazione predefinita solo i file con estensioni di e-book note sono aggiunti.

--add

Uno schema per il nome del file (glob pattern), i file che corrispondono a questo schema saranno aggiunti durante la scansione delle cartelle, anche se il loro tipo di file non è quello tipico di un e-book. Può essere specificato più volte per schemi multipli.

--ignore

Uno schema per il nome del file (glob pattern), i file che corrispondono a questo schema saranno ignorati durante la scansione delle cartelle. Può essere specificato più volte per schemi multipli. Ad esempio: *.pdf escluderà tutti i file PDF

--one-book-per-directory, -1

Assumi che ogni cartella abbia un solo libro "logico" e che tutti i file in essa contenuti siano differenti formati e-book dello stesso libro

--recurse, -r

Controlla cartelle ricorsivamente

remove

```
calibredb remove id
```

Rimuove dal database i libri identificati dagli id. Gli id dovrebbero essere una lista separata da virgole di numeri id (puoi ottenere i numeri id usando il comando search). Per esempio, 23,34,57-85 (quando si specifica un intervallo, l'ultimo numero dell'intervallo è escluso).

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--permanent

Non usare il Cestino

add_format

```
calibredb add_format [options] id ebook_file
```

Aggiungi l'e-book in `ebook_file` ai formati disponibili per il libro identificato per `id`. Puoi trovare l'id usando il comando `search`, se il formato è già presente verrà sostituito a meno che l'opzione "non sostituire" non sia stata specificata.

Ogni qualvolta vengono passati argomenti con degli spazi a `calibredb`, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--as-extra-data-file

Aggiungi il file come un file di dati aggiuntivo, non come formato di e-book

--dont-replace

Non sostituire il formato se esiste già

remove_format

```
calibredb remove_format [opzioni] id fmt
```

Rimuovi il formato `fmt` identificato da `id` dal libro logico. Puoi ottenere l'id usando il comando `search`. `fmt` deve essere un'estensione di file come LRF, TXT o EPUB. Se il libro logico non ha un `fmt` disponibile, non fa niente.

Ogni qualvolta vengono passati argomenti con degli spazi a `calibredb`, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

show_metadata

```
calibredb show_metadata [opzioni] id
```

Mostra i metadati salvati nel database di calibre per il libro identificato da `id`. `id` è un numero identificativo del comando di ricerca.

Ogni qualvolta vengono passati argomenti con degli spazi a `calibredb`, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--as-opf

Stampa i metadati in formato OPF (XML)

set_metadata

```
calibredb set_metadata [opzioni] book_id [/path/to/metadata.opf]
```

Imposta i metadati memorizzati nel database di calibre per il libro identificato da `book_id` dal file OPF `metadata.opf`. `book_id` è un numero di identificazione del libro dal comando di ricerca. Puoi farti un'idea del formato OPF usando `--as-opf` passa al comando `show_metadata`. Puoi anche impostare i metadati di singoli campi con l'opzione `--field`. Se usi l'opzione `--field`, lì non è necessario specificare un file OPF.

Ogni qualvolta vengono passati argomenti con degli spazi a `calibredb`, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--field, -f

Il campo da impostare. Il formato è nome_campo:valore, per esempio: `--field` (pagina 309) tags:tag1,tag2. Usa `--list-fields` (pagina 310) per ottenere una lista di tutti i nomi dei campi. Puoi specificare questa opzione più volte per impostare più di un campo. Nota bene: per le lingue devi usare i codici lingua ISO639 (p. es. en per inglese, it per italiano e così via). Per gli identificatori, la sintassi è `--field` (pagina 309) identifiers:isbn:XXXX,doi:YYYYY. Per campi booleani (sì/no) usa true e false oppure yes e no.

--list-fields, -l

Elenca i nomi dei campi di metadati che possono essere utilizzati con l'opzione `--field` (pagina 309)

export

```
calibredb export [opzioni] id
```

Esporta i libri specificati con id (una lista separata da virgole) su disco. L'operazione di esportazione salva tutti i formati del libro, la copertina e i metadati (in un file OPF). Anche tutti i file di dati aggiuntivi associati al libro vengono salvati. Puoi ottenere i numeri di id con il comando search.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--all

Esporta tutti i libri del database, ignorando la lista di id.

--dont-asciiize

Fai convertire a calibre tutti i caratteri non inglesi nei loro equivalenti inglesi nei nomi di file. Utile se si sta salvando in un vecchio filesystem che non ha il pieno supporto ai nomi di file Unicode. Selezionando questa opzione verrà disattivata questa funzione.

--dont-save-cover

Normalmente calibre salva le copertine in un file separato insieme ai file degli e-book. Selezionando questa opzione verrà disattivata questa funzione.

--dont-save-extra-files

Salvare tutti i file di dati associati al libro quando si salva il libro. Selezionando questa opzione verrà disattivata questa funzione.

--dont-update-metadata

Normalmente calibre aggiorna i metadati nei file salvati usando quelli che si trovano nella biblioteca di calibre. Questo rende il salvataggio più lento. Selezionando questa opzione verrà disattivata questa funzione.

--dont-write-opf

Normalmente calibre scrive i metadati in documenti OPF separati assieme ai file contenenti i libri. Selezionando questa opzione verrà disattivata questa funzione.

--formats

Elenco di formati separati da virgole da salvare per ogni libro. In modo predefinito, saranno salvati tutti i formati disponibili.

--progress

Avanzamento report

--replace-whitespace

Sostituisci gli spazi con trattini bassi.

--single-dir

Esporta tutti i libri in una singola cartella

--template

Lo schema che controlla i nomi di file e l'albero delle cartelle dei file salvati. Il valore predefinito è "{author_sort}/{title}/{title} - {authors}" che salva i libri in una sottocartella per autore mentre i nomi di file contengono titolo e autore. I controlli disponibili sono: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt

Il formato di visualizzazione delle date. %d - giorno, %b - mese, %m - numero mese, %Y - anno. Predefinito: %b, %Y

--to-dir

Esporta i libri nella cartella specificata. Il valore predefinito è .

--to-lowercase

Converti i percorsi in lettere minuscole.

catalog

```
calibredb catalog /path/to/destination.(csv|epub|mobi|xml...) [options]
```

Export a **catalog** in format specified by path/to/destination extension. Options control how entries are displayed in the generated **catalog** output. Note that different **catalog** formats support different sets of options. To see the different options, specify the name of the output file and then the **--help** option.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--ids, -i

Lista degli identificativi ID per il catalogo in campi separati da virgole. Se dichiarati, **--search** (pagina 311) è ignorata. Valore predefinito: tutti

--search, -s

Filtra i risultati in base alla query di ricerca. Per il formato della query di ricerca fare riferimento alla documentazione sulla sintassi di ricerca nel manuale utente. Predefinito: nessun filtraggio

--verbose, -v

Mostra un output dettagliato. Utile per il debug

EPUB OPZIONI**--catalog-title**

Titolo del catalogo generato usato come titolo nei metadati. Valore predefinito: 'My Books' Si applica a: formati di output AZW3, EPUB, MOBI

--cross-reference-authors

Crea riferimenti incrociati nella sezione Autori per libri con autori multipli. Predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

--debug-pipeline

Salva l'output dei differenti stadi di conversione raggiunti nella cartella specificata. Utile se non si conosce in quale stadio del processo di conversione si verifica l'errore. Valore predefinito: 'None' Si applica a: formati di output AZW3, EPUB, MOBI

--exclude-genre

Regex che descrive i tag da escludere come generi. Valore predefinito: '[.]|^+\$' esclude i tag tra parentesi quadre, p. es. '[Progetto Gutenberg]', e '+', il tag predefinito per i libri letti. Si applica a: formati di output AZW3, EPUB, MOBI

--exclusion-rules

Specifica le regole usate per escludere i libri dal catalogo generato. Il modello per una regola di esclusione è o ('<nome regola>', 'Tag', '<lista di tag separati da virgola>') o ('<nome regola>', '<colonna personalizzata>', '<pattern>'). Ad Esempio: (('Libri archiviati', '#stato', 'Archiviato'),) escluderà un libro con valore 'Archiviato' nella colonna personalizzata 'stato'. Tutte le regole specificate verranno applicate. Valore predefinito: " (('Catalogs', 'Tags', 'Catalog'),)" Si applica a: formati di output AZW3, EPUB, MOBI

--generate-authors

Includi la sezione 'Autori' nel catalogo. Valore predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

--generate-descriptions

Includi la sezione 'Descrizioni' nel catalogo. Valore predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

--generate-genres

Includi la sezione 'Generi' nel catalogo. Valore predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

--generate-recently-added

Includi la sezione 'Aggiunti di recente' nel catalogo. Valore predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

--generate-series

Includi la sezione 'Serie' nel catalogo. Valore predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

--generate-titles

Includi la sezione 'Titoli' nel catalogo. Valore predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

--genre-source-field

Campo sorgente per la sezione Generi. Valore predefinito: 'Tag' Si applica a: formati di output AZW3, EPUB, MOBI

--header-note-source-field

Campo personalizzato che contiene una nota di testo da inserire nell'intestazione della Descrizione. Valore predefinito: '' Si applica a: formati di output AZW3, EPUB, MOBI

--merge-comments-rule

#<custom field>:[before|after]:[True|False] specificando: <custom field> Campo personalizzato contenente note da unire ai commenti [before|after] Posizione delle note rispetto ai commenti (prima o dopo) [True|False] - Viene inserita una riga orizzontale tra le note e i commenti Valore predefinito: '::' Si applica a: formati di output AZW3, EPUB, MOBI

--output-profile

Specifica il profilo di output. In alcuni casi un profilo di output è richiesto per ottimizzare il catalogo per il dispositivo. Ad esempio 'kindle' o 'kindle_dx' creano un Indice strutturato con Sezioni e Articoli. Valore predefinito: 'None' Si applica a: formati di output AZW3, EPUB, MOBI

--prefix-rules

Specifica le regole utilizzate per includere i prefissi che indicano i libri letti, gli elementi della lista dei desideri e altri prefissi specificati dall'utente. Il modello per una regola di prefisso è ('<nome regola>', '<campo sorgente>', '<pattern>', '<prefisso>'). Quando più regole sono definite viene applicata la prima regola che ha una corrispondenza basata sul pattern. Valore predefinito: '((('Read books', 'tags', '+', '✓ '), ('Wishlist item', 'tags', 'Wishlist', '×'))' Si applica a: formati di output AZW3, EPUB, MOBI

--preset

Usa un profilo con nome creato con l'interfaccia di costruzione catalogo. Un profilo specifica tutte le impostazioni necessarie per costruire un catalogo. Valore predefinito: 'None' Si applica a: formati di output AZW3, EPUB, MOBI

--thumb-width

Suggerimento di dimensione (in pollici) per le copertine dei libri nel catalogo. Intervallo: 1.0 - 2.0 Valore predefinito: '1.0' Si applica a: formati di output AZW3, EPUB, MOBI

--use-existing-cover

Sostituisci la copertina esistente quando generi il catalogo. Valore predefinito: 'False' Si applica a: formati di output AZW3, EPUB, MOBI

saved_searches

```
calibredb saved_searches [opzioni] (list|add|remove)
```

Gestisci le ricerche salvate conservate in questo database. Se provi ad aggiungere una ricerca con un nome già esistente, sarà sostituita.

Sintassi per aggiungere:

```
calibredb saved_searches add nome_ricerca espressione_di_ricerca
```

Sintassi per rimuovere:

```
calibredb saved_searches remove nome_ricerca
```

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

add_custom_column

```
calibredb add_custom_column [opzioni] etichetta nome tipo
```

Crea una colonna personalizzata. etichetta è il nome amichevole della macchina della colonna. Non deve contenere spazi o punteggiatura. nome è il nome amichevole della colonna. tipo è uno di: bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--display

Un dizionario di opzioni per personalizzare l'interpretazione dei dati di questa colonna. Questa è una stringa JSON. Per le colonne di enumerazione, usare **--display** (pagina 313) "{ \"enum_values\": [\"val1\", \"val2\"] }". Ci sono molte opzioni che possono entrare nella variabile di visualizzazione. Le opzioni per tipo di colonna sono: composite: composite_template, composite_sort, make_category, contains_html, use_decorations data e ora: formato_data enumerazione: enum_valori, enum_colori, uso_decorazioni int, float: numero_formato testo: is_nomi,

uso_decorazioni Il modo migliore per trovare combinazioni ammesse è quello di creare una colonna personalizzata del tipo appropriato nella GUI e poi guardare l'OPF di backup per un libro (assicurarsi che sia stato creato un nuovo OPF da quando la colonna è stata aggiunta). Vedrete il JSON per il "display" per la nuova colonna nell'OPF.

--is-multiple

Questa colonna salva i tag come dati (es. valori separati da virgole). Applicato solo se il tipo di dato è testo.

custom_columns

```
calibredb custom_columns [options]
```

Elenca le colonne personalizzate disponibili. Mostra le etichette delle colonne e gli ID.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--details, -d

Mostra i dettagli per ogni colonna

remove_custom_column

```
calibredb remove_custom_column [opzioni] etichetta
```

Rimuove la colonna personalizzata identificata dall'etichetta. Puoi vedere le colonne disponibili con il comando custom_columns.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--force, -f

Non chiedere conferma

set_custom

```
calibredb set_custom [opzioni] colonna id valore
```

Imposta il valore di una colonna personalizzata per il libro identificato da id. Puoi ottenere una lista di id usando il comando search. Puoi ottenere una lista dei nomi delle colonne personalizzate usando il comando custom_columns.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--append, -a

Se la colonna salva più valori, aggiunge i valori specificati a quelli esistenti, altrimenti li sostituisce.

restore_database

```
calibredb restore_database [opzioni]
```

Ripristina il database a partire dai metadati conservati nei file OPF in ognuna delle cartelle della biblioteca di calibre. Utile se il tuo file metadata.db si è corrotto.

ATTENZIONE: Questo comando rigenera completamente il database. Perderai tutte le ricerche salvate, le categorie utente, i pannelli di controllo, le impostazioni di conversione per singolo libro e le ricette personalizzate. I metadati recuperati saranno accurati solo quanto lo erano i file OPF.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--really-do-it, -r

Effettua realmente il ripristino. Il comando non sarà eseguito a meno che questa opzione non sia specificata.

check_library

```
calibredb check_library [opzioni]
```

Esegue alcuni controlli sul filesystem che rappresenta la biblioteca. I resoconti sono invalid_titles, extra_titles, invalid_authors, extra_authors, missing_formats, extra_formats, extra_files, missing_covers, extra_covers, failed_folders

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--csv, -c

Output in CSV

--ignore_extensions, -e

Elenco separato da virgole delle estensioni che devono essere ignorate. Predefinito: tutte

--ignore_names, -n

Elenco separato da virgole dei nomi da ignorare. Predefinito: tutti

--report, -r

Elenco separato da virgole dei resoconti. Predefinito: tutti

--vacuum-fts-db

Svuota il database di ricerca del testo completo. Questo può risultare molto lento e richiedere molta memoria, a seconda delle dimensioni del database.

list_categories

```
calibredb list_categories [options]
```

Crea un rapporto delle informazioni sulle categorie del database. Le informazioni sono equivalenti a ciò che è mostrato nel navigatore dei Tag.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--categories, -r

Elenco separato da virgole di nomi di riferimento per categorie. Predefinito: tutti

--csv, -c

Output in CSV

--dialect

Tipo di file CSV da produrre. Alternative: excel, excel-tab, unix

--item_count, -i

Visualizza solo il numero di oggetti nella categoria invece del totale per oggetto all'interno della categoria

--width, -w

La larghezza massima di una singola riga nell'output. L'impostazione predefinita è quella di rilevare la dimensione dello schermo.

backup_metadata

```
calibredb backup_metadata [opzioni]
```

Effettua il backup dei metadati presenti nel database in file OPF dedicati nella cartella di ogni libro. Normalmente questo avviene in maniera automatica ma puoi eseguire questo comando per forzare una nuova creazione dei file OPF, con l'opzione **--all**.

Nota che normalmente non è necessario farlo dato che il backup dei file OPF viene effettuato automaticamente ogni volta che i metadati cambiano.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--all

Di solito, questo comando opera solo su libri che hanno file OPF datati. Questa opzione lo applica a tutti i libri.

clone

```
calibredb clone percorso/della/nuova/biblioteca
```

Crea un **clone** della biblioteca corrente. Questo crea una nuova biblioteca, vuota, che ha tutte le stesse colonne personalizzate, biblioteche virtuali e altre impostazioni della biblioteca corrente.

La biblioteca clonata non conterrà libri. Se desideri creare un duplicato completo, che includa tutti i libri, è sufficiente usare gli strumenti del filesystem per copiare la cartella della biblioteca.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

embed_metadata

```
calibredb embed_metadata [opzioni] id_libro
```

Aggiorna i metadati dentro i file dei libri conservati nella biblioteca di calibre con i metadati del database di calibre. Normalmente i metadati sono aggiornati solo quando vengono esportati file da calibre, questo comando è utile se vuoi che i file siano aggiornati dove si trovano. Tieni presente che formati di file differenti supportano diversi livelli di metadati. Puoi usare il valore speciale "all" come id_libro per aggiornare i metadati in tutti i libri. Puoi anche specificare più id libro separati da spazi e intervalli di id separati da trattini. Per esempio: calibredb **embed_metadata** 1 2 10-15 23

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--only-formats, -f

Aggiorna i metadati solamente nei file del formato specificato. Specificalo più volte per più formati. Per impostazione predefinita vengono aggiornati tutti i formati.

search

```
calibredb search [opzioni] espressione di ricerca
```

Cerca nella biblioteca il termine di ricerca specificato, restituisce una lista separata da virgole di id libro che risultano dall'espressione di ricerca. Il formato di uscita è comodo da usare in altri comandi che accettano una lista di id in ingresso.

L'espressione di ricerca può essere qualsiasi cosa che rispetti il potente linguaggio di ricerca di calibre, ad esempio: **calibredb search** author:asimov "title:»i robot«"

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--limit, -l

Il numero massimo di risultati da fornire. Il valore predefinito è tutti i risultati.

fts_index

```
calibredb fts_index [options] enable/disable/status/reindex
```

Controlla il processo di indicizzazione dei contenuti.

enable

Abilita l'indicizzazione per la Ricerca FT in questa biblioteca

disable

Disabilita l'indicizzazione per la Ricerca FT in questa biblioteca

status

Mostra lo stato di indicizzazione corrente

reindex

Può essere usato per reindicizzare libri specifici o anche l'intera biblioteca. Per reindicizzare libri specifici devi fornire i loro id come argomenti aggiuntivi dopo il comando reindex. Se non viene specificato alcun id, l'intera biblioteca verrà reindicizzata.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--indexing-speed

La velocità di indicizzazione. Usa rapida per fare in modo che l'indicizzazione usi tutte le risorse disponibili del computer, lenta per un'indicizzazione con meno impatto sulle risorse. Tieni presente che la velocità è riportata a lenta dopo ogni utilizzo.

--wait-for-completion

Aspetta che tutti i libri siano indicizzati, mostrando periodicamente il progresso dell'indicizzazione

fts_search

```
calibredb fts_search [options] search expression
```

Esegui una ricerca nei contenuti sull'intera biblioteca o su una sua parte.

Ogni qualvolta vengono passati argomenti con degli spazi a calibredb, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

--do-not-match-on-related-words

Trova solo le parole esatte e non quelle derivate. Per esempio, correggere non troverà correzione.

--include-snippets

Includi estratti del testo nelle vicinanze dei risultati. Tieni presente che questo rende la ricerca molto più lenta.

--indexing-threshold

Quale percentuale della biblioteca deve essere indicizzata prima che la ricerca sia permessa. Il valore predefinito è 90

--match-end-marker

Il marcatore usato per indicare la fine di una parola corrispondente dentro un estratto

--match-start-marker

Il marcatore usato per indicare l'inizio di una parola corrispondente dentro un estratto

--output-format

Il formato da usare per i risultati. La scelta è tra "text" per testo semplice e "json" per il formato JSON.

--restrict-to

Limita la ricerca a libri specifici, usando un'espressione di ricerca oppure gli id. Per esempio: ids:1,2,3 limita usando gli id, mentre search:tag:qualcosa limita la ricerca ai libri con il tag "qualcosa".

13.1.7 ebook-convert

```
ebook-convert file_di_input file_di_output [opzioni]
```

Converti un e-book da un formato a un altro.

file_di_input è il file da convertire e file_di_output è il file convertito. Entrambi devono essere specificati come primi due argomenti del comando.

Il formato di output viene dedotto dall'estensione del file convertito file_di_output. file_di_output può anche essere del formato speciale .EXT dove EXT è l'estensione del file di output. In questo caso il nome del file di output è derivato dal nome del file di input. Nota che i nomi dei file non devono cominciare un trattino. Infine, se file_di_output non ha estensione verrà trattato come cartella e un «open e-book» (OEB) formato da file HTML viene inserito nella cartella. Questi file HTML sono quelli che normalmente sarebbero stati passati al plugin atto a generare l'output.

Dopo aver specificato i file di input e output è possibile personalizzare la conversione specificando varie opzioni. Le opzioni disponibili dipendono dal tipo di file di input e output. Per avere maggiori informazioni sulle opzioni specifiche, indica i file di input e output e poi inserisci l'opzione -h.

Per la documentazione completa sul sistema di conversione, vedi *Conversione E-book* (pagina 53)

Ogni qualvolta vengono passati argomenti con degli spazi a **ebook-convert**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

Le opzioni e i valori predefiniti cambiano a seconda sia del formato di input che di output, quindi bisognerebbe sempre controllare:

```
ebook-convert myfile.input_format myfile.output_format -h
```

Sotto sono presentate le opzioni comuni per tutte le conversioni, seguite da opzioni specifiche per ogni formato di input e di output.

- *OPZIONI DI INPUT* (pagina 320)
- *OPZIONI DI OUTPUT* (pagina 320)
- *IMPRESSIONE VISIVA* (pagina 321)
- *ELABORAZIONE EURISTICA* (pagina 324)
- *CERCA E SOSTITUISCI* (pagina 324)
- *RILEVAZIONE STRUTTURA* (pagina 325)
- *INDICE* (pagina 326)
- *METADATI* (pagina 326)
- *DEBUG* (pagina 327)

--help, -h

mostra questo messaggio di aiuto ed esci

--input-profile

Specifica il profilo di input. Il profilo di input fornisce informazioni al sistema di conversione su come interpretare i vari dati del documento di input. Per esempio le lunghezze dipendenti dalla risoluzione (es. lunghezze in pixel). Le scelte sono: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes

Elenca i nomi delle ricette integrate. Puoi creare un e-book da una ricetta integrata in questo modo: ebook-convert "Nome Ricetta.recipe" output.epub

--output-profile

Specifica il profilo di output. Il profilo di output indica al sistema di conversione come ottimizzare il documento creato per il dispositivo indicato. In alcuni casi il profilo di output può essere usato per ottimizzare l'output per un particolare dispositivo, ma ciò è necessario raramente. Le scelte sono: cybookg3, cybook_opus, default, generic_eink, generic_eink_hd, generic_eink_large, hanlinv3, hanlinv5, illiad, ipad, ipad3, irexdr1000, irexdr800, jetbook5, kindle, kindle_dx, kindle_fire, kindle_oasis, kindle_pw, kindle_pw3, kindle_scribe, kindle_voyage, kobo, msreader, mobipocket, nook, nook_color, nook_hd_plus, pocketbook_inkpad3, pocketbook_lux, pocketbook_hd, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet

--version

mostra la versione del programma ed esci

OPZIONI DI INPUT

Opzioni per controllare l'elaborazione dei file di input mobi

--input-encoding

Specifica la codifica dei caratteri del documento di origine. Se impostata, questa opzione sovrascrive l'eventuale codifica indicata nel documento stesso. Particolarmente utile per documenti in cui la codifica non è esplicitamente indicata o che hanno dichiarazioni di codifica errate.

OPZIONI DI OUTPUT

Opzioni per controllare l'elaborazione dell'output epub

--dont-split-on-page-breaks

Disattiva la divisione dei file alle interruzioni di pagina. Normalmente i file di input sono automaticamente divisi a ogni interruzione di pagina in due file. Ciò dà in output un e-book che può essere decodificato più velocemente e con minore uso di risorse. Tuttavia, la suddivisione è lenta e se il tuo file sorgente contiene un gran numero di interruzioni di pagina, è consigliabile disattivare la suddivisione alle interruzioni di pagina.

--epub-flatten

Questa opzione è necessaria solo se si intende usare l'epub con FBReaderJ. Verrà linearizzata la struttura all'interno dell'epub, tutti i file saranno spostati nella cartella principale.

--epub-inline-toc

Inserisci un indice che apparirà come parte del contenuto principale del libro

--epub-max-image-size

La massima dimensione dell'immagine (larghezza x altezza). Il valore none vuol dire che verrà usata la dimensione dello schermo del profilo di output. Il valore profile non applicherà alcuna limitazione alla dimensione. Per esempio, un valore di 100x200 ridimensionerà tutte le immagini in modo che la loro larghezza non sia più di 100 pixel, né la loro altezza più di 200 pixel. Tieni presente che questa opzione si riferisce solo alla dimensione effettiva dei file di immagine stessi. Un'immagine potrebbe apparire con una dimensione differente in base agli stili ad essa applicati nel documento.

--epub-toc-at-end

Metti l'indice in linea inserito alla fine del libro invece che all'inizio.

--epub-version

La versione del file EPUB da generare. EPUB 2 è quello largamente compatibile, usa EPUB 3 solo se ritieni sia necessario.

--extract-to

Estrai i contenuti del file EPUB generato nella cartella selezionata. I contenuti della cartella vengono prima eliminati, quindi fai attenzione.

--flow-size

Separa tutti i file HTML più grandi di questa dimensione (in KB). Ciò è necessario in quanto molti lettori EPUB non sono in grado di gestire file di dimensioni maggiori. La dimensione di 260KB è quella predefinita per le Edizioni Digitali di Adobe. Imposta a 0 per disabilitare la divisione basata sulla dimensione.

--no-default-epub-cover

In genere, se il file di input non ha una copertina e non ne viene specificata una, viene generata una copertina predefinita con il titolo, l'autore, etc. Questa opzione disabilita la generazione di questa copertina.

--no-svg-cover

Non utilizzare SVG per la copertina del libro. Usa questa opzione se l'EPUB verrà usato su un dispositivo che non supporta SVG, come l'iPhone o il JetBook Lite. Senza questa opzione, questi dispositivi visualizzeranno come copertina una pagina bianca.

--preserve-cover-aspect-ratio

Quando viene usata una copertina SVG, questa opzione ridimensiona la copertina per coprire l'area dello schermo, ma preservando le proporzioni. Questo significa che potrebbero esserci dei bordi bianchi intorno alla copertina, ma questa non verrà distorta. Senza questa opzione l'immagine potrebbe essere sproporzionata anche se non avrà bordi bianchi.

--pretty-print

Se specificato, il plugin tenta di creare un output il più leggibile possibile. Potrebbe non avere effetto per alcuni plugin.

--toc-title

Titolo per ogni indice in linea generato.

IMPRESSIONE VISIVA

Opzioni per controllare l'aspetto dell'output

--asciize

Converti i caratteri Unicode in una rappresentazione ASCII. Usa con attenzione perché sostituirà i caratteri Unicode con caratteri ASCII. Per esempio sostituirà "Pelé" con "Pele". In più, nota che nei casi in cui ci sia più di una rappresentazione per un carattere (caratteri presenti sia in cinese che in giapponese per esempio) la rappresentazione basata sul linguaggio corrente dell'interfaccia di calibre sarà usata.

--base-font-size

La dimensione base dei caratteri in pt. Tutte le dimensioni dei caratteri nel libro prodotto saranno scalate in base a questa dimensione. Scegliendo una dimensione maggiore si possono ottenere caratteri più grandi nell'output e viceversa. In modo predefinito, quando il valore è zero, la dimensione base dei caratteri è scelta in base al profilo di output scelto.

--change-justification

Cambia allineamento del testo. Impostato a "sinistra" il testo della sorgente sarà convertito con un allineamento a sinistra. Impostato a "giustificato" il testo verrà distribuito per larghezza della pagina. Impostato a "originale" (predefinito) l'allineamento non verrà cambiato. Notare che solo alcuni formati di output supportano il testo giustificato.

--disable-font-rescaling

Disabilita ogni ridimensionamento delle dimensioni dei caratteri.

--embed-all-fonts

Incorpora qualsiasi carattere che sia referenziato ma non ancora incorporato nel documento di input. Verranno cercati i caratteri nel sistema e, se trovati, verranno incorporati. L'incorporamento funziona solo se il formato di destinazione, come EPUB, AZW3, DOCX o PDF, supporta l'incorporamento dei caratteri. Assicurati di avere una licenza valida per l'incorporamento dei caratteri usati nel documento.

--embed-font-family

Incorpora la famiglia caratteri specificata nel libro. Permette di specificare il carattere di "base" utilizzato per il libro. Se il documento di input specifica i propri caratteri, potrebbero sovrascrivere il carattere di base. Nota che l'incorporamento dei caratteri funziona solo con alcuni formati di output, tra cui EPUB, AZW3 e DOCX.

--expand-css

Per impostazione predefinita, calibre utilizza la forma abbreviata per varie proprietà CSS come margin, padding, border, ecc. Questa opzione farà usare invece la loro forma completa. Tieni presente che il CSS è sempre espanso quando si generano file EPUB con il profilo di output impostato su uno dei profili Nook, in quanto il Nook non può gestire un CSS abbreviato.

--extra-css

Questo può essere il percorso di un foglio di stile CSS o semplice codice CSS. Il CSS verrà unito alle regole di stile del file sorgente e pertanto può essere usato per prevalere su di esse.

--filter-css

Un elenco separato da virgole delle proprietà CSS che saranno rimosse da tutte le regole di stile CSS. Ciò è utile se la presenza di alcune informazioni di stile impedisce l'applicazione delle regole del dispositivo. Ad esempio: font-family,color,margin-left,margin-right

--font-size-mapping

Correla le dimensioni relative dei caratteri CSS a valori di dimensione in pt. Un esempio di valori è 10,12,14,16,18,20,22,24. Questi sono i valori per le dimensioni che vanno da xx-small a xx-large, con l'ultima dimensione usata per caratteri enormi. L'algoritmo di ridimensionamento usa questi riferimenti per ridimensionare i caratteri. L'impostazione predefinita è l'utilizzo di valori basati sul profilo di output scelto.

--insert-blank-line

Inserisci una riga vuota tra i paragrafi. Non funzionerà se il file sorgente non usa i paragrafi (tag <p> o <div>).

--insert-blank-line-size

Imposta l'altezza delle righe vuote inserite (in em). L'altezza delle righe tra i paragrafi sarà doppia del valore impostato qui.

--keep-ligatures

Mantieni le legature presenti nel documento di input. Una legatura è una formattazione speciale di una coppia di caratteri come ff, fi, fl, eccetera. La maggior parte dei lettori non supporta le legature nei loro caratteri predefiniti, perciò probabilmente queste non verranno mostrate correttamente. Per impostazione predefinita calibre divide le legature nei suoi due caratteri corrispondenti. Questa opzione invece le manterrà.

--line-height

Altezza della riga, espressa in punti. Gestisce la spaziatura tra righe consecutive di testo. Si applica solo ad elementi che non definiscono la propria altezza di riga. Nella maggior parte dei casi, l'opzione che definisce l'altezza minima della riga è più utile. Per impostazione predefinita non viene applicata alcuna manipolazione di altezza di riga.

--linearize-tables

Alcuni documenti mal costruiti usano tabelle per controllare la struttura del testo nella pagina. Questi documenti, una volta convertiti, spesso presentano testi non allineati nella pagina ed altri difetti grafici. Questa opzione estrae il contenuto dalle tabelle e lo presenta in maniera lineare.

--margin-bottom

Imposta il margine inferiore in pt. Il valore predefinito è 5.0. L'impostazione a un valore minore di zero non produrrà alcuna variazione del margine (rimarrà impostato il valore originale). Nota: formati basati sulle pagine come PDF e DOCX hanno impostazioni interne sui margini, che prevalgono.

--margin-left

Imposta il margine sinistro in pt. Il valore predefinito è 5.0. L'impostazione a un valore minore di zero non produrrà alcuna variazione del margine (rimarrà impostato il valore originale). Nota: formati basati sulle pagine come PDF e DOCX hanno impostazioni interne sui margini, che prevalgono.

--margin-right

Imposta il margine destro in pt. Il valore predefinito è 5.0. L'impostazione a un valore minore di zero non produrrà

alcuna variazione del margine (rimarrà impostato il valore originale). Nota: formati basati sulle pagine come PDF e DOCX hanno impostazioni interne sui margini, che prevalgono.

--margin-top

Imposta il margine superiore in pt. Il valore predefinito è 5.0. L'impostazione a un valore minore di zero non produrrà alcuna variazione del margine (rimarrà impostato il valore originale). Nota: formati basati sulle pagine come PDF e DOCX hanno impostazioni interne sui margini, che prevalgono.

--minimum-line-height

L'altezza minima della riga, come percentuale della dimensione degli elementi del carattere. Calibre farà in modo che ogni elemento abbia almeno questa dimensione, senza tenere conto di quanto specifica il documento in input. Per disabilitare, imposta il valore a zero. Il valore predefinito è 120%. Usa questa impostazione rispetto all'impostazione diretta dell'altezza della riga, a meno di conoscere esattamente cosa stai facendo. Ad esempio, è possibile impostare la "Spaziatura doppia" utilizzando il valore 240.

--remove-paragraph-spacing

Rimuovi gli spazi fra i paragrafi. Imposta anche un rientro dei paragrafi di 1.5em. La rimozione degli spazi non funzionerà se il documento sorgente non usa la notazione per i paragrafi del tipo <p> o <div>.

--remove-paragraph-spacing-indent-size

Quando calibre rimuove le righe vuote tra paragrafi, imposta automaticamente un rientro di paragrafo, per assicurare che i paragrafi siano facilmente distinguibili. Questa opzione controlla l'ampiezza di questo rientro (in em). Se imposti un valore negativo, viene utilizzato il rientro specificato nel documento in ingresso, in altre parole, calibre non cambierà il rientro.

--smarten-punctuation

Converti gli apici semplici, i trattini e i puntini di sospensione nei loro equivalenti tipografici corretti. Per i dettagli, vedi <https://daringfireball.net/projects/smartyants>.

--subset-embedded-fonts

Crea un sottoinsieme di tutti i caratteri integrati. Ogni carattere integrato è ridotto per contenere solo i glifi utilizzati in questo documento. Ciò riduce la dimensione dei file dei caratteri. Utile se stai incorporando un carattere particolarmente grande con molti glifi non utilizzati.

--transform-css-rules

Percorso a un file che contiene regole per trasformare gli stili CSS di questo libro. Il modo più semplice per creare questo file è quello di usare la procedura guidata per la creazione di regole nell'interfaccia utente di calibre. Avvialo dalla sezione della finestra di conversione "Aspetto->Trasforma stili". Dopo aver creato queste regole, puoi usare il pulsante "Esporta" per salvarle in un file.

--transform-html-rules

Percorso a un file che contiene regole per trasformare il codice HTML di questo libro. Il modo più semplice per creare questo file è quello di usare la procedura guidata per la creazione di regole nell'interfaccia utente di calibre. Avvialo dalla sezione della finestra di conversione "Aspetto->Trasforma HTML". Dopo aver creato queste regole, puoi usare il pulsante "Esporta" per salvarle in un file.

--unsmarten-punctuation

Converti virgolette eleganti, trattini e punti di sospensione nel loro equivalente semplice.

ELABORAZIONE EURISTICA

Modifica il testo e la struttura del documento utilizzando schemi comuni. Disabilitata in modo predefinito. Usa `--enable-heuristics` per abilitarla. Azioni singole possono essere disabilitate con le opzioni `--disable-*`.

`--disable-dehyphenate`

Analizza nel documento le parole che contengono un trattino. Lo stesso documento è utilizzato come un dizionario per determinare quando i trattini devono essere rimossi o mantenuti.

`--disable-delete-blank-paragraphs`

Rimuovi i paragrafi vuoti dal documento quando sono presenti tra gli altri paragrafi

`--disable-fix-indents`

Cambia il rientro ottenuto con spazi multipli in un rientro CSS.

`--disable-format-scene-breaks`

I marcatori di interruzione di scena allineati a sinistra sono centrati. Sostituisci le interruzioni di scena leggere, che usano più paragrafi vuoti, con un tratteggio orizzontale.

`--disable-italicize-common-cases`

Cerca parole e segni che indicano l'uso del corsivo e rendili in corsivo.

`--disable-markup-chapter-headings`

Rileva i titoli e i sottotitoli di paragrafo non formattati. Cambia i tag in h2 e h3. Questa impostazione non creerà un indice, ma può essere utilizzata con la funzione di rilevamento della struttura crearne uno.

`--disable-renumber-headings`

Cerca occorrenze di sequenze di tag `<h1>` o `<h2>`. I tag sono rinumerati per prevenire la divisione a metà del titolo di capitolo.

`--disable-unwrap-lines`

Unisci le righe utilizzando la punteggiatura o altri metodi di formattazione.

`--enable-heuristics`

Abilita l'elaborazione euristica. Questa opzione deve essere impostata per attivare l'elaborazione euristica.

`--html-unwrap-factor`

Scala usata per determinare la posizione alla quale una riga dovrebbe essere ricongiunta. I valori ammessi sono i decimali compresi tra 0 e 1. Il valore predefinito è 0.4, appena inferiore al valore mediano della lunghezza. Se solo alcune righe del documento richiedono il ricongiungimento questo valore dovrebbe essere ridotto.

`--replace-scene-breaks`

Sostituisci i marcatori di interruzione scena con un testo specifico. In modo predefinito, viene usato il testo del documento.

CERCA E SOSTITUISCI

Modifica il testo e la struttura del documento usando pattern definiti.

`--search-replace`

Percorso a un file che contiene espressioni regolari di ricerca e sostituzione. Il file deve contenere linee alternate di espressione regolare seguita da un modello di sostituzione (anche una riga vuota). L'espressione regolare deve seguire la sintassi regex di Python e il file deve utilizzare la codifica UTF-8.

`--sr1-replace`

Elemento sostitutivo del testo trovato con `sr1-search`.

--sr1-search

Pattern di ricerca (espressione regolare) da sostituire con sr1-replace.

--sr2-replace

Elemento sostitutivo del testo trovato con sr2-search.

--sr2-search

Pattern di ricerca (espressione regolare) da sostituire con sr2-replace.

--sr3-replace

Elemento sostitutivo del testo trovato con sr3-search.

--sr3-search

Pattern di ricerca (espressione regolare) da sostituire con sr3-replace.

RILEVAZIONE STRUTTURA

Controlla il rilevamento automatico della struttura del documento.

--chapter

Un'espressione XPath per rilevare i titoli dei capitoli. Per impostazione predefinita si considerano come titoli di capitolo i tag <h1> o <h2> che contengono le parole "chapter", "book", "section", "prologue", "epilogue" o "part", così come ogni tag che contenga class="chapter". L'espressione usata deve restituire un elenco di elementi. Per disabilitare la rilevazione dei capitoli, usa l'espressione "/". Vedi la Guida XPath nel manuale utente di calibre per ulteriore aiuto nell'usare questa funzione.

--chapter-mark

Specificare come marcare i capitoli rilevati. Il valore "pagebreak" inserirà un'interruzione di pagina prima del capitolo. Il valore "rule" inserirà una linea prima del capitolo. Il valore "none" disabiliterà la marcatura dei capitoli e il valore "both" userà sia l'interruzione di pagina che la linea per marcare i capitoli.

--disable-remove-fake-margins

Alcuni documenti specificano i margini destro e sinistro per ogni singolo paragrafo. calibre cercherà di trovare ed eliminare questi margini. Qualche volta questo meccanismo elimina però dei margini che non dovrebbero essere rimossi. In questo caso disabilita la funzione.

--insert-metadata

Inserisci i metadati del libro all'inizio. Utile se il dispositivo non supporta la visualizzazione/ricerca di metadati direttamente.

--page-breaks-before

Un'espressione XPath. Le interruzioni di pagina sono inserite prima degli elementi specificati. Per disabilitarla, usa l'espressione: /

--prefer-metadata-cover

Usa la copertina rilevata dal file di origine al posto di quella specificata.

--remove-first-image

Rimuovi la prima immagine dall'e-book in ingresso. Utile se il documento in ingresso ha un'immagine di copertina che non è identificata come tale. In questo caso, se imposti una copertina tramite calibre, il documento si troverà con due immagini di copertina se non specifichi questa opzione.

--start-reading-at

Un'espressione XPath per rilevare la posizione nel documento da cui cominciare la lettura. Alcuni programmi di lettura e-book (in particolare Kindle) usano questa posizione come quella in cui il libro viene aperto. Vedi la guida XPath nel manuale utente di calibre per maggiori informazioni sull'utilizzo di questa funzione.

INDICE

Gestisce la generazione automatica dell'indice. In modo predefinito, se il file sorgente ha un indice, questo sarà preferito a quello generato automaticamente.

--duplicate-links-in-toc

Quando crei un indice dai collegamenti nel documento in ingresso, consenti la duplicazione delle voci, ovvero permetti più di una voce con lo stesso testo, a condizione che si riferiscano a una posizione diversa.

--level1-toc

L'espressione XPath che specifica tutti i tag che devono essere aggiunti al primo livello dell'indice. Se è specificato, ha la precedenza su altre forme di rilevamento automatico. Vedi la guida a XPath del manuale utente di calibre per gli esempi.

--level2-toc

L'espressione XPath che specifica tutti i tag che devono essere aggiunti al secondo livello dell'indice. Ogni voce è aggiunta sotto la precedente voce di primo livello. Vedi la guida a XPath del manuale utente di calibre per gli esempi.

--level3-toc

L'espressione XPath che specifica tutti i tag che devono essere aggiunti al terzo livello dell'indice. Ogni voce è aggiunta sotto la precedente voce di secondo livello. Vedi la guida a XPath del manuale utente di calibre per gli esempi.

--max-toc-links

Massimo numero di collegamenti da inserire nell'indice. Inserisci 0 per disabilitare. Il valore predefinito è 50. I collegamenti sono aggiunti all'indice soltanto se non viene raggiunto il valore soglia dei capitoli.

--no-chapters-in-toc

Non aggiungere i capitoli rilevati automaticamente all'indice.

--toc-filter

Rimuovi dall'indice le voci il cui titolo corrisponde alla espressione specificata. Le voci che corrispondono saranno rimosse insieme alle voci "figlie".

--toc-threshold

Se viene rilevato un numero di capitoli inferiore a questo numero, i collegamenti interni al libro saranno aggiunti all'indice. Predefinito: 6

--use-auto-toc

Normalmente, se il file sorgente possiede già un indice, esso è usato di preferenza rispetto a quello auto-generato. Con questa opzione, verrà sempre usato l'indice auto-generato.

METADATI

Opzioni per impostare i metadati in output

--author-sort

Stringa da utilizzare quando si ordina per autore.

--authors

Imposta gli autori. Autori multipli devono essere separati dal carattere &.

--book-producer

Imposta il produttore del libro.

--comments

Imposta la descrizione dell'e-book.

--cover

Seleziona la copertina da un file o da un URL

--isbn

Imposta il codice ISBN del libro.

--language

Imposta la lingua.

--pubdate

Imposta la data di pubblicazione (viene usata l'ora locale, a meno che il fuso orario non sia specificato esplicitamente)

--publisher

Imposta l'editore dell'e-book.

--rating

Imposta la valutazione. Deve essere un numero compreso tra 1 e 5.

--read-metadata-from-opf, --from-opf, -m

Leggi i metadati dal file OPF specificato. I metadati letti da questo file sovrascrivono qualunque altro metadato presente nel file sorgente.

--series

Imposta la serie alla quale questo e-book appartiene.

--series-index

Imposta la numerazione del libro in questa serie.

--tags

Imposta i tag per il libro. Deve essere un elenco separato da virgole.

--timestamp

Imposta la marca temporale del libro (non più utilizzata ovunque)

--title

Imposta il titolo.

--title-sort

La versione del titolo da usare per l'ordinamento.

DEBUG

Opzioni per aiutare a individuare problemi con la conversione

--debug-pipeline, -d

Salva l'output dei differenti stadi di conversione raggiunti nella cartella specificata. Utile se non si conosce in quale stadio del processo di conversione si verifica l'errore.

--verbose, -v

Livello di verbosità. Specificare più volte per maggiore verbosità. Due volte verbosità massima, una volta per verbosità media e zero per quella più bassa.

13.1.8 ebook-edit

```
%p [opts] [percorso_dell_ebook] [nome_del_file_nel_libro ...]
```

Lancia lo strumento di calibre Modifica libro. Volendo puoi anche specificare i nomi dei file dentro il libro che saranno aperti automaticamente per la modifica.

Ogni qualvolta vengono passati argomenti con degli spazi a **ebook-edit**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--detach

Scollega dal terminale di controllo, se presente (solo su Linux)

--help, -h

mostra questo messaggio di aiuto ed esci

--select-text

Il testo da selezionare nel libro quando viene aperto per la modifica

--version

mostra la versione del programma ed esci

13.1.9 ebook-meta

```
ebook-meta ebook_file [opzioni]
```

Leggi/Scrivi metadati da/su file e-book.

Formati supportati per la lettura dei metadati: azw, azw1, azw3, azw4, cb7, cbc, cbr, cbz, chm, docx, epub, fb2, fbz, html, htmlz, imp, lit, lrf, lrx, mobi, odt, oebzip, opf, pdb, pdf, pml, pmlz, pobi, prc, rar, rb, rtf, snb, tpz, txt, txtz, updb, zip

Formati supportati per la scrittura dei metadati: azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

Tipi di file differenti supportano differenti tipi di metadati. Se imposti alcuni metadati in un tipo di file che non li supporta, i metadati saranno ignorati silenziosamente.

Ogni qualvolta vengono passati argomenti con degli spazi a **ebook-meta**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--author-sort

La stringa che verrà usata nell'ordinamento per autore. Se non specificata, essa verrà generata automaticamente dal nome dell'autore.

--authors, -a

Imposta gli autori. Multipli autori devono essere divisi dal carattere &. I nomi degli autori devono essere nell'ordine Nome Cognome.

--book-producer, -k

Imposta il produttore del libro.

--category

Imposta la categoria del libro.

--comments, -c

Imposta la descrizione dell ' e-book.

--cover

Imposta la copertina per il file specificato.

--date, -d

Imposta la data di pubblicazione.

--from-opf

Leggi i metadati dal file OPF specificato e usalo per impostare i metadati nell ' ebook. I metadati specificati da linea di comando sovrascriveranno quelli letti dal file OPF

--get-cover

Estrai la copertina dall 'e-book e salvala in un file specificato.

--help, -h

mostra questo messaggio di aiuto ed esci

--identifier

Imposta gli identificatori del libro, possono essere specificati più volte. Per esempio: `--identifier` (pagina 329) `uri:https://acme.com` `--identifier` (pagina 329) `isbn:12345` Per rimuovere un identificatore non specificare alcun valore `--identifier` (pagina 329) `isbn:` Nota che per i file EPUB, un identificatore impostato come identificatore del pacchetto non può essere rimosso.

--index, -i

Imposta la numerazione del libro in questa serie.

--isbn

Imposta il codice ISBN del libro.

--language, -l

Imposta la lingua.

--lrf-bookid

Imposta il BookID nei file LRF

--publisher, -p

Imposta l ' editore dell ' e-book.

--rating, -r

Imposta la valutazione. Deve essere un numero compreso tra 1 e 5.

--series, -s

Imposta la serie alla quale questo e-book appartiene.

--tags

Imposta i tag per il libro. Deve essere un elenco separato da virgole.

--title, -t

Imposta il titolo.

--title-sort

La versione del titolo da usare per l ' ordinamento. Se non specificata, la versione verrà generata automaticamente dal titolo.

--to-opf

Specifica il nome di un file OPF. I metadati saranno scritti in questo file.

--version

mostra la versione del programma ed esci

13.1.10 ebook-polish

```
ebook-polish [opzioni] file_di_input [file_di_output]
```

La rifinitura dei libri è ciò che ti serve per raggiungere il non plus ultra negli e-book che con cura hai costruito.

La rifinitura prova a minimizzare i cambiamenti al codice interno del tuo e-book. Diversamente dalla conversione, non appiattisce il CSS, rinomina file, cambia la dimensione dei caratteri, modifica i margini, ecc. Ogni azione effettua solo i minimi cambiamenti necessari ad ottenere l'effetto desiderato.

Dovresti usare questo strumento come ultimo passo nel processo di creazione del tuo e-book.

Nota che la rifinitura funziona solo sui file nei formati AZW3 oppure EPUB.

Ogni qualvolta vengono passati argomenti con degli spazi a **ebook-polish**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--add-soft-hyphens, -H

Aggiungi trattini morbidi a tutte le parole nel libro. Questo permette al libro di essere impaginato meglio quando il testo è giustificato, nei lettori che non supportano la divisione in sillabe.

--compress-images, -i

Comprimi le immagini nel libro per ridurre le dimensioni del file senza alcuna perdita di qualità.

--cover, -c

Percorso di un'immagine di copertina. Cambia la copertina specificata nell'e-book. Se nessuna copertina è presente, o se la copertina non è identificata correttamente, viene inserita una nuova copertina.

--embed-fonts, -e

Incorpora tutti i caratteri che sono referenziati nel documento ma non ancora incorporati. Verranno cercati i caratteri nel computer e, se trovati, verranno incorporati nel documento. Assicurati di avere una licenza valida per l'incorporamento dei caratteri usati nel documento.

--help, -h

mostra questo messaggio di aiuto ed esci

--jacket, -j

Inserisci una pagina di "sovracopertina" all'inizio del libro che contiene tutti i metadati del libro come titolo, tag, autori, serie, commenti, ecc. Ogni sovracopertina precedente sarà rimpiazzata.

--opf, -o

Percorso a un file OPF. I metadati nel libro sono aggiornati dal file OPF.

--remove-jacket

Rimuovi una sovracopertina inserita precedentemente.

--remove-soft-hyphens

Rimuovi i trattini morbidi da tutto il testo del libro.

--remove-unused-css, -u

Rimuovi tutte le regole CSS non utilizzate dai fogli di stile e i tag `<style>`. Alcuni libri creati da modelli professionali possono avere un gran numero di regole CSS che non corrispondono ad alcun contenuto effettivo. Le regole aggiuntive tendono a diminuire le prestazioni dei dispositivi di lettura.

--smarten-punctuation, -p

Converti trattini, parentesi, apici, apostrofi, ecc. nelle loro rispettive forme tipografiche corrette. L'algoritmo potrebbe generare a volte risultati indesiderati, specialmente quando si presentano apici singoli all'inizio di contrazioni.

--subset-fonts, -f

Creare un sottoinsieme dei caratteri significa fare in modo che un carattere integrato contenga solo i simboli utilizzati da quel carattere nel libro. Ciò riduce di molto la dimensione dei file dei caratteri (è comune dimezzarne la dimensione). Ad esempio, se il libro usa un carattere specifico per i titoli, la riduzione farà sì che quel carattere contenga solo i simboli presenti attualmente nei titoli del libro. Oppure se il libro contiene la versione in corsivo e in grassetto di un carattere, ma il testo in grassetto o in corsivo è relativamente raro, oppure completamente assente, allora i caratteri in corsivo e grassetto possono essere ridotti a solamente alcuni simboli, oppure completamente eliminati. L'unico svantaggio del ridurre i caratteri è che, se in seguito vuoi aggiungere altro testo al tuo libro, per il testo aggiunto potrebbero non essere sufficienti i caratteri integrati.

--upgrade-book, -U

Aggiorna la struttura interna del libro, se possibile. Per esempio, aggiorna libri in formato EPUB 2 a EPUB3.

--verbose

Genera un risultato più prolisso, utile per debug.

--version

mostra la versione del programma ed esci

13.1.11 ebook-viewer

```
ebook-viewer [opzioni] file
```

Vedi un e-book.

Ogni qualvolta vengono passati argomenti con degli spazi a **ebook-viewer**, è necessario mettere l'argomento fra virgolette. Ad esempio: `</some path/with spaces>`

[opzioni]**--continue**

Prosegui la lettura dell'ultimo libro aperto

--detach

Scollega dal terminale di controllo, se presente (solo su Linux)

--force-reload

Ricarica forzatamente tutti i libri aperti

--full-screen, --fullscreen, -f

Se selezionato, la finestra del lettore e-book proverà ad aprirsi a schermo intero all'avvio.

--help, -h

mostra questo messaggio di aiuto ed esci

--open-at

La posizione alla quale aprire il libro specificato. La posizione è una collocazione o posizione a cui puoi recarti usando l'azione Vai a->Collocazione nei controlli del lettore. In alternativa puoi usare lo schema toc:qualcosa e il libro si aprirà alla collocazione della prima voce dell'indice che contenga la stringa "qualcosa". Lo schema toc-href:qualcosa cercherà nello href (la destinazione interna del collegamento) degli elementi dell'indice. La corrispondenza è di tipo esatto. Per cercare una sottostringa, usa lo schema toc-href-contains:qualcosa. Lo schema ref:qualcosa userà i riferimenti della Modalità Riferimento. Lo schema search:qualcosa cercherà "qualcosa" dopo aver aperto il libro. Lo schema regex:qualcosa cercherà con l'espressione regolare "qualcosa" dopo aver aperto il libro.

--raise-window

Se selezionato, la finestra del lettore e-book proverà a portarsi in primo piano all'avvio.

--version

mostra la versione del programma ed esci

13.1.12 **fetch-ebook-metadata**

```
fetch-ebook-metadata [options]
```

Recupera i metadati del libro dalle fonti online. È necessario specificare almeno uno tra titolo, autori o ISBN.

Ogni qualvolta vengono passati argomenti con degli spazi a **fetch-ebook-metadata**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--allowed-plugin, -p

Specifica il nome di un plugin di download dei metadati da usare. Per impostazione predefinita, tutti i plugin dei metadati verranno utilizzati. Può essere specificato più volte per più plugin. Tutti i nomi dei plugin: Google, Google Images, Amazon.com, Edelweiss, Open Library, Big Book Search

--authors, -a

Autore/i del libro

--cover, -c

Specifica un nome file. La copertina, se disponibile, sarà salvata all'interno di esso. Senza quest'opzione, non sarà scaricata nessuna copertina.

--help, -h

mostra questo messaggio di aiuto ed esci

--identifier, -I

Identificatori come ASIN/Id Goodreads ecc. Può essere specificato più volte per più identificatori. Per esempio: *--identifier* (pagina 332) asin:B0082BAJA0

--isbn, -i

ISBN del libro

--opf, -o

Restituisce i metadati in formato OPF invece di testo leggibile.

--timeout, -d

Tempo massimo in secondi. Predefinito è 30

- title, -t**
Titolo del libro
- verbose, -v**
Stampa il log sulla console (stderr)
- version**
mostra la versione del programma ed esci

13.1.13 lrf2lrs

```
lrf2lrs libro.lrf
```

Converti un file LRF in un file LRS (XML codificato UTF-8)

Ogni qualvolta vengono passati argomenti con degli spazi a **lrf2lrs**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

- dont-output-resources**
Non salvare le immagini incorporate e i caratteri sul disco
- help, -h**
mostra questo messaggio di aiuto ed esci
- output, -o**
Restituisce un file LRS
- verbose**
Aumenta la verbosità
- version**
mostra la versione del programma ed esci

13.1.14 lrfviewer

```
lrfviewer [opzioni] libro.lrf
```

Leggi l'e-book LRF libro.lrf

Ogni qualvolta vengono passati argomenti con degli spazi a **lrfviewer**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--disable-hyphenation

Disabilita Sillabazione. Dovrebbe velocizzare il rendering in maniera significativa.

--help, -h

mostra questo messaggio di aiuto ed esci

--profile

Profila il renderer LRF

--verbose

Stampa altre informazioni sul processo di resa

--version

mostra la versione del programma ed esci

--visual-debug

Attiva un aiuto visuale per il debug del motore di rendering

--white-background

Per impostazione predefinita, lo sfondo è bianco sporco in quanto meno stancante per gli occhi. Utilizzare questa opzione per rendere lo sfondo bianco assoluto.

13.1.15 **lrs2lrf**

```
lrs2lrf [opzioni] file.lrs
```

Compila un file LRS dentro un file LRF

Ogni qualvolta vengono passati argomenti con degli spazi a **lrs2lrf**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--help, -h

mostra questo messaggio di aiuto ed esci

--lrs

Converti LRS in LRS, utile per il debug

--output, -o

Percorso del file in uscita

--verbose

Elaborazione dettagliata

--version

mostra la versione del programma ed esci

13.1.16 web2disk

```
web2disk URL
```

Dove URL è per esempio <https://google.it>

Ogni qualvolta vengono passati argomenti con degli spazi a **web2disk**, è necessario mettere l'argomento fra virgolette. Ad esempio: «/some path/with spaces»

[opzioni]

--base-dir, -d

Cartella base in cui l'URL è salvato. Il valore predefinito è: .

--delay

Intervallo minimo in secondi tra due scaricamenti consecutivi. Predefinito: 0 s

--dont-download-stylesheets

Non scaricare i fogli di stile CSS

--encoding

La codifica caratteri del sito web che si sta cercando di scaricare. L'impostazione predefinita è quella di provare e indovinare la codifica.

--filter-regexp

Tutti i link che corrispondono a questa espressione regolare saranno ignorati. Questa opzione può essere usata più di una volta, nel qual caso fintantoché almeno un link corrisponde a una delle regex, sarà ignorato. Per impostazione predefinita, nessun link è ignorato. Se entrambe le regex filter e match sono specificate, la regex filter viene applicata per prima.

--help, -h

mostra questo messaggio di aiuto ed esci

--match-regexp

Saranno seguiti solamente i collegamenti che corrispondono a questa espressione regolare. Questa opzione può essere specificata più volte, in questo modo se un collegamento corrisponde a una delle espressioni regolari sarà seguito. Per impostazione predefinita i collegamenti non vengono seguiti

--max-files, -n

Il numero massimo di file da scaricare. Questo si applica solo ai file con tag <a href>. Il valore predefinito è 9223372036854775807

--max-recursions, -r

Numero massimo di livelli ricorsivi, cioè profondità dei link da seguire. Predefinito: 1

--timeout, -t

Timeout in secondi da aspettare per una risposta dal server. Predefinito: 10.0 s

--verbose

Mostra un output dettagliato. Utile per il debug

--version

mostra la versione del programma ed esci

13.2 Comandi non documentati

- ebook-device
- markdown-calibre

Puoi vedere le istruzioni per i comandi non documentati eseguendoli nel terminale senza argomenti.

Creazione di un ambiente di sviluppo di calibre

calibre is completely open source, licensed under the [GNU GPL v3](https://www.gnu.org/licenses/gpl.html)¹⁰⁸. This means that you are free to download and modify the program to your heart's content. In this section, you will learn how to get a calibre development environment set up on the operating system of your choice. calibre is written primarily in [Python](https://www.python.org)¹⁰⁹ with some C/C++ code for speed and system interfacing. Note that calibre requires at least Python 3.8.

Contenuti

- *Filosofia di progettazione* (pagina 338)
 - *Vista codice* (pagina 338)
- *Ottenere il codice* (pagina 339)
 - *Inviare le proprie modifiche affinché siano incluse* (pagina 339)
- *Ambiente di sviluppo Windows* (pagina 340)
- *Ambiente di sviluppo macOS* (pagina 341)
- *Ambiente di sviluppo Linux* (pagina 342)
- *Convivenza tra due installazioni di calibre, «normale» e «di sviluppo», sullo stesso computer* (pagina 342)
- *Consigli per il debug* (pagina 343)
 - *Using print statements* (pagina 343)
 - *Using an interactive Python interpreter* (pagina 343)
 - *Using the Python debugger as a remote debugger* (pagina 343)
 - *Using the debugger in your favorite Python IDE* (pagina 344)
 - *Executing arbitrary scripts in the calibre Python environment* (pagina 344)
- *Usare calibre nei tuoi progetti* (pagina 344)

¹⁰⁸ <https://www.gnu.org/licenses/gpl.html>

¹⁰⁹ <https://www.python.org>

- *Binary install of calibre* (pagina 345)
- *Installazione da sorgenti su Linux* (pagina 345)
- *Documentazione delle API di varie componenti di calibre* (pagina 345)

14.1 Filosofia di progettazione

calibre has its roots in the Unix world, which means that its design is highly modular. The modules interact with each other via well defined interfaces. This makes adding new features and fixing bugs in calibre very easy, resulting in a frenetic pace of development. Because of its roots, calibre has a comprehensive command line interface for all its functions, documented in `generated/en/cli-index`.

The modular design of calibre is expressed via `Plugins`. There is a *tutorial* (pagina 247) on writing calibre plugins. For example, adding support for a new device to calibre typically involves writing less than a 100 lines of code in the form of a device driver plugin. You can browse the *built-in drivers*¹¹⁰. Similarly, adding support for new conversion formats involves writing input/output format plugins. Another example of the modular design is the *recipe system* (pagina 29) for fetching news. For more examples of plugins designed to add features to calibre, see the *Index of plugins*¹¹¹.

14.1.1 Vista codice

All the calibre Python code is in the `calibre` package. This package contains the following main sub-packages

- `devices` - All the device drivers. Just look through some of the built-in drivers to get an idea for how they work.
 - For details, see: `devices.interface` which defines the interface supported by device drivers and `devices.usbms` which defines a generic driver that connects to a USBMS device. All USBMS based drivers in calibre inherit from it.
- `e-books` - All the e-book conversion/metadata code. A good starting point is `calibre.ebooks.conversion.cli` which is the module powering the **ebook-convert** command. The conversion process is controlled via `conversion.plumber`. The format independent code is all in `ebooks.oeb` and the format dependent code is in `ebooks.format_name`.
 - Metadata reading, writing, and downloading is all in `ebooks.metadata`
 - Conversion happens in a pipeline, for the structure of the pipeline, see *Introduzione* (pagina 55). The pipeline consists of an input plugin, various transforms and an output plugin. The code that constructs and drives the pipeline is in `plumber.py`. The pipeline works on a representation of an e-book that is like an unzipped epub, with manifest, spine, toc, guide, html content, etc. The class that manages this representation is `OEBBook` in `ebooks.oeb.base`. The various transformations that are applied to the book during conversions live in `oeb/transforms/*.py`. And the input and output plugins live in `conversion/plugins/*.py`.
 - E-book editing happens using a different container object. It is documented in *Documentazione API per gli strumenti di modifica di un ebook* (pagina 354).
- `db` - The database back-end. See *Documentazione API per l'interfaccia del database* (pagina 345) for the interface to the calibre library.
- `Content server`: `srv` is the calibre Content server.
- `gui2` - The Graphical User Interface. GUI initialization happens in `gui2.main` and `gui2.ui`. The e-book-viewer is in `gui2.viewer`. The e-book editor is in `gui2.tweak_book`.

¹¹⁰ <https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices>

¹¹¹ <https://www.mobileread.com/forums/showthread.php?p=1362767#post1362767>

If you want to locate the entry points for all the various calibre executables, look at the `entry_points` structure in `linux.py`¹¹².

If you need help understanding the code, post in the [development forum](#)¹¹³ and you will most likely get help from one of calibre's many developers.

14.2 Ottenere il codice

You can get the calibre source code in two ways, using a version control system or directly downloading a [tarball](#)¹¹⁴.

calibre uses [Git](#)¹¹⁵, a distributed version control system. Git is available on all the platforms calibre supports. After installing Git, you can get the calibre source code with the command:

```
git clone https://github.com/kovidgoyal/calibre.git
```

On Windows you will need the complete path name, that will be something like `C:\Program Files\Git\git.exe`.

calibre is a very large project with a very long source control history, so the above can take a while (10 mins to an hour depending on your internet speed).

If you want to get the code faster, the source code for the latest release is always available as an [archive](#)¹¹⁶.

To update a branch to the latest code, use the command:

```
git pull --no-edit
```

You can also browse the code at [GitHub](#)¹¹⁷.

14.2.1 Inviare le proprie modifiche affinché siano incluse

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](#)¹¹⁸. 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](#)¹¹⁹. 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
```

¹¹² <https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.py>

¹¹³ <https://www.mobileread.com/forums/forumdisplay.php?f=240>

¹¹⁴ <https://calibre-ebook.com/dist/src>

¹¹⁵ <https://www.git-scm.com/>

¹¹⁶ <https://calibre-ebook.com/dist/src>

¹¹⁷ <https://github.com/kovidgoyal/calibre>

¹¹⁸ <https://bugs.launchpad.net/calibre>

¹¹⁹ <https://bugs.launchpad.net/calibre>

Be careful to not include merges when using `HEAD~n`.

If you plan to do a lot of development on calibre, then the best method is to create a [GitHub](#)¹²⁰ account. Below is a basic guide to setting up your own fork of calibre in a way that will allow you to submit pull requests for inclusion into the main calibre repository:

- Setup git on your machine as described in this article: [Setup Git](#)¹²¹
- Setup ssh keys for authentication to GitHub, as described here: [Generating SSH keys](#)¹²²
- Go to <https://github.com/kovidgoyal/calibre> and click the *Fork* button.
- In un terminale, eseguire:

```
git clone git@github.com:<username>/calibre.git
git remote add upstream https://github.com/kovidgoyal/calibre.git
```

Replace `<username>` above with your GitHub username. That will get your fork checked out locally.

- You can make changes and commit them whenever you like. When you are ready to have your work merged, do a:

```
git push
```

and go to <https://github.com/<username>/calibre> and click the *Pull Request* button to generate a pull request that can be merged.

- You can update your local copy with code from the main repo at any time by doing:

```
git pull upstream
```

You should also keep an eye on the calibre [development forum](#)¹²³. 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 di sviluppo Windows

Nota: You must also get the calibre source code separately as described above.

Install calibre normally, using the [Windows installer](#)¹²⁴. Then open a Command Prompt and change to the previously checked out calibre code folder. For example:

```
cd C:\Users\kovid\work\calibre
```

calibre is the folder that contains the src and resources sub-folders.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src folder. So, following the example above, it would be `C:\Users\kovid\work\calibre\src`. [Here is a short guide](#)¹²⁵ 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:

¹²⁰ <https://github.com>

¹²¹ <https://help.github.com/articles/set-up-git>

¹²² <https://help.github.com/articles/generating-ssh-keys>

¹²³ <https://www.mobilerread.com/forums/forumdisplay.php?f=240>

¹²⁴ https://calibre-ebook.com/download_windows

¹²⁵ <https://docs.python.org/using/windows.html#excursus-setting-environment-variables>


```
echo %CALIBRE_DEVELOP_FROM%
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file `src\calibre__init__.py` in your favorite editor and add the line:

```
print ("Hello, world!")
```

near the top of the file. Now run the command **calibredb**. The very first line of output should be `Hello, world!`.

You can also setup a calibre development environment inside the free Microsoft Visual Studio, if you like, following the instructions [here](#)¹²⁶.

14.4 Ambiente di sviluppo macOS

Nota: You must also get the calibre source code separately as described above.

Install calibre normally using the [provided .dmg](#)¹²⁷. 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.

Crea un file di testo semplice:

```
#!/bin/sh
export CALIBRE_DEVELOP_FROM="/Users/kovid/work/calibre/src"
calibre-debug -g
```

Salva questo file come `/usr/local/bin/calibre-develop`, quindi imposta i suoi permessi in modo da renderlo eseguibile:

```
chmod +x /usr/local/bin/calibre-develop
```

Once you have done this, run:

```
calibre-develop
```

You should see some diagnostic information in the Terminal window as calibre starts up, and you should see an asterisk after the version number in the GUI window, indicating that you are running from source.

¹²⁶ <https://www.mobileread.com/forums/showthread.php?t=251201>

¹²⁷ https://calibre-ebook.com/download_osx

14.5 Ambiente di sviluppo Linux

Nota: You must also get the calibre source code separately as described above.

calibre is primarily developed on Linux. You have two choices in setting up the development environment. You can install the calibre binary as normal and use that as a runtime environment to do your development. This approach is similar to that used in Windows and macOS. Alternatively, you can install calibre from source. Instructions for setting up a development environment from source are in the INSTALL file in the source tree. Here we will address using the binary as a runtime, which is the recommended method.

Install calibre using the [binary installer](#)¹²⁸. Then open a terminal and change to the previously checked out calibre code folder, for example:

```
cd /home/kovid/work/calibre
```

calibre is the folder that contains the src and resources sub-folders.

The next step is to set the environment variable CALIBRE_DEVELOP_FROM to the absolute path of the src folder. So, following the example above, it would be /home/kovid/work/calibre/src. How to set environment variables depends on your Linux distribution and what shell you are using.

Nota: It is recommended to use the binary installer provided from upstream. Should you insist on using a package provided by your distribution, use the CALIBRE_PYTHON_PATH and CALIBRE_RESOURCES_PATH variables instead.

Once you have set the environment variable, open a new terminal and check that it was correctly set by using the command:

```
echo $CALIBRE_DEVELOP_FROM
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file src/calibre/__init__.py in your favorite editor and add the line:

```
print ("Hello, world!")
```

near the top of the file. Now run the command **calibredb**. The very first line of output should be `Hello, world!`.

14.6 Convivenza tra due installazioni di calibre, «normale» e «di sviluppo», sullo stesso computer

The calibre source tree is very stable and rarely breaks, but if you feel the need to run from source on a separate test library and run the released calibre version with your everyday library, you can achieve this easily using .bat files or shell scripts to launch calibre. The example below shows how to do this on Windows using .bat files (the instructions for other platforms are the same, just use a shell script instead of a .bat file)

To launch the release version of calibre with your everyday library:

calibre-normal.bat:

```
calibre.exe "--with-library=C:\path\to\everyday\library folder"
```

¹²⁸ https://calibre-ebook.com/download_linux

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 Consigli per il debug

Python is a dynamically typed language with excellent facilities for introspection. Kovid wrote the core calibre code without once using a debugger. There are many strategies to debug calibre code:

14.7.1 Using print statements

This is Kovid's favorite way to debug. Simply insert print statements at points of interest and run your program in the terminal. For example, you can start the GUI from the terminal as:

```
calibre-debug -g
```

Similarly, you can start the E-book viewer as:

```
calibre-debug -w /path/to/file/to/be/viewed
```

The e-book-editor can be started as:

```
calibre-debug -t /path/to/be/edited
```

14.7.2 Using an interactive Python interpreter

You can insert the following two lines of code to start an interactive Python session at that point:

```
from calibre import ipython
ipython(locals())
```

When running from the command line, this will start an interactive Python interpreter with access to all locally defined variables (variables in the local scope). The interactive prompt even has Tab completion for object properties and you can use the various Python facilities for introspection, such as `dir()`, `type()`, `repr()`, etc.

14.7.3 Using the Python debugger as a remote debugger

You can use the builtin Python debugger (pdb) as a remote debugger from the command line. First, start the remote debugger at the point in the calibre code you are interested in, like this:

```
from calibre.rpdb import set_trace
set_trace()
```

Then run calibre, either as normal, or using one of the calibre-debug commands described in the previous section. Once the above point in the code is reached, calibre will freeze, waiting for the debugger to connect.

Now open a terminal or command prompt and use the following command to start the debugging session:

```
calibre-debug -c "from calibre.rpdb import cli; cli()"
```

You can read about how to use the Python debugger in the [Python stdlib docs for the pdb module](#)¹²⁹.

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 Using the debugger in your favorite Python IDE

It is possible to use the builtin debugger in your favorite Python IDE, if it supports remote debugging. The first step is to add the calibre src checkout to the `PYTHONPATH` in your IDE. In other words, the folder you set as `CALIBRE_DEVELOP_FROM` above, must also be in the `PYTHONPATH` of your IDE.

Then place the IDE's remote debugger module into the `src` sub-folder of the calibre source code checkout. Add whatever code is needed to launch the remote debugger to calibre at the point of interest, for example in the main function. Then run calibre as normal. Your IDE should now be able to connect to the remote debugger running inside calibre.

14.7.5 Executing arbitrary scripts in the calibre Python environment

The **calibre-debug** command provides a couple of handy switches to execute your own code, with access to the calibre modules:

```
calibre-debug -c "some Python code"
```

is great for testing a little snippet of code on the command line. It works in the same way as the `-c` switch to the Python interpreter:

```
calibre-debug myscript.py
```

can be used to execute your own Python script. It works in the same way as passing the script to the Python interpreter, except that the calibre environment is fully initialized, so you can use all the calibre code in your script. To use command line arguments with your script, use the form:

```
calibre-debug myscript.py -- --option1 arg1
```

The `--` causes all subsequent arguments to be passed to your script.

14.8 Usare calibre nei tuoi progetti

It is possible to directly use calibre functions/code in your Python project. Two ways exist to do this:

¹²⁹ <https://docs.python.org/library/pdb.html#debugger-commands>

14.8.1 Binary install of calibre

If you have a binary install of calibre, you can use the Python interpreter bundled with calibre, like this:

```
calibre-debug /path/to/your/python/script.py -- arguments to your script
```

14.8.2 Installazione da sorgenti su 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 Documentazione delle API di varie componenti di calibre

14.9.1 Documentazione API per l'interfaccia del database

Questa API è thread-safe (utilizza un lettore multiplo, con schema di blocco per singolo scrittore). Puoi accedere a questa API in questo modo

```
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

Se sei in un plugin di Calibre che è parte della GUI principale, puoi accedervi in questo modo alternativo:

```
db = self.gui.current_db.new_api
```

class `calibre.db.cache.Cache` (*backend, library_database_instance=None*)

Una cache in-memory del file `metadata.db` da una libreria calibre. Questa classe serve anche da API thread-safe per l'accesso al database. La cache in-memory è mantenuta in forma normale per le massime prestazioni.

SQLITE viene semplicemente utilizzato come modo per leggere e scrivere in modo affidabile da `metadata.db`. Tutta la logica di lettura/ordinamento/ricerca/cache della tabella viene nuovamente implementata. Ciò era necessario per la massima prestazione e flessibilità.

class `EventType` (*value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None*)

book_created = 4

Quando un nuovo libro è creato nel database, con un id libro come unico argomento

book_edited = 8

Quando il formato di un libro è modificato, con argomenti: (book_id,fmt)

books_removed = 5

Quando i libri sono rimossi dal database avente la lista di id come unico argomento

format_added = 2

Quando un formato è aggiunto ad un libro, con argomenti: (book_id, format)

formats_removed = 3

Quando dei formati sono rimossi da un libro, con argomenti: (mappatura dell' id di un libro per impostare formati rimossi dal libro)

indexing_progress_changed = 9

Quando cambia l'avanzamento dell'indicizzazione

items_removed = 7

When items such as tags or authors are removed from some books. Arguments: (field_name, affected book ids, ids of removed items)

items_renamed = 6

When items such as tags or authors are renamed in some or all books. Arguments: (field_name, affected book ids, map of old item id to new item id)

metadata_changed = 1

When some metadata is changed for some books, with arguments: (name of changed field, set of affected book ids)

add_books (*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)

Aggiunge i libri specificati alla biblioteca. I libri dovrebbero essere un iterabile di 2-tuples, ogni 2-tuple del modulo (*mi*, *format_map*) dove *mi* è un oggetto Metadata e *format_map* è un dizionario del modulo {*fmt*: *path_or_stream*} `, ad esempio :code:`{'EPUB': '/path/to/file.epub'}`.

Restituisce una coppia di elenchi: *ids*, *duplicates.ids* ' ' contiene gli ID libro per tutti i libri appena creati nel database. ``*duplicates* contiene il (*mi*, *format_map*) per tutti i libri che già esistono nel database secondo la semplice euristica di rilevamento dei duplicati usata da *has_book()* (pagina 350).

add_custom_book_data (*name*, *val_map*, *delete_first=False*)

Aggiunge i dati per nome dove *val_map* è una mappa di *book_ids* ai valori. Se *delete_first* è True, tutti i dati precedentemente memorizzati per il nome verranno rimossi.

add_extra_files (*book_id*, *map_of_relp_path_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*)

Aggiunge un formato al libro specificato. Restituisce True se il formato è stato aggiunto correttamente.

Parametri

- **replace** – Se True sostituisce il formato esistente, altrimenti se il formato esiste già, restituisce False.
- **run_hooks** – Se True, i plug-in di tipo file vengono eseguiti nel formato prima e dopo l'aggiunta.
- **dbapi** – Solo per uso interno.

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* (pagina 345), *library_id*, *event_type_specific_data*)

all_book_ids (*type=<class 'frozenset'>*)

Insieme congelato di tutti gli ID libro conosciuti.

all_field_for (*field, book_ids, default_value=None*)

Come per `field_for`, tranne per il fatto che opera su più libri contemporaneamente

all_field_ids (*name*)

Insieme di ID congelati per tutti i valori nel campo `name`.

all_field_names (*field*)

Insieme congelato di tutti i nomi dei campi (deve essere utilizzato solo per i campi many-one e many-many)

author_data (*author_ids=None*)

Restituisce i dati dell'autore come un dizionario con le chiavi: nome, ordinamento, collegamento

Se non viene trovato alcun autore con gli ID specificati, viene restituito un dizionario vuoto. Se `author_ids` è `None`, vengono restituiti i dati per tutti gli autori.

author_sort_from_authors (*authors, key_func=<function make_change_case_func.<locals>.change_case>*)

Dato un elenco di autori, restituisce la stringa `author_sort` per gli autori, preferendo l'ordinamento per autore associato all'autore rispetto alla stringa calcolata.

books_for_field (*name, item_id*)

Restituisce tutti i libri associati all'elemento identificato da `item_id`, dove l'articolo appartiene al campo `name`.

Returned value is a set of book ids, or the empty set if the item or the field does not exist.

books_in_virtual_library (*vl, search_restriction=None, virtual_fields=None*)

Return the set of books in the specified virtual library

compress_covers (*book_ids, jpeg_quality=100, progress_callback=None*)

Compress the cover images for the specified books. A compression quality of 100 will perform lossless compression, otherwise lossy compression.

The progress callback will be called with the `book_id` and the old and new sizes for each book that has been processed. If an error occurs, the new size will be a string with the error details.

copy_cover_to (*book_id, dest, use_hardlink=False, report_file_size=None*)

Copy the cover to the file like object `dest`. Returns `False` if no cover exists or `dest` is the same file as the current cover. `dest` can also be a path in which case the cover is copied to it if and only if the path is different from the current path (taking case sensitivity into account).

copy_format_to (*book_id, fmt, dest, use_hardlink=False, report_file_size=None*)

Copy the format `fmt` to the file like object `dest`. If the specified format does not exist, raises `NoSuchFormat` error. `dest` can also be a path (to a file), in which case the format is copied to it, iff the path is different from the current path (taking case sensitivity into account).

cover (*book_id, as_file=False, as_image=False, as_path=False, as_pixmap=False*)

Return the cover image or `None`. By default, returns the cover as a bytestring.

WARNING: Using `as_path` will copy the cover to a temp file and return the path to the temp file. You should delete the temp file when you are done with it.

Parametri

- **as_file** – If `True` return the image as an open file object (a `SpooledTemporaryFile`)
- **as_image** – Se `True` restituisce l'immagine come oggetto `QImage`

- **as_pixmap** – Se True restituisce l'immagine come oggetto QPixmap
- **as_path** – Se True restituisce l'immagine come un percorso che punta a un file temporaneo

data_for_find_identical_books()

Return data that can be used to implement *find_identical_books()* (pagina 348) in a worker process without access to the db. See db.utils for an implementation.

data_for_has_book()

Return data suitable for use in *has_book()* (pagina 350). This can be used for an implementation of *has_book()* (pagina 350) in a worker process without access to the db.

delete_custom_book_data(name, book_ids=())

Delete data for name. By default deletes all data, if you only want to delete data for some book ids, pass in a list of book ids.

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

fast_field_for(field_obj, book_id, default_value=None)

Same as field_for, except that it avoids the extra lookup to get the field object

field_for(name, book_id, default_value=None)

Return the value of the field name for the book identified by book_id. If no such book exists or it has no defined value for the field name or no such field exists, then default_value is returned.

default_value is not used for title, title_sort, authors, author_sort and series_index. This is because these always have values in the db. default_value is used for all custom columns.

The returned value for is_multiple fields are always tuples, even when no values are found (in other words, default_value is ignored). The exception is identifiers for which the returned value is always a dictionary. The returned tuples are always in link order, that is, the order in which they were created.

field_ids_for(name, book_id)

Return the ids (as a tuple) for the values that the field name has on the book identified by book_id. If there are no values, or no such book, or no such field, an empty tuple is returned.

find_identical_books(mi, search_restriction="", book_ids=None)

Finds books that have a superset of the authors in mi and the same title (title is fuzzy matched). See also *data_for_find_identical_books()* (pagina 348).

format(book_id, fmt, as_file=False, as_path=False, preserve_filename=False)

Return the e-book format as a bytestring or None if the format doesn't exist, or we don't have permission to write to the e-book file.

Parametri

- **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()* (pagina 347) instead for performance.
- **as_path** – Copies the format file to a temp file and returns the path to the temp file

- **preserve_filename** – If `True` and returning a path the filename is the same as that used in the library. Note that using this means that repeated calls yield the same temp file (which is re-created each time)

format_abspath (*book_id*, *fmt*)

Return absolute path to the e-book file of format *format*. You should almost never use this, as it breaks the threadsafe promise of this API. Instead use, `copy_format_to()` (pagina 347).

Currently used only in calibredb list, the viewer, edit book, compare_format to original format, open with, bulk metadata edit and the catalogs (via `get_data_as_dict()`).

Apart from the viewer, open with and edit book, I don't believe any of the others do any file write I/O with the results of this call.

format_hash (*book_id*, *fmt*)

Return the hash of the specified format for the specified book. The kind of hash is backend dependent, but is usually SHA-256.

format_metadata (*book_id*, *fmt*, *allow_cache=True*, *update_db=False*)

Return the path, size and mtime for the specified format for the specified book. You should not use path unless you absolutely have to, since accessing it directly breaks the threadsafe guarantees of this API. Instead use the `copy_format_to()` (pagina 347) method.

Parametri

- **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_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 `{}`.

Parametri

book_id – the book id in question.

Ritorna

`{field: {field_value, link_value}, ...}` for all fields with a *field_value* having a non-empty link value for that book

get_categories (*sort='name'*, *book_ids=None*, *already_fixed=None*, *first_letter_sort=False*)

Used internally to implement the Tag Browser

get_custom_book_data (*name*, *book_ids=()*, *default=None*)

Get data for *name*. By default returns data for all *book_ids*, pass in a list of book ids if you only want some data. Returns a map of *book_id* to values. If a particular value could not be decoded, uses default for it.

get_id_map (*field*)

Return a mapping of id numbers to values for the specified field. The field must be a many-one or many-many field, otherwise a `ValueError` is raised.

get_ids_for_custom_book_data (*name*)

Return the set of book ids for which name has data.

get_item_id (*field, item_name*)

Return the item id for item_name (case-insensitive) or None if not found

get_item_ids (*field, item_names*)

Return the item id for item_name (case-insensitive)

get_item_name (*field, item_id*)

Return the item name for the item specified by item_id in the specified field. See also [get_id_map\(\)](#) (pagina 349).

get_link_map (*for_field*)

Return a dictionary of links for the supplied field.

Parametri

for_field – the lookup name of the field for which the link map is desired

Ritorna

{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](#) (pagina 201) 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.

Parametri

- **field** – The series-like field (defaults to the builtin series column)
- **current_indices** – If True, returns a mapping of book_id to current series_index value instead.

get_proxy_metadata (*book_id*)

Like [get_metadata\(\)](#) (pagina 350) 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\(\)](#) (pagina 348).

has_format (*book_id, fmt*)

Restituisce True se il formato esiste sul disco

has_id (*book_id*)

Return True iff the specified book_id exists in the db

init ()

Inizializza questa cache con i dati del motore.

list_extra_files (*book_id*, *use_cache=False*, *pattern=""*) → Tuple[ExtraFile, ...]

Get information about extra files in the book's directory.

Parametri

- **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.

Ritorna

A tuple of all extra files matching the specified pattern. Each element of the tuple is Extra-File(*relpath*, *file_path*, *stat_result*). Where *relpath* is the relative path of the file to the book directory using / as a separator. *stat_result* is the result of calling *os.stat()* on the file.

merge_extra_files (*dest_id*, *src_ids*, *replace=False*)

Merge the extra files from *src_ids* into *dest_id*. Conflicting files are auto-renamed unless *replace=True* in which case they are replaced.

move_book_from_trash (*book_id*)

Undelete a book from the trash directory

move_format_from_trash (*book_id*, *fnt*)

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.

pref (*name*, *default=None*, *namespace=None*)

Return the value for the specified preference or the value specified as *default* if the preference is not set.

read_backup (*book_id*)

Return the OPF metadata backup for the book as a bytestring or *None* if no such backup exists.

remove_books (*book_ids*, *permanent=False*)

Remove the books specified by the *book_ids* from the database and delete their format files. If *permanent* is *False*, then the format files are placed in the per-library trash directory.

remove_formats (*formats_map*, *db_only=False*)

Remove the specified formats from the specified books.

Parametri

- **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.

Ritorna

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.

Parametri

- **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=()*)

Ripristina la voce del libro nel database per un libro che esiste già nel 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.

Parametri

- **restriction** – A restriction that is ANDed to the specified query. Note that restrictions are cached, therefore the search for a AND b will be slower than a with restriction b.
- **virtual_fields** – Used internally (virtual fields such as on_device to search over).
- **book_ids** – If not None, a set of book ids for which books will be searched instead of searching all books.

set_conversion_options (*options, fmt='PIPE'*)

options must be a map of the form {book_id:conversion_options}

set_cover (*book_id_data_map*)

Imposta la copertina per questo libro. Il dato può essere QImage, QPixmap, un oggetto file o una stringa di byte. Può anche essere None, nel qual caso ogni copertina esistente viene rimossa.

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.

Parametri

- **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** – Utilizzato internamente, non dovresti mai modificarlo.

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`

Parametri

- **field** – the lookup name
- **value_to_link_map** – dict(field_value:link, ...). Note that these are values, not field ids.

Ritorna

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_pref (*name, val, namespace=None*)

Set the specified preference to the specified value. See also `pref()` (pagina 351).

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.

Parametri

- **delta** – A timedelta object or None. If None, then all ids with the tag are returned.
- **must_have_tag** – If not None the list of matches will be restricted to books that have this tag
- **must_have_authors** – A list of authors. If not None the list of matches will be restricted to books that have these authors (case insensitive).

user_categories_for_books (*book_ids, proxy_metadata_map=None*)

Return the user categories for the specified books. `proxy_metadata_map` is optional and is useful for a performance boost, in contexts where a ProxyMetadata object for the books already exists. It should be a mapping of `book_ids` to their corresponding ProxyMetadata objects.

14.9.2 Documentazione API per gli strumenti di modifica di un ebook

Gli strumenti di modifica dell'e-book sono costituiti da un oggetto `calibre.ebooks.oeb.polish.container.Container` (pagina 354) che rappresenta un libro come una raccolta di HTML + file di risorse e vari strumenti che possono essere utilizzati per eseguire operazioni sul container. Tutti gli strumenti sono sotto forma di funzioni a livello di modulo nei vari moduli `calibre.ebooks.oeb.polish.*`. Ottieni un oggetto contenitore per un libro in un percorso come questo:

```
from calibre.ebooks.oeb.polish.container import get_container
container = get_container('Path to book file', tweak_mode=True)
```

Se stai scrivendo un plugin per l'editor di e-book, ottieni il contenitore corrente per il libro che viene modificato in questo modo:

```
from calibre.gui2.tweak_book import current_container
container = current_container()
if container is None:
    report_error # No book has been opened yet
```

L'oggetto Contenitore

```
class calibre.ebooks.oeb.polish.container.Container (rootpath, opfpath, log,
                                                    clone_data=None)
```

A container represents an open e-book as a folder full of files and an OPF file. There are two important concepts:

- The root folder. This is the base of the e-book. All the e-books files are inside this folder or in its sub-folders.
- Names: These are paths to the books' files relative to the root folder. They always contain POSIX separators and are unquoted. They can be thought of as canonical identifiers for files in the book. Most methods on the container object work with names. Names are always in the NFC Unicode normal form.
- Clones: the container object supports efficient on-disk cloning, which is used to implement checkpoints in the e-book editor. In order to make this work, you should never access files on the filesystem directly. Instead, use `raw_data()` (pagina 357) or `open()` (pagina 356) to read/write to component files in the book.

When converting between hrefs and names use the methods provided by this class, they assume all hrefs are quoted.

abspath_to_name (*fullpath*, *root=None*)

Convert an absolute path to a canonical name relative to *root*

Parametri

root – The base folder. By default the root for this container object is used.

add_file (*name*, *data*, *media_type=None*, *spine_index=None*, *modify_name_if_needed=False*, *process_manifest_item=None*)

Add a file to this container. Entries for the file are automatically created in the OPF manifest and spine (if the file is a text document)

add_name_to_manifest (*name*, *process_manifest_item=None*)

Add an entry to the manifest for a file with the specified name. Returns the manifest id.

add_properties (*name*, **properties*)

Add the specified properties to the manifest item identified by name.

apply_unique_properties (*name*, **properties*)

Ensure that the specified properties are set on only the manifest item identified by name. You can pass None as the name to remove the property from all items.

book_type = 'oeb'

Il tipo di libro (epub per i file EPUB e azw3 per i file AZW3)

commit (*outpath=None*, *keep_parsed=False*)

Commit all dirtied parsed objects to the filesystem and write out the e-book file at outpath.

Parametri

- **output** – The path to write the saved e-book file to. If None, the path of the original book file is used.
- **keep_parsed** – If True the parsed representations of committed items are kept in the cache.

commit_item (*name*, *keep_parsed=False*)

Commit a parsed object to disk (it is serialized and written to the underlying file). If *keep_parsed* is True the parsed representation is retained in the cache. See also: [parsed\(\)](#) (pagina 357)

dirty (*name*)

Mark the parsed object corresponding to name as dirty. See also: [parsed\(\)](#) (pagina 357).

exists (*name*)

True iff a file/folder corresponding to the canonical name exists. Note that this function suffers from the limitations of the underlying OS filesystem, in particular case (in)sensitivity. So on a case insensitive filesystem this will return True even if the case of name is different from the case of the underlying filesystem file. See also [has_name\(\)](#) (pagina 355)

filesize (*name*)

Return the size in bytes of the file represented by the specified canonical name. Automatically handles dirtied parsed objects. See also: [parsed\(\)](#) (pagina 357)

generate_item (*name*, *id_prefix=None*, *media_type=None*, *unique_href=True*)

Add an item to the manifest with href derived from the given name. Ensures uniqueness of href and id automatically. Returns generated item.

get_file_path_for_processing (*name*, *allow_modification=True*)

Similar to open() except that it returns a file path, instead of an open file object.

property_guide_type_map

Mapping of guide type to canonical name

has_name (*name*)

Return True iff a file with the same canonical name as that specified exists. Unlike [exists\(\)](#) (pagina 355) this method is always case-sensitive.

href_to_name (*href*, *base=None*)

Convert an href (relative to base) to a name. base must be a name or None, in which case self.root is used.

insert_into_xml (*parent*, *item*, *index=None*)

Insert item into parent (or append if index is None), fixing indentation. Only works with self closing items.

is_dir = False

Se questo contenitore rappresenta un libro estratto (una cartella)

iterlinks (*name*, *get_line_numbers=True*)

Iterate over all links in *name*. If *get_line_numbers* is *True* the yields results of the form (link, line_number, offset). Where line_number is the line_number at which the link occurs and offset is the number of characters from the start of the line. Note that offset could actually encompass several lines if not zero.

make_name_unique (*name*)

Ensure that *name* does not already exist in this book. If it does, return a modified version that does not exist.

manifest_has_name (*name*)

Return *True* if the manifest has an entry corresponding to *name*

property manifest_id_map

Mapping of manifest id to canonical names

manifest_items_of_type (*predicate*)

The names of all manifest items whose media-type matches *predicate*. *predicate* can be a set, a list, a string or a function taking a single argument, which will be called with the media-type.

manifest_items_with_property (*property_name*)

All manifest items that have the specified property

property manifest_type_map

Mapping of manifest media-type to list of canonical names of that media-type

property mi

The metadata of this book as a Metadata object. Note that this object is constructed on the fly every time this property is requested, so use it sparingly.

name_to_abspath (*name*)

Convert a canonical name to an absolute OS dependent path

name_to_href (*name*, *base=None*)

Convert a name to a href relative to *base*, which must be a name or *None* in which case *self.root* is used as the base

property names_that_must_not_be_changed

Set of names that must never be renamed. Depends on the e-book file format.

property names_that_must_not_be_removed

Set of names that must never be deleted from the container. Depends on the e-book file format.

property names_that_need_not_be_manifested

Set of names that are allowed to be missing from the manifest. Depends on the e-book file format.

open (*name*, *mode='rb'*)

Open the file pointed to by *name* for direct read/write. Note that this will commit the file if it is dirtied and remove it from the parse cache. You must finish with this file before accessing the parsed version of it again, or bad things will happen.

property opf

Il file OPF elaborato

opf_get_or_create (*name*)

Convenience method to either return the first XML element with the specified name or create it under the *opf:package* element and then return it, if it does not already exist.

property opf_version

The version set on the OPF's <package> element

property opf_version_parsed

The version set on the OPF's <package> element as a tuple of integers

opf_xpath (*expr*)

Convenience method to evaluate an XPath expression on the OPF file, has the opf: and dc: namespace prefixes pre-defined.

parsed (*name*)

Return a parsed representation of the file specified by name. For HTML and XML files an lxml tree is returned. For CSS files a css_parser stylesheet is returned. Note that parsed objects are cached for performance. If you make any changes to the parsed object, you must call [dirty\(\)](#) (pagina 355) so that the container knows to update the cache. See also [replace\(\)](#) (pagina 357).

raw_data (*name*, *decode=True*, *normalize_to_nfc=True*)

Return the raw data corresponding to the file specified by name

Parametri

- **decode** – If True and the file has a text based MIME type, decode it and return a unicode object instead of raw bytes.
- **normalize_to_nfc** – If True the returned unicode object is normalized to the NFC normal form as is required for the EPUB and AZW3 file formats.

relpath (*path*, *base=None*)

Convert an absolute path (with os separators) to a path relative to base (defaults to self.root). The relative path is *not* a name. Use [abspath_to_name\(\)](#) (pagina 354) for that.

remove_from_spine (*spine_items*, *remove_if_no_longer_in_spine=True*)

Remove the specified items (by canonical name) from the spine. If *remove_if_no_longer_in_spine* is True, the items are also deleted from the book, not just from the spine.

remove_from_xml (*item*)

Removes item from parent, fixing indentation (works only with self closing items)

remove_item (*name*, *remove_from_guide=True*)

Remove the item identified by name from this container. This removes all references to the item in the OPF manifest, guide and spine as well as from any internal caches.

rename (*current_name*, *new_name*)

Renames a file from *current_name* to *new_name*. It automatically rebases all links inside the file if the folder the file is in changes. Note however, that links are not updated in the other files that could reference this file. This is for performance, such updates should be done once, in bulk.

replace (*name*, *obj*)

Replace the parsed object corresponding to name with obj, which must be a similar object, i.e. an lxml tree for HTML/XML or a css_parser stylesheet for a CSS file.

replace_links (*name*, *replace_func*)

Replace all links in name using *replace_func*, which must be a callable that accepts a URL and returns the replaced URL. It must also have a “replaced” attribute that is set to True if any actual replacement is done. Convenient ways of creating such callables are using the `LinkReplacer` and `LinkRebaser` classes.

serialize_item (*name*)

Convert a parsed object (identified by canonical name) into a bytestring. See [parsed\(\)](#) (pagina 357).

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* (pagina 358) and *spine_items* (pagina 358).

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* (pagina 358) and *spine_items* (pagina 358).

property spine_names

An iterator yielding name and is_linear for every item in the books' spine. See also: *spine_iter* (pagina 358) and *spine_items* (pagina 358).

Gestione dei file dei componenti in un contenitore

`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.

Parametri

- **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.

Parametri

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.

Errori di formattazione e correzione automatica

`calibre.ebooks.oeb.polish.pretty.fix_html (container, raw)`

Fix any parsing errors in the HTML represented as a string in raw. Fixing is done using the HTML5 parsing algorithm.

`calibre.ebooks.oeb.polish.pretty.fix_all_html (container)`

Fix any parsing errors in all HTML files in the container. Fixing is done using the HTML5 parsing algorithm.

`calibre.ebooks.oeb.polish.pretty.pretty_html (container, name, raw)`

Pretty print the HTML represented as a string in raw

`calibre.ebooks.oeb.polish.pretty.pretty_css (container, name, raw)`

Pretty print the CSS represented as a string in raw

`calibre.ebooks.oeb.polish.pretty.pretty_xml (container, name, raw)`

Pretty print the XML represented as a string in raw. If name is the name of the OPF, extra OPF-specific prettying is performed.

`calibre.ebooks.oeb.polish.pretty.pretty_all (container)`

Pretty print all HTML/CSS/XML files in the container

Gestione delle copertine dei libri

`calibre.ebooks.oeb.polish.jacket.remove_jacket (container)`

Remove an existing jacket, if any. Returns False if no existing jacket was found.

`calibre.ebooks.oeb.polish.jacket.add_or_replace_jacket (container)`

Either create a new jacket from the book's metadata or replace an existing jacket. Returns True if an existing jacket was replaced.

Divisione e unione dei file

`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.

Parametri

- **loc_or_xpath** – Should be an XPath expression such as `//h:div[@id=»split_here»]`. Can also be a *loc* which is used internally to implement splitting in the preview panel.
- **before** – If True the split occurs before the identified element otherwise after it.
- **totals** – Utilizzato 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\(\)](#) (pagina 359). Splitting automatically migrates all links and references to the affected files.

Parametri

- **before** – If True the splits occur before the identified element otherwise after it.

`calibre.ebooks.oeb.polish.split.merge` (*container, category, names, master*)

Merge the specified files into a single file, automatically migrating all links and references to the affected files. The file must all either be HTML or CSS files.

Parametri

- **category** – Must be either 'text' for HTML files or 'styles' for CSS files
- **names** – L'elenco dei file da unire
- **master** – Which of the merged files is the *master* file, that is, the file that will remain after merging.

Gestione delle copertine

`calibre.ebooks.oeb.polish.cover.set_cover` (*container, cover_path, report=None, options=None*)

Set the cover of the book to the image pointed to by `cover_path`.

Parametri

- **cover_path** – Either the absolute path to an image file or the canonical name of an image in the book. When using an image in the book, you must also set options, see below.
- **report** – An optional callable that takes a single argument. It will be called with information about the tasks being processed.
- **options** – None or a dictionary that controls how the cover is set. The dictionary can have entries: **keep_aspect**: True or False (Preserve aspect ratio of covers in EPUB) **no_svg**: True or False (Use an SVG cover wrapper in the EPUB titlepage) **existing**: True or False (`cover_path` refers to an existing image in the book)

`calibre.ebooks.oeb.polish.cover.mark_as_cover` (*container, name*)

Mark the specified image as the cover image.

`calibre.ebooks.oeb.polish.cover.mark_as_titlepage` (*container, name, move_to_start=True*)

Mark the specified HTML file as the titlepage of the EPUB.

Parametri

move_to_start – If True the HTML file is moved to the start of the spine

Lavorare con i 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.

Parametri

- **report** – An optional callable that takes a single argument. It is called with information about the operations being performed.

- **remove_unused_classes** – If True, class attributes in the HTML that do not match any CSS rules are also removed.
- **merge_rules** – If True, rules with identical selectors are merged.
- **merge_rules_with_identical_properties** – If True, rules with identical properties are merged.
- **remove_unreferenced_sheets** – If True, stylesheets that are not referenced by any content are removed

`calibre.ebooks.oeb.polish.css.filter_css(container, properties, names=())`

Remove the specified CSS properties from all CSS rules in the book.

Parametri

- **properties** – Set of properties to remove. For example: {'font-family', 'color'}.
- **names** – The files from which to remove the properties. Defaults to all HTML and CSS files in the book.

Al lavoro con la tabella dei contenuti

`calibre.ebooks.oeb.polish.toc.from_xpaths(container, xpaths, prefer_title=False)`

Generate a Table of Contents from a list of XPath expressions. Each expression in the list corresponds to a level of the generate ToC. For example: ['//h:h1', '//h:h2', '//h:h3'] will generate a three level Table of Contents from the <h1>, <h2> and <h3> tags.

`calibre.ebooks.oeb.polish.toc.from_links(container)`

Generate a Table of Contents from links in the book.

`calibre.ebooks.oeb.polish.toc.from_files(container)`

Generate a Table of Contents from files in the book.

`calibre.ebooks.oeb.polish.toc.create_inline_toc(container, title=None)`

Create an inline (HTML) Table of Contents from an existing NCX Table of Contents.

Parametri

- title** – The title for this table of contents.

Strumento modifica libro

class `calibre.gui2.tweak_book.plugin.Tool`

Basi: object

The base class for individual tools in an Edit Book plugin. Useful members include:

- `self.plugin`: A reference to the `calibre.customize.Plugin` (pagina 248) object to which this tool belongs.
- `self.boss` (pagina 362)
- `self.gui` (pagina 362)

Methods that must be overridden in sub classes:

- `create_action()` (pagina 362)

- `register_shortcut()` (pagina 362)

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` (pagina 363) object. Used to control the user interface.

property gui

La finestra principale dell'interfaccia utente

property current_container

Return the current `calibre.ebooks.oeb.polish.container.Container` (pagina 354) object that represents the book being edited.

register_shortcut (*qaction, unique_name, default_keys=(), short_text=None, description=None, **extra_data*)

Register a keyboard shortcut that will trigger the specified `qaction`. This keyboard shortcut will become automatically customizable by the user in the Keyboard shortcuts section of the editor preferences.

Parametri

- **qaction** – A QAction object, it will be triggered when the configured key combination is pressed by the user.
- **unique_name** – A unique name for this shortcut/action. It will be used internally, it must not be shared by any other actions in this plugin.
- **default_keys** – A list of the default keyboard shortcuts. If not specified no default shortcuts will be set. If the shortcuts specified here conflict with either builtin shortcuts or shortcuts from user configuration/other plugins, they will be ignored. In that case, users will have to configure the shortcuts manually via Preferences. For example: `default_keys=('Ctrl+J', 'F9')`.
- **short_text** – An optional short description of this action. If not specified the text from the QAction will be used.
- **description** – An optional longer description of this action, it will be used in the preferences entry for this shortcut.

create_action (*for_toolbar=True*)

Create a QAction that will be added to either the plugins toolbar or the plugins menu depending on `for_toolbar`. For example:

```
def create_action(self, for_toolbar=True):
    ac = QAction(get_icons('myicon.png'), 'Do something')
    if for_toolbar:
        # We want the toolbar button to have a popup menu
        menu = QMenu()
```

(continues on next page)

(continua dalla pagina precedente)

```

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

```

Vedi anche:Method `register_shortcut()` (pagina 362).

Controllare l'interfaccia dell'editor

L'interfaccia utente dell'editor di e-book è controllata da un singolo oggetto globale *Boss*. Questo ha molti metodi utili che possono essere utilizzati nel codice del plugin per eseguire attività comuni.

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.

Parametri

mark_as_modified – If True, the book will be marked as modified, so the user will be prompted to save it when quitting.

close_editor (*name*)

Close the editor that is editing the file specified by *name*

commit_all_editors_to_container ()

Commit any changes that the user has made to files open in editors to the container. You should call this method before performing any actions on the current container

property currently_editing

Return the name of the file being edited currently or None if no file is being edited

edit_file (*name*, *syntax=None*, *use_template=None*)

Open the file specified by *name* in an editor

Parametri

- **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.

Parametri

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 ()

Salvare il libro. Il salvataggio è eseguito sullo sfondo

set_modified ()

Segna il libro come Modificato

show_current_diff (*allow_revert=True, to_container=None*)

Show the changes to the book from its last checkpointed state

Parametri

- **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*)

Mostra l'editore che sta modificando il file specificato per *nome*

sync_preview_to_editor ()

Sync the position of the preview panel to the current cursor position in the current editor

Gestione dei Diritti Digitali (DRM)

La gestione dei diritti digitali (DRM) è un termine generico per le tecnologie di controllo degli accessi che possono essere utilizzate da produttori di hardware, editori, detentori del copyright e privati per tentare di imporre limiti all'uso di contenuti e dispositivi digitali. A volte è anche denigrata con l'espressione Gestione delle Restrizioni Digitali. Il termine è usato per descrivere qualsiasi tecnologia che inibisca gli usi (legittimi o meno) di contenuti digitali che non siano stati desiderati o previsti dal fornitore di contenuti. Il termine generalmente non si riferisce ad altre forme di protezione dalla copia che possono essere eluse senza modificare il file o il dispositivo, come numeri di serie o file di chiave. Può anche fare riferimento a restrizioni associate a casi specifici di opere o dispositivi digitali. Le tecnologie DRM tentano di controllare l'uso dei media digitali impedendo l'accesso, la copia o la conversione in altri formati da parte degli utenti finali. Vedi *Wikipedia* <https://it.wikipedia.org/wiki/Digital_rights_management> _.

15.1 Cosa implica DRM per me personalmente?

Quando acquisti un e-book con DRM non lo possiedi davvero, ma hai acquistato l'autorizzazione per usarlo in un modo dettato dal venditore. DRM limita ciò che puoi fare con gli e-book che hai «acquistato». Spesso le persone che acquistano libri con DRM non sono consapevoli della portata di queste restrizioni. Queste restrizioni ti impediscono di riformattare l'e-book a tuo piacimento, anche apportando modifiche stilistiche come la regolazione delle dimensioni dei caratteri, anche se esiste un software che ti consente di fare tali cose per i libri non DRM. Le persone sono spesso sorprese che un e-book che hanno acquistato in un particolare formato non possa essere convertito in un altro formato se l'e-book ha il DRM. Quindi, se hai un Kindle di Amazon e acquisti un libro venduto da Barnes and Nobles, dovresti sapere che se quell'e-book ha il DRM non sarai in grado di leggerlo sul tuo Kindle. Si noti che si parla di un libro che hai acquistato, non rubato o piratato ma ACQUISTATO.

15.2 Cosa fa DRM per gli autori?

Gli editori di e-book con DRM sostengono che il DRM è tutto per il bene degli autori e per proteggere la loro integrità artistica e prevenire la pirateria. Ma il DRM NON impedisce la pirateria. Le persone che vogliono piratare contenuti o utilizzare contenuti piratati lo fanno ancora e hanno successo. I tre principali schemi DRM per e-book oggi sono gestiti da Amazon, Adobe e Barnes e Noble e tutti e tre gli schemi DRM sono stati decifrati. Tutto ciò che DRM fa è complicare la vita agli utenti legittimi. Si può supporre che in realtà danneggia gli autori, in quanto le persone che avrebbero acquistato il libro scelgono di cercare una versione piratata in quanto non sono disposti a sopportare il DRM. Coloro che piraterebbero in assenza di DRM lo fanno anche in sua presenza. Per ribadire, il punto chiave è che DRM *non impedisce la pirateria*. Quindi il DRM non è solo inutile e dannoso per gli acquirenti di e-book, ma anche uno spreco di denaro.

15.3 DRM e libertà

Sebbene i contenuti digitali possano essere utilizzati per rendere le informazioni e le opere creative facilmente accessibili a tutti ed emancipare l'umanità, ciò non è nell'interesse di alcuni editori che vogliono allontanare le persone da questa possibilità di libertà, semplicemente per mantenere la loro rilevanza in un mondo in un'evoluzione così rapida che non riescono a tenere il passo.

15.4 Perché calibre non supporta DRM?

calibre è un software a sorgente aperta mentre il DRM è, per sua stessa natura, chiuso. Se calibre dovesse supportare l'apertura o la visualizzazione di file con DRM, potrebbe essere facilmente modificato per essere utilizzato come strumento di rimozione del DRM, il che è illegale secondo le leggi odierne. Il software open source e il DRM seguono principi opposti. Mentre il DRM si basa sul controllo dell'utente, il software open source vuole lasciare il potere all'utente. Le due cose semplicemente non possono coesistere.

15.5 Qual è l'opinione di Calibre sui fornitori di contenuti?

Crediamo fermamente che gli autori e gli altri fornitori di contenuti debbano essere compensati per i loro sforzi, ma il DRM non è la strada da percorrere. Stiamo sviluppando questo database di e-book senza DRM da varie fonti per aiutarti a trovare alternative prive di DRM e per aiutare autori ed editori indipendenti di e-book senza DRM a pubblicizzare i loro contenuti. Ci auguriamo che lo troverai utile e ti chiediamo di non piratare i contenuti messi a tua disposizione qui.

15.6 Come posso aiutare a combattere il DRM?

Come persona che legge e acquista e-book puoi aiutare a combattere il DRM. Non acquistare e-book con DRM. Ci sono alcuni editori che pubblicano e-book senza DRM. Fai uno sforzo per vedere se vendono l'e-book che stai cercando. Se ti piacciono i libri di alcuni autori indipendenti che vendono e-book senza DRM e ne hai la possibilità, fai loro delle donazioni. Si tratta di soldi ben spesi in quanto i loro e-book tendono ad essere più economici (potrebbero esserci delle eccezioni) rispetto a quelli che compreresti dagli editori di libri con DRM e probabilmente funzioneranno su tutti i dispositivi che possiederai in futuro, permettendoti di risparmiare il costo del riacquisto dello stesso e-book. Non scoraggiare editori e autori di e-book privi di DRM piratandone il contenuto. I fornitori di contenuti meritano di essere ricompensati per i loro sforzi. Non punirli per aver cercato di migliorare la tua esperienza di lettura rendendo disponibili libri elettronici senza DRM. A lungo termine, questo è dannoso per te. Se hai acquistato libri da venditori che dispongono sia di libri con DRM sia di libri privi di DRM, non sapendo se erano dotati di DRM o meno, fai in modo di lasciare un commento o una recensione sul sito Web per informare i futuri acquirenti dello stato del DRM. Molti venditori non ritengono importante

indicare chiaramente ai propri acquirenti se un e-book rechi o meno un DRM. Qui¹³⁰ troverai una guida per vivere senza DRM.

¹³⁰ <https://www.defectivebydesign.org/guide/ebooks>

RSS

RSS (*Really Simple Syndication*) è un formato di feed web usato per pubblicare contenuti aggiornati frequentemente, come notizie, post sui blog, ecc. È un formato studiato per essere letto sui computer ed è quindi il sistema migliore per ricevere in ebook i contenuti del web. Ci sono molti altri formati di feed usati su internet e calibre è compatibile con la maggior parte di essi. In particolare ha un buon supporto per il formato *ATOM*, spesso usato per i blog.

Ricetta

Una ricetta è un set di istruzioni che insegna a calibre come convertire una sorgente di notizie online, come un magazine o un blog, in e-book. Una ricetta è, in sostanza, codice *Python*¹³¹. Pertanto è possibile convertire arbitrariamente delle sorgenti complesse in e-book. Semplificando, è un insieme di variabili, come gli URL, che forniscono a calibre informazioni sufficienti per andare in internet e scaricare le notizie.

HTML

HTML (*Hyper Text Mark-Up Language*), è un sottoinsieme dello Standard Generalized Mark-Up Language (SGML) per la pubblicazione elettronica. È lo standard specifico usato nel World Wide Web.

CSS

CSS (*Cascading Style Sheets*) è un linguaggio usato per descrivere come un documento *HTML* deve essere renderizzato (stile di visualizzazione).

API

API (*Application Programming Interface*) è un'interfaccia del codice sorgente che viene fornita da una libreria per permettere ad altri programmi di eseguire richieste di servizio.

LRF

LRF Il formato di e-book usato dai lettori di e-book SONY.

URL

URL (*Uniform Resource Locator*) ad esempio: `http://example.com`

regex

Le **espressioni regolari** forniscono un metodo rapido e flessibile per identificare stringhe di testo di interesse, come ad esempio caratteri, parole o sequenze di caratteri particolari. Vedi *la guida* (pagina 205) per un'introduzione alle espressioni regolari.

¹³¹ <https://www.python.org>

C

- `calibre.customize`, 248
- `calibre.customize.conversion`, 258
- `calibre.db.cache`, 345
- `calibre.devices.interface`, 260
- `calibre.ebooks.metadata.book.base`, 201
- `calibre.ebooks.metadata.sources.base`, 254
- `calibre.ebooks.oeb.polish.container`, 354
- `calibre.ebooks.oeb.polish.cover`, 360
- `calibre.ebooks.oeb.polish.css`, 360
- `calibre.ebooks.oeb.polish.jacket`, 359
- `calibre.ebooks.oeb.polish.pretty`, 359
- `calibre.ebooks.oeb.polish.replace`, 358
- `calibre.ebooks.oeb.polish.split`, 359
- `calibre.ebooks.oeb.polish.toc`, 361
- `calibre.gui2.tweak_book.boss`, 363
- `calibre.utils.formatter_functions`, 181
- `calibre.web.feeds.news`, 37

Simboli

- l calibredb-add command line option, 308
- H ebook-polish command line option, 330
- I calibredb-add command line option, 307
fetch-ebook-metadata command line option, 332
- S calibredb-add command line option, 308
- T calibredb-add command line option, 308
- U ebook-polish command line option, 331
- access-log
calibre-server command line option, 299
- add
calibredb-add command line option, 308
- add-plugin
calibre-customize command line option, 296
- add-simple-plugin
calibre-debug command line option, 297
- add-soft-hyphens
ebook-polish command line option, 330
- ajax-timeout
calibre-server command line option, 299
- all
calibredb-backup_metadata command line option, 316
- calibredb-export command line option, 310
- allowed-plugin
fetch-ebook-metadata command line option, 332
- append
calibredb-set_custom command line option, 314
- as-extra-data-file
calibredb-add_format command line option, 309
- as-opf
calibredb-show_metadata command line option, 309
- ascending
calibredb-list command line option, 306
- asciiize
ebook-convert command line option, 321
- attachment
calibre-smtp command line option, 303
- auth-mode
calibre-server command line option, 299
- author-sort
ebook-convert command line option, 326
ebook-meta command line option, 328
- authors
calibredb-add command line option, 307
ebook-convert command line option, 326
ebook-meta command line option, 328
fetch-ebook-metadata command line option, 332
- auto-reload
calibre-server command line option, 299

--automerge
 calibredb-add command line option, 307
 --ban-after
 calibre-server command line option, 299
 --ban-for
 calibre-server command line option, 299
 --base-dir
 web2disk command line option, 335
 --base-font-size
 ebook-convert command line option, 321
 --book-list-mode
 calibre-server command line option, 299
 --book-producer
 ebook-convert command line option, 326
 ebook-meta command line option, 328
 --build-plugin
 calibre-customize command line option, 296
 --cafile
 calibre-smtp command line option, 304
 --catalog-title
 calibredb-catalog command line option, 311
 --categories
 calibredb-list_categories command line option, 315
 --category
 ebook-meta command line option, 328
 --change-justification
 ebook-convert command line option, 321
 --chapter
 ebook-convert command line option, 325
 --chapter-mark
 ebook-convert command line option, 325
 --command
 calibre-debug command line option, 297
 --comments
 ebook-convert command line option, 326
 ebook-meta command line option, 329
 --compress-images
 ebook-polish command line option, 330
 --compress-min-size
 calibre-server command line option, 300
 --continue
 ebook-viewer command line option, 331
 --cover
 calibredb-add command line option, 307
 ebook-convert command line option, 327
 ebook-meta command line option, 329
 ebook-polish command line option, 330
 fetch-ebook-metadata command line option, 332
 --cross-reference-authors
 calibredb-catalog command line option, 311
 --csv
 calibredb-check_library command line option, 315
 calibredb-list_categories command line option, 315
 --custom-list-template
 calibre-server command line option, 300
 --customize-plugin
 calibre-customize command line option, 296
 --daemonize
 calibre-server command line option, 300
 --date
 ebook-meta command line option, 329
 --debug-device-driver
 calibre-debug command line option, 297
 --debug-pipeline
 calibredb-catalog command line option, 311
 ebook-convert command line option, 327
 --default-programs
 calibre-debug command line option, 297
 --delay
 web2disk command line option, 335
 --detach
 calibre command line option, 296
 ebook-edit command line option, 328
 ebook-viewer command line option, 331
 --details
 calibredb-custom_columns command line option, 314
 --dialect

calibredb-list_categories	command line option, 316	ebook-convert	command line option, 324
--diff		--disable-use-bonjour	calibre-server command line option, 300
calibre-debug	command line option, 297	--disable-use-sendfile	calibre-server command line option, 300
--disable-allow-socket-preallocation		--display	calibredb-add_custom_column command line option, 313
calibre-server	command line option, 300	--displayed-fields	calibre-server command line option, 300
--disable-auth		--do-not-match-on-related-words	calibredb-fts_search command line option, 318
calibre-server	command line option, 300	--dont-asciiize	calibredb-export command line option, 310
--disable-dehyphenate		--dont-download-stylesheets	web2disk command line option, 335
ebook-convert	command line option, 324	--dont-output-resources	lrf2lrs command line option, 333
--disable-delete-blank-paragraphs		--dont-replace	calibredb-add_format command line option, 309
ebook-convert	command line option, 324	--dont-save-cover	calibredb-export command line option, 310
--disable-fallback-to-detected-interface		--dont-save-extra-files	calibredb-export command line option, 310
calibre-server	command line option, 300	--dont-split-on-page-breaks	ebook-convert command line option, 320
--disable-fix-indents		--dont-update-metadata	calibredb-export command line option, 310
ebook-convert	command line option, 324	--dont-verify-server-certificate	calibre-smtp command line option, 304
--disable-font-rescaling		--dont-write-opf	calibredb-export command line option, 310
ebook-convert	command line option, 321	--duplicate-links-in-toc	ebook-convert command line option, 326
--disable-format-scene-breaks		--duplicates	calibredb-add command line option, 307
ebook-convert	command line option, 324	--edit-book	calibre-debug command line option, 297
--disable-hyphenation		--embed-all-fonts	
lrfviewer	command line option, 334		
--disable-italicize-common-cases			
ebook-convert	command line option, 324		
--disable-local-write			
calibre-server	command line option, 300		
--disable-log-not-found			
calibre-server	command line option, 300		
--disable-markup-chapter-headings			
ebook-convert	command line option, 324		
--disable-plugin			
calibre-customize	command line option, 296		
--disable-remove-fake-margins			
ebook-convert	command line option, 325		
--disable-renumber-headings			
ebook-convert	command line option, 324		
--disable-unwrap-lines			

ebook-convert command line option, 321	--epub-version ebook-convert command line option, 320
--embed-font-family ebook-convert command line option, 321	--exclude-genre calibredb-catalog command line option, 312
--embed-fonts ebook-polish command line option, 330	--exclusion-rules calibredb-catalog command line option, 312
--empty calibredb-add command line option, 307	--exec-file calibre-debug command line option, 298
--enable-allow-socket-preallocation calibre-server command line option, 300	--expand-css ebook-convert command line option, 321
--enable-auth calibre-server command line option, 300	--explode-book calibre-debug command line option, 298
--enable-fallback-to-detected-interface calibre-server command line option, 300	--export-all-calibre-data calibre-debug command line option, 298
--enable-heuristics ebook-convert command line option, 324	--extra-css ebook-convert command line option, 322
--enable-local-write calibre-server command line option, 300	--extract-to ebook-convert command line option, 320
--enable-log-not-found calibre-server command line option, 300	--field calibredb-set_metadata command line option, 309
--enable-plugin calibre-customize command line option, 296	--fields calibredb-list command line option, 306
--enable-use-bonjour calibre-server command line option, 300	--filter-css ebook-convert command line option, 322
--enable-use-sendfile calibre-server command line option, 300	--filter-regexp web2disk command line option, 335
--encoding web2disk command line option, 335	--fix-multiprocessing calibre-debug command line option, 298
--encryption-method calibre-smtp command line option, 304	--flow-size ebook-convert command line option, 320
--epub-flatten ebook-convert command line option, 320	--font-size-mapping ebook-convert command line option, 322
--epub-inline-toc ebook-convert command line option, 320	--for-machine calibredb-list command line option, 306
--epub-max-image-size ebook-convert command line option, 320	--force calibredb-remove_custom_column command line option, 314
--epub-toc-at-end ebook-convert command line option, 320	--force-reload

ebook-viewer command line option, 331
 --fork
 calibre-smtp command line option, 303
 --formats
 calibredb-export command line option, 310
 --from-opf
 ebook-convert command line option, 327
 ebook-meta command line option, 329
 --full-screen
 ebook-viewer command line option, 331
 --fullscreen
 ebook-viewer command line option, 331
 --generate-authors
 calibredb-catalog command line option, 312
 --generate-descriptions
 calibredb-catalog command line option, 312
 --generate-genres
 calibredb-catalog command line option, 312
 --generate-recently-added
 calibredb-catalog command line option, 312
 --generate-series
 calibredb-catalog command line option, 312
 --generate-titles
 calibredb-catalog command line option, 312
 --genre-source-field
 calibredb-catalog command line option, 312
 --get-cover
 ebook-meta command line option, 329
 --gui
 calibre-debug command line option, 298
 --gui-debug
 calibre-debug command line option, 298
 --header-note-source-field
 calibredb-catalog command line option, 312
 --help
 calibre command line option, 296
 calibre-customize command line option, 297
 calibre-debug command line option, 298
 calibre-server command line option, 301
 calibre-smtp command line option, 303
 command line option, 305
 ebook-convert command line option, 319
 ebook-edit command line option, 328
 ebook-meta command line option, 329
 ebook-polish command line option, 330
 ebook-viewer command line option, 331
 fetch-ebook-metadata command line option, 332
 lrf2lrs command line option, 333
 lrfviewer command line option, 334
 lrs2lrf command line option, 334
 web2disk command line option, 335
 --html-unwrap-factor
 ebook-convert command line option, 324
 --identifier
 calibredb-add command line option, 307
 ebook-meta command line option, 329
 fetch-ebook-metadata command line option, 332
 --ids
 calibredb-catalog command line option, 311
 --ignore
 calibredb-add command line option, 308
 --ignore_extensions
 calibredb-check_library command line option, 315
 --ignore_names
 calibredb-check_library command line option, 315
 --ignore-plugins
 calibre command line option, 296
 --ignored-fields
 calibre-server command line option, 301
 --implode-book
 calibre-debug command line option, 298
 --import-calibre-data
 calibre-debug command line option, 298
 --include-snippets
 calibredb-fts_search command line option, 318
 --index
 ebook-meta command line option, 329
 --indexing-speed
 calibredb-fts_index command line option, 317

--indexing-threshold
 calibredb-fts_search command line option, 318
 --input-encoding
 ebook-convert command line option, 320
 --input-profile
 ebook-convert command line option, 319
 --insert-blank-line
 ebook-convert command line option, 322
 --insert-blank-line-size
 ebook-convert command line option, 322
 --insert-metadata
 ebook-convert command line option, 325
 --inspect-mobi
 calibre-debug command line option, 298
 --is-multiple
 calibredb-add_custom_column command line option, 314
 --isbn
 calibredb-add command line option, 307
 ebook-convert command line option, 327
 ebook-meta command line option, 329
 fetch-ebook-metadata command line option, 332
 --item_count
 calibredb-list_categories command line option, 316
 --jacket
 ebook-polish command line option, 330
 --keep-ligatures
 ebook-convert command line option, 322
 --language
 ebook-convert command line option, 327
 ebook-meta command line option, 329
 --languages
 calibredb-add command line option, 307
 --level1-toc
 ebook-convert command line option, 326
 --level2-toc
 ebook-convert command line option, 326
 --level3-toc
 ebook-convert command line option, 326
 --library-path
 command line option, 305
 --limit
 calibredb-list command line option, 306
 calibredb-search command line option, 317
 --line-height
 ebook-convert command line option, 322
 --line-width
 calibredb-list command line option, 306
 --linearize-tables
 ebook-convert command line option, 322
 --list-fields
 calibredb-set_metadata command line option, 310
 --list-plugins
 calibre-customize command line option, 297
 --list-recipes
 ebook-convert command line option, 319
 --listen-on
 calibre-server command line option, 301
 --localhost
 calibre-smtp command line option, 303
 --log
 calibre-server command line option, 301
 --lrf-bookid
 ebook-meta command line option, 329
 --lrs
 lrs2lrf command line option, 334
 --manage-users
 calibre-server command line option, 301
 --margin-bottom
 ebook-convert command line option, 322
 --margin-left
 ebook-convert command line option, 322
 --margin-right
 ebook-convert command line option, 322
 --margin-top
 ebook-convert command line option, 323

--match-end-marker
calibredb-fts_search command line option, 318

--match-regex
web2disk command line option, 335

--match-start-marker
calibredb-fts_search command line option, 318

--max-files
web2disk command line option, 335

--max-header-line-size
calibre-server command line option, 301

--max-job-time
calibre-server command line option, 301

--max-jobs
calibre-server command line option, 301

--max-log-size
calibre-server command line option, 301

--max-opds-items
calibre-server command line option, 301

--max-opds-ungrouped-items
calibre-server command line option, 301

--max-recursions
web2disk command line option, 335

--max-request-body-size
calibre-server command line option, 301

--max-toc-links
ebook-convert command line option, 326

--merge-comments-rule
calibredb-catalog command line option, 312

--minimum-line-height
ebook-convert command line option, 323

--no-chapters-in-toc
ebook-convert command line option, 326

--no-default-epub-cover
ebook-convert command line option, 320

--no-svg-cover
ebook-convert command line option, 320

--no-update-check
calibre command line option, 296

--num-per-page
calibre-server command line option, 301

--one-book-per-directory
calibredb-add command line option, 308

--only-formats
calibredb-embed_metadata command line option, 316

--open-at
ebook-viewer command line option, 331

--opf
ebook-polish command line option, 330
fetch-ebook-metadata command line option, 332

--outbox
calibre-smtp command line option, 303

--output
lrf2lrs command line option, 333
lrs2lrf command line option, 334

--output-format
calibredb-fts_search command line option, 318

--output-profile
calibredb-catalog command line option, 312
ebook-convert command line option, 319

--page-breaks-before
ebook-convert command line option, 325

--password
calibre-smtp command line option, 304
command line option, 306

--paths
calibre-debug command line option, 298

--permanent
calibredb-remove command line option, 308

--pidfile
calibre-server command line option, 302

--port
calibre-server command line option, 302
calibre-smtp command line option, 304

--prefer-metadata-cover
ebook-convert command line option, 325

--prefix
calibredb-list command line option, 306

--prefix-rules

calibredb-catalog command line option, 312	--remove-soft-hyphens ebook-polish command line option, 330
--preserve-cover-aspect-ratio ebook-convert command line option, 321	--remove-unused-css ebook-polish command line option, 330
--preset calibredb-catalog command line option, 313	--replace-scene-breaks ebook-convert command line option, 324
--pretty-print ebook-convert command line option, 321	--replace-whitespace calibredb-export command line option, 310
--profile lrfviewer command line option, 334	--report calibredb-check_library command line option, 315
--progress calibredb-export command line option, 310	--restrict-to calibredb-fts_search command line option, 318
--pubdate ebook-convert command line option, 327	--run-plugin calibre-debug command line option, 298
--publisher ebook-convert command line option, 327	--run-test calibre-debug command line option, 298
ebook-meta command line option, 329	--run-without-debug calibre-debug command line option, 298
--raise-window ebook-viewer command line option, 332	--search calibredb-catalog command line option, 311
--rating ebook-convert command line option, 327	calibredb-list command line option, 306
ebook-meta command line option, 329	--search-replace ebook-convert command line option, 324
--read-metadata-from-opf ebook-convert command line option, 327	--search-the-net-urls calibre-server command line option, 302
--really-do-it calibredb-restore_database command line option, 315	--select-text ebook-edit command line option, 328
--recurse calibredb-add command line option, 308	--separator calibredb-list command line option, 306
--relay calibre-smtp command line option, 304	--series calibredb-add command line option, 307
--remove-first-image ebook-convert command line option, 325	ebook-convert command line option, 327
--remove-jacket ebook-polish command line option, 330	ebook-meta command line option, 329
--remove-paragraph-spacing ebook-convert command line option, 323	--series-index calibredb-add command line option, 308
--remove-paragraph-spacing-indent-size ebook-convert command line option, 323	ebook-convert command line option, 327
--remove-plugin calibre-customize command line option, 297	--shutdown-running-calibre calibre command line option, 296

calibre-debug command line option, 298	--tags calibreddb-add command line option, 308
--shutdown-timeout calibre-server command line option, 302	ebook-convert command line option, 327
--single-dir calibreddb-export command line option, 310	ebook-meta command line option, 329
--smarten-punctuation ebook-convert command line option, 323	--template calibreddb-export command line option, 311
ebook-polish command line option, 331	calibreddb-list command line option, 307
--sort-by calibreddb-list command line option, 307	--template_file calibreddb-list command line option, 307
--sr1-replace ebook-convert command line option, 324	--template_heading calibreddb-list command line option, 307
--sr1-search ebook-convert command line option, 324	--test-build calibre-debug command line option, 299
--sr2-replace ebook-convert command line option, 325	--thumb-width calibreddb-catalog command line option, 313
--sr2-search ebook-convert command line option, 325	--timefmt calibreddb-export command line option, 311
--sr3-replace ebook-convert command line option, 325	--timeout calibre-server command line option, 302
--sr3-search ebook-convert command line option, 325	calibre-smtp command line option, 303
--ssl-certfile calibre-server command line option, 302	command line option, 306
--ssl-keyfile calibre-server command line option, 302	fetch-ebook-metadata command line option, 332
--start-in-tray calibre command line option, 296	web2disk command line option, 335
--start-reading-at ebook-convert command line option, 325	--timestamp ebook-convert command line option, 327
--subject calibre-smtp command line option, 303	--title calibreddb-add command line option, 308
--subset-embedded-fonts ebook-convert command line option, 323	ebook-convert command line option, 327
--subset-font calibre-debug command line option, 299	ebook-meta command line option, 329
--subset-fonts ebook-polish command line option, 331	fetch-ebook-metadata command line option, 332
	--title-sort ebook-convert command line option, 327
	ebook-meta command line option, 329
	--to-dir calibreddb-export command line option, 311
	--to-lowercase

calibredb-export command line option, 311

--to-opf
ebook-meta command line option, 329

--toc-filter
ebook-convert command line option, 326

--toc-threshold
ebook-convert command line option, 326

--toc-title
ebook-convert command line option, 321

--transform-css-rules
ebook-convert command line option, 323

--transform-html-rules
ebook-convert command line option, 323

--trusted-ips
calibre-server command line option, 302

--unsmarten-punctuation
ebook-convert command line option, 323

--upgrade-book
ebook-polish command line option, 331

--url-prefix
calibre-server command line option, 302

--use-auto-toc
ebook-convert command line option, 326

--use-existing-cover
calibredb-catalog command line option, 313

--userdb
calibre-server command line option, 302

--username
calibre-smtp command line option, 304
command line option, 306

--vacuum-fts-db
calibredb-check_library command line option, 315

--verbose
calibre command line option, 296
calibre-smtp command line option, 303
calibredb-catalog command line option, 311
ebook-convert command line option, 327
ebook-polish command line option, 331

fetch-ebook-metadata command line option, 333

lrf2lrs command line option, 333

lrfviewer command line option, 334

lrs2lrf command line option, 334

web2disk command line option, 335

--version
calibre command line option, 296
calibre-customize command line option, 297
calibre-debug command line option, 299
calibre-server command line option, 302
calibre-smtp command line option, 303
command line option, 306
ebook-convert command line option, 319
ebook-edit command line option, 328
ebook-meta command line option, 330
ebook-polish command line option, 331
ebook-viewer command line option, 332
fetch-ebook-metadata command line option, 333
lrf2lrs command line option, 333
lrfviewer command line option, 334
lrs2lrf command line option, 334
web2disk command line option, 335

--viewer
calibre-debug command line option, 299

--visual-debug
lrfviewer command line option, 334

--wait-for-completion
calibredb-fts_index command line option, 317

--white-background
lrfviewer command line option, 334

--width
calibredb-list_categories command line option, 316

--with-library
calibre command line option, 296
command line option, 305

--worker-count
calibre-server command line option, 302

-a
calibre-customize command line option, 296
calibre-smtp command line option, 303
calibredb-add command line option, 307

- calibredb-set_custom command line option, 314
- ebook-meta command line option, 328
- fetch-ebook-metadata command line option, 332
- b
 - calibre-customize command line option, 296
- c
 - calibre-debug command line option, 297
 - calibredb-add command line option, 307
 - calibredb-check_library command line option, 315
 - calibredb-list_categories command line option, 315
 - ebook-meta command line option, 329
 - ebook-polish command line option, 330
 - fetch-ebook-metadata command line option, 332
- d
 - calibre-debug command line option, 297
 - calibredb-add command line option, 307
 - calibredb-custom_columns command line option, 314
 - ebook-convert command line option, 327
 - ebook-meta command line option, 329
 - fetch-ebook-metadata command line option, 332
 - web2disk command line option, 335
- e
 - calibre-debug command line option, 298
 - calibre-smtp command line option, 304
 - calibredb-add command line option, 307
 - calibredb-check_library command line option, 315
 - ebook-polish command line option, 330
- f
 - calibre-debug command line option, 299
 - calibre-smtp command line option, 303
 - calibredb-embed_metadata command line option, 316
 - calibredb-list command line option, 306
 - calibredb-remove_custom_column command line option, 314
- calibredb-set_metadata command line option, 309
- ebook-polish command line option, 331
- ebook-viewer command line option, 331
- g
 - calibre-debug command line option, 298
- h
 - calibre command line option, 296
 - calibre-customize command line option, 297
 - calibre-debug command line option, 298
 - calibre-server command line option, 301
 - calibre-smtp command line option, 303
 - command line option, 305
 - ebook-convert command line option, 319
 - ebook-edit command line option, 328
 - ebook-meta command line option, 329
 - ebook-polish command line option, 330
 - ebook-viewer command line option, 331
 - fetch-ebook-metadata command line option, 332
 - lrf2lrs command line option, 333
 - lrfviewer command line option, 334
 - lrs2lrf command line option, 334
 - web2disk command line option, 335
- i
 - calibre-debug command line option, 298
 - calibredb-add command line option, 307
 - calibredb-catalog command line option, 311
 - calibredb-list_categories command line option, 316
 - ebook-meta command line option, 329
 - ebook-polish command line option, 330
 - fetch-ebook-metadata command line option, 332
- j
 - ebook-polish command line option, 330
- k
 - ebook-meta command line option, 328
- l
 - calibre-customize command line option, 297
 - calibre-smtp command line option, 303
 - calibredb-add command line option, 307
 - calibredb-search command line option, 317

- calibredb-set_metadata command line option, 310
 - ebook-meta command line option, 329
 - m
 - calibre-debug command line option, 298
 - calibredb-add command line option, 307
 - ebook-convert command line option, 327
 - n
 - calibredb-check_library command line option, 315
 - web2disk command line option, 335
 - o
 - calibre-smtp command line option, 303
 - ebook-polish command line option, 330
 - fetch-ebook-metadata command line option, 332
 - lrf2lrs command line option, 333
 - lrs2lrf command line option, 334
 - p
 - calibre-smtp command line option, 304
 - ebook-meta command line option, 329
 - ebook-polish command line option, 331
 - fetch-ebook-metadata command line option, 332
 - r
 - calibre-customize command line option, 297
 - calibre-debug command line option, 298
 - calibre-smtp command line option, 304
 - calibredb-add command line option, 308
 - calibredb-check_library command line option, 315
 - calibredb-list_categories command line option, 315
 - calibredb-restore_database command line option, 315
 - ebook-meta command line option, 329
 - web2disk command line option, 335
 - s
 - calibre command line option, 296
 - calibre-debug command line option, 298
 - calibre-smtp command line option, 303
 - calibredb-add command line option, 307
 - calibredb-catalog command line option, 311
 - calibredb-list command line option, 306
 - ebook-meta command line option, 329
 - t
 - calibre-debug command line option, 298
 - calibre-smtp command line option, 303
 - calibredb-add command line option, 308
 - calibredb-list command line option, 307
 - ebook-meta command line option, 329
 - fetch-ebook-metadata command line option, 332
 - web2disk command line option, 335
 - u
 - calibre-smtp command line option, 304
 - ebook-polish command line option, 330
 - v
 - calibre command line option, 296
 - calibre-smtp command line option, 303
 - calibredb-catalog command line option, 311
 - ebook-convert command line option, 327
 - fetch-ebook-metadata command line option, 333
 - w
 - calibre-debug command line option, 299
 - calibredb-list command line option, 306
 - calibredb-list_categories command line option, 316
 - x
 - calibre-debug command line option, 298
- A**
- abort_article() (*calibre.web.feeds.news.BasicNewsRecipe* metodo), 37
 - abort_recipe_processing() (*calibre.web.feeds.news.BasicNewsRecipe* metodo), 37
 - abspath_to_name() (*calibre.ebooks.oeb.polish.container.Container* metodo), 354
 - accept_drag_move_event() (*calibre.gui2.actions.InterfaceAction* metodo), 275
 - accept_enter_event() (*calibre.gui2.actions.InterfaceAction* metodo), 275
 - accepts_drops (*calibre.gui2.actions.InterfaceAction* attributo), 275

<code>action_add_menu</code>	(<i>calibre.gui2.actions.InterfaceAction</i> attributo), 275	<code>all_field_ids()</code>	(<i>calibre.db.cache.Cache</i> metodo), 347
<code>action_menu_clone_qaction</code>	(<i>calibre.gui2.actions.InterfaceAction</i> attributo), 275	<code>all_field_keys()</code>	(<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 201
<code>action_shortcut_name</code>	(<i>calibre.gui2.actions.InterfaceAction</i> attributo), 275	<code>all_field_names()</code>	(<i>calibre.db.cache.Cache</i> metodo), 347
<code>action_spec</code>	(<i>calibre.gui2.actions.InterfaceAction</i> attributo), 275	<code>all_non_none_fields()</code>	(<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 201
<code>action_type</code>	(<i>calibre.gui2.actions.InterfaceAction</i> attributo), 275	<code>allowed_in_menu</code>	(<i>calibre.gui2.tweak_book.plugin.Tool</i> attributo), 362
<code>add_annotation_to_library()</code>	(<i>calibre.devices.usbms.device.Device</i> metodo), 272	<code>allowed_in_toolbar</code>	(<i>calibre.gui2.tweak_book.plugin.Tool</i> attributo), 362
<code>add_book()</code>	(<i>calibre.devices.interface.BookList</i> metodo), 268	API, 369	
<code>add_books()</code>	(<i>calibre.db.cache.Cache</i> metodo), 346	<code>apply_container_update_to_gui()</code>	(<i>calibre.gui2.tweak_book.boss.Boss</i> metodo), 363
<code>add_books_to_metadata()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo della classe), 265	<code>apply_unique_properties()</code>	(<i>calibre.ebooks.oeb.polish.container.Container</i> metodo), 354
<code>add_books_to_metadata()</code>	(<i>calibre.devices.usbms.driver.USBMS</i> metodo), 273	<code>articles_are_obfuscated</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 41
<code>add_custom_book_data()</code>	(<i>calibre.db.cache.Cache</i> metodo), 346	<code>ASK_TO_ALLOW_CONNECT</code>	(<i>calibre.devices.interface.DevicePlugin</i> attributo), 262
<code>add_extra_files()</code>	(<i>calibre.db.cache.Cache</i> metodo), 346	<code>author</code>	(<i>calibre.customize.InterfaceActionBase</i> attributo), 277
<code>add_file()</code>	(<i>calibre.ebooks.oeb.polish.container.Container</i> metodo), 354	<code>author</code>	(<i>calibre.customize.MetadataReaderPlugin</i> attributo), 252
<code>add_format()</code>	(<i>calibre.db.cache.Cache</i> metodo), 346	<code>author</code>	(<i>calibre.customize.MetadataWriterPlugin</i> attributo), 253
<code>add_listener()</code>	(<i>calibre.db.cache.Cache</i> metodo), 346	<code>author</code>	(<i>calibre.customize.Plugin</i> attributo), 249
<code>add_name_to_manifest()</code>	(<i>calibre.ebooks.oeb.polish.container.Container</i> metodo), 354	<code>author</code>	(<i>calibre.customize.PreferencesPlugin</i> attributo), 278
<code>add_or_replace_jacket()</code>	(nel modulo <i>calibre.ebooks.oeb.polish.jacket</i>), 359	<code>author</code>	(<i>calibre.devices.usbms.driver.USBMS</i> attributo), 272
<code>add_properties()</code>	(<i>calibre.ebooks.oeb.polish.container.Container</i> metodo), 354	<code>author</code>	(<i>calibre.ebooks.metadata.sources.base.Source</i> attributo), 254
<code>add_savepoint()</code>	(<i>calibre.gui2.tweak_book.boss.Boss</i> metodo), 363	<code>author_data()</code>	(<i>calibre.db.cache.Cache</i> metodo), 347
<code>add_toc_thumbnail()</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> metodo), 37	<code>author_sort_from_authors()</code>	(<i>calibre.db.cache.Cache</i> metodo), 347
<code>adeify_images()</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> metodo della classe), 37	<code>auto_cleanup</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 42
<code>all_book_ids()</code>	(<i>calibre.db.cache.Cache</i> metodo), 346	<code>auto_cleanup_keep</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 42
<code>all_field_for()</code>	(<i>calibre.db.cache.Cache</i> metodo), 347	<code>auto_repeat</code>	(<i>calibre.gui2.actions.InterfaceAction</i> attributo), 275

auto_trim_covers (calibre.ebooks.metadata.sources.base.Source attributo), 255	BuiltinAuthorSorts (classe in calibre.utils.formatter_functions), 185
B	BuiltinBookCount (classe in calibre.utils.formatter_functions), 199
BACKLOADING_ERROR_MESSAGE (calibre.devices.usbms.device.Device attributo), 270	BuiltinBooksize (classe in calibre.utils.formatter_functions), 186
BasicNewsRecipe (classe in calibre.web.feeds.news), 37	BuiltinBookValues (classe in calibre.utils.formatter_functions), 200
BCD (calibre.devices.interface.DevicePlugin attributo), 261	BuiltinCapitalize (classe in calibre.utils.formatter_functions), 197
BCD (calibre.devices.usbms.device.Device attributo), 269	BuiltinCeiling (classe in calibre.utils.formatter_functions), 182
book_class (calibre.devices.usbms.driver.USBMS attributo), 272	BuiltinCharacter (classe in calibre.utils.formatter_functions), 197
book_created (calibre.db.cache.Cache.EventType attributo), 345	BuiltinCmp (classe in calibre.utils.formatter_functions), 196
book_edited (calibre.db.cache.Cache.EventType attributo), 345	BuiltinConnectedDeviceName (classe in calibre.utils.formatter_functions), 186
book_type (calibre.ebooks.oeb.polish.container.Container attributo), 355	BuiltinConnectedDeviceUUID (classe in calibre.utils.formatter_functions), 186
BookList (classe in calibre.devices.interface), 268	BuiltinContains (classe in calibre.utils.formatter_functions), 189
booklist_class (calibre.devices.usbms.driver.USBMS attributo), 272	BuiltinCount (classe in calibre.utils.formatter_functions), 191
books() (calibre.devices.interface.DevicePlugin metodo), 264	BuiltinCurrentLibraryName (classe in calibre.utils.formatter_functions), 186
books() (calibre.devices.usbms.driver.USBMS metodo), 272	BuiltinCurrentLibraryPath (classe in calibre.utils.formatter_functions), 186
books_for_field() (calibre.db.cache.Cache metodo), 347	BuiltinCurrentVirtualLibraryName (classe in calibre.utils.formatter_functions), 186
books_in_virtual_library() (calibre.db.cache.Cache metodo), 347	BuiltinDaysBetween (classe in calibre.utils.formatter_functions), 183
books_removed (calibre.db.cache.Cache.EventType attributo), 345	BuiltinDivide (classe in calibre.utils.formatter_functions), 182
boss (calibre.gui2.tweak_book.plugin.Tool property), 362	BuiltinEval (classe in calibre.utils.formatter_functions), 195
Boss (classe in calibre.gui2.tweak_book.boss), 363	BuiltinExtraFileModtime (classe in calibre.utils.formatter_functions), 200
BuiltinAdd (classe in calibre.utils.formatter_functions), 181	BuiltinExtraFileNames (classe in calibre.utils.formatter_functions), 200
BuiltinAnd (classe in calibre.utils.formatter_functions), 183	BuiltinExtraFileSize (classe in calibre.utils.formatter_functions), 200
BuiltinAnnotationCount (classe in calibre.utils.formatter_functions), 185	
BuiltinApproximateFormats (classe in calibre.utils.formatter_functions), 185	
BuiltinArguments (classe in calibre.utils.formatter_functions), 195	
BuiltinAssign (classe in calibre.utils.formatter_functions), 195	
BuiltinAuthorLinks (classe in calibre.utils.formatter_functions), 185	

BuiltinField	(classe in <i>bre.utils.formatter_functions</i>), 187	cali-	BuiltinListDifference	(classe in <i>calibre.utils.formatter_functions</i>), 192
BuiltinFieldExists	(classe in <i>libre.utils.formatter_functions</i>), 189	ca-	BuiltinListEquals	(classe in <i>libre.utils.formatter_functions</i>), 192
BuiltinFinishFormatting	(classe in <i>calibre.utils.formatter_functions</i>), 184		BuiltinListIntersection	(classe in <i>calibre.utils.formatter_functions</i>), 192
BuiltinFirstMatchingCmp	(classe in <i>calibre.utils.formatter_functions</i>), 196		BuiltinListItem	(classe in <i>bre.utils.formatter_functions</i>), 191
BuiltinFirstNonEmpty	(classe in <i>calibre.utils.formatter_functions</i>), 190		BuiltinListJoin	(classe in <i>bre.utils.formatter_functions</i>), 192
BuiltinFloor	(classe in <i>bre.utils.formatter_functions</i>), 182	cali-	BuiltinListRe	(classe in <i>bre.utils.formatter_functions</i>), 193
BuiltinFormatDateField	(classe in <i>calibre.utils.formatter_functions</i>), 184		BuiltinListReGroup	(classe in <i>libre.utils.formatter_functions</i>), 193
BuiltinFormatNumber	(classe in <i>calibre.utils.formatter_functions</i>), 184		BuiltinListRemoveDuplicates	(classe in <i>bre.utils.formatter_functions</i>), 193
BuiltinFormatsModtimes	(classe in <i>calibre.utils.formatter_functions</i>), 187		BuiltinListSort	(classe in <i>bre.utils.formatter_functions</i>), 193
BuiltinFormatsPaths	(classe in <i>calibre.utils.formatter_functions</i>), 187		BuiltinListSplit	(classe in <i>libre.utils.formatter_functions</i>), 193
BuiltinFormatsSizes	(classe in <i>calibre.utils.formatter_functions</i>), 187		BuiltinListUnion	(classe in <i>libre.utils.formatter_functions</i>), 194
BuiltinFractionalPart	(classe in <i>calibre.utils.formatter_functions</i>), 182	cali-	BuiltinLookup	(classe in <i>bre.utils.formatter_functions</i>), 190
BuiltinGlobals	(classe in <i>bre.utils.formatter_functions</i>), 195		BuiltinLowercase	(classe in <i>libre.utils.formatter_functions</i>), 197
BuiltinHasCover	(classe in <i>bre.utils.formatter_functions</i>), 187	cali-	BuiltinMod	(classe in <i>calibre.utils.formatter_functions</i>), 182
BuiltinHasExtraFiles	(classe in <i>calibre.utils.formatter_functions</i>), 200		BuiltinMultiply	(classe in <i>bre.utils.formatter_functions</i>), 182
BuiltinHumanReadable	(classe in <i>calibre.utils.formatter_functions</i>), 184		BuiltinNot	(classe in <i>calibre.utils.formatter_functions</i>), 183
BuiltinIdentifierInList	(classe in <i>calibre.utils.formatter_functions</i>), 190	cali-	BuiltinOnDevice	(classe in <i>bre.utils.formatter_functions</i>), 188
BuiltinIfempty	(classe in <i>bre.utils.formatter_functions</i>), 189		BuiltinOr	(classe in <i>calibre.utils.formatter_functions</i>), 183
BuiltinInList	(classe in <i>bre.utils.formatter_functions</i>), 191	cali-	BuiltinPrint	(classe in <i>bre.utils.formatter_functions</i>), 195
BuiltinIsMarked	(classe in <i>bre.utils.formatter_functions</i>), 187	cali-	BuiltinRange	(classe in <i>bre.utils.formatter_functions</i>), 194
BuiltinLanguageCodes	(classe in <i>calibre.utils.formatter_functions</i>), 188			
BuiltinLanguageStrings	(classe in <i>calibre.utils.formatter_functions</i>), 188			
BuiltinListCountMatching	(classe in <i>calibre.utils.formatter_functions</i>), 192			

BuiltinRatingToStars	(classe in calibre.utils.formatter_functions), 184	in	199
BuiltinRawField	(classe in calibre.utils.formatter_functions), 188	cali-	BuiltinSubtract (classe in calibre.utils.formatter_functions), 183
BuiltinRawList	(classe in calibre.utils.formatter_functions), 188	cali-	BuiltinSwapAroundArticles (classe in calibre.utils.formatter_functions), 199
BuiltinRe	(classe in calibre.utils.formatter_functions), 197		BuiltinSwapAroundComma (classe in calibre.utils.formatter_functions), 199
BuiltinReGroup	(classe in calibre.utils.formatter_functions), 198	cali-	BuiltinSwitch (classe in calibre.utils.formatter_functions), 190
BuiltinRound	(classe in calibre.utils.formatter_functions), 182	cali-	BuiltinSwitchIf (classe in calibre.utils.formatter_functions), 190
BuiltinSelect	(classe in calibre.utils.formatter_functions), 191	cali-	BuiltinTemplate (classe in calibre.utils.formatter_functions), 196
BuiltinSeriesSort	(classe in calibre.utils.formatter_functions), 188	ca-	BuiltinTest (classe in calibre.utils.formatter_functions), 189
BuiltinSetGlobals	(classe in calibre.utils.formatter_functions), 200	ca-	BuiltinTitlecase (classe in calibre.utils.formatter_functions), 197
BuiltinShorten	(classe in calibre.utils.formatter_functions), 198	cali-	BuiltinToday (classe in calibre.utils.formatter_functions), 183
BuiltinStrcat	(classe in calibre.utils.formatter_functions), 198	cali-	BuiltinToHex (classe in calibre.utils.formatter_functions), 199
BuiltinStrcatMax	(classe in calibre.utils.formatter_functions), 198	ca-	BuiltinTransliterate (classe in calibre.utils.formatter_functions), 199
BuiltinStrcmp	(classe in calibre.utils.formatter_functions), 196	cali-	BuiltinUppercase (classe in calibre.utils.formatter_functions), 197
BuiltinStrcmpcase	(classe in calibre.utils.formatter_functions), 196	ca-	BuiltinUrlsFromIdentifiers (classe in calibre.utils.formatter_functions), 185
BuiltinStrInList	(classe in calibre.utils.formatter_functions), 191	ca-	BuiltinUserCategories (classe in calibre.utils.formatter_functions), 189
BuiltinStrlen	(classe in calibre.utils.formatter_functions), 198	cali-	BuiltinVirtualLibraries (classe in calibre.utils.formatter_functions), 189
BuiltinSubitems	(classe in calibre.utils.formatter_functions), 194	cali-	
BuiltinSublist	(classe in calibre.utils.formatter_functions), 194	cali-	
BuiltinSubstr	(classe in calibre.utils.formatter_functions),	cali-	

C

Cache	(classe in calibre.db.cache), 345
Cache.EventType	(classe in calibre.db.cache), 345
cached_cover_url_is_reliable	(calibre.ebooks.metadata.sources.base.Source attributo), 254
calibre command line option	
--detach	296
--help	296
--ignore-plugins	296
--no-update-check	296
--shutdown-running-calibre	296
--start-in-tray	296

- verbose, 296
- version, 296
- with-library, 296
- h, 296
- s, 296
- v, 296
- calibre.customize
 - module, 248
- calibre.customize.conversion
 - module, 258
- calibre.db.cache
 - module, 345
- calibre.devices.interface
 - module, 260
- calibre.ebooks.metadata.book.base
 - module, 201
- calibre.ebooks.metadata.sources.base
 - module, 254
- calibre.ebooks.oeb.polish.container
 - module, 354
- calibre.ebooks.oeb.polish.cover
 - module, 360
- calibre.ebooks.oeb.polish.css
 - module, 360
- calibre.ebooks.oeb.polish.jacket
 - module, 359
- calibre.ebooks.oeb.polish.pretty
 - module, 359
- calibre.ebooks.oeb.polish.replace
 - module, 358
- calibre.ebooks.oeb.polish.split
 - module, 359
- calibre.ebooks.oeb.polish.toc
 - module, 361
- calibre.gui2.tweak_book.boss
 - module, 363
- calibre.utils.formatter_functions
 - module, 181
- calibre.web.feeds.news
 - module, 37
- calibre-customize command line option
 - add-plugin, 296
 - build-plugin, 296
 - customize-plugin, 296
 - disable-plugin, 296
 - enable-plugin, 296
 - help, 297
 - list-plugins, 297
 - remove-plugin, 297
 - version, 297
 - a, 296
 - b, 296
 - h, 297
 - l, 297
 - r, 297
- calibre-debug command line option
 - add-simple-plugin, 297
 - command, 297
 - debug-device-driver, 297
 - default-programs, 297
 - diff, 297
 - edit-book, 297
 - exec-file, 298
 - explode-book, 298
 - export-all-calibre-data, 298
 - fix-multiprocessing, 298
 - gui, 298
 - gui-debug, 298
 - help, 298
 - implode-book, 298
 - import-calibre-data, 298
 - inspect-mobi, 298
 - paths, 298
 - run-plugin, 298
 - run-test, 298
 - run-without-debug, 298
 - shutdown-running-calibre, 298
 - subset-font, 299
 - test-build, 299
 - version, 299
 - viewer, 299
 - c, 297
 - d, 297
 - e, 298
 - f, 299
 - g, 298
 - h, 298
 - i, 298
 - m, 298
 - r, 298
 - s, 298
 - t, 298
 - w, 299
 - x, 298
- calibre-server command line option
 - access-log, 299
 - ajax-timeout, 299
 - auth-mode, 299
 - auto-reload, 299
 - ban-after, 299
 - ban-for, 299
 - book-list-mode, 299
 - compress-min-size, 300
 - custom-list-template, 300
 - daemonize, 300
 - disable-allow-socket-
 - preallocation, 300

```

--disable-auth, 300
--disable-fallback-to-
    detected-interface,
    300
--disable-local-write, 300
--disable-log-not-found, 300
--disable-use-bonjour, 300
--disable-use-sendfile, 300
--displayed-fields, 300
--enable-allow-socket-
    preallocation,
    300
--enable-auth, 300
--enable-fallback-to-
    detected-interface,
    300
--enable-local-write, 300
--enable-log-not-found, 300
--enable-use-bonjour, 300
--enable-use-sendfile, 300
--help, 301
--ignored-fields, 301
--listen-on, 301
--log, 301
--manage-users, 301
--max-header-line-size, 301
--max-job-time, 301
--max-jobs, 301
--max-log-size, 301
--max-opds-items, 301
--max-opds-ungrouped-items, 301
--max-request-body-size, 301
--num-per-page, 301
--pidfile, 302
--port, 302
--search-the-net-urls, 302
--shutdown-timeout, 302
--ssl-certfile, 302
--ssl-keyfile, 302
--timeout, 302
--trusted-ips, 302
--url-prefix, 302
--userdb, 302
--version, 302
--worker-count, 302
-h, 301
calibre-smtp command line option
--attachment, 303
--cafile, 304
--dont-verify-server-certificate,
    304
--encryption-method, 304
--fork, 303
--help, 303
--localhost, 303
--outbox, 303
--password, 304
--port, 304
--relay, 304
--subject, 303
--timeout, 303
--username, 304
--verbose, 303
--version, 303
-a, 303
-e, 304
-f, 303
-h, 303
-l, 303
-o, 303
-p, 304
-r, 304
-s, 303
-t, 303
-u, 304
-v, 303
calibredb-add command line option
-l, 308
-I, 307
-S, 308
-T, 308
--add, 308
--authors, 307
--automerge, 307
--cover, 307
--duplicates, 307
--empty, 307
--identifier, 307
--ignore, 308
--isbn, 307
--languages, 307
--one-book-per-directory, 308
--recurse, 308
--series, 307
--series-index, 308
--tags, 308
--title, 308
-a, 307
-c, 307
-d, 307
-e, 307
-i, 307
-l, 307
-m, 307
-r, 308
-s, 307
-t, 308

```

calibredb-add_custom_column command
 line option
 --display, 313
 --is-multiple, 314

calibredb-add_format command line option
 --as-extra-data-file, 309
 --dont-replace, 309

calibredb-backup_metadata command line option
 --all, 316

calibredb-catalog command line option
 --catalog-title, 311
 --cross-reference-authors, 311
 --debug-pipeline, 311
 --exclude-genre, 312
 --exclusion-rules, 312
 --generate-authors, 312
 --generate-descriptions, 312
 --generate-genres, 312
 --generate-recently-added, 312
 --generate-series, 312
 --generate-titles, 312
 --genre-source-field, 312
 --header-note-source-field, 312
 --ids, 311
 --merge-comments-rule, 312
 --output-profile, 312
 --prefix-rules, 312
 --preset, 313
 --search, 311
 --thumb-width, 313
 --use-existing-cover, 313
 --verbose, 311
 -i, 311
 -s, 311
 -v, 311

calibredb-check_library command line option
 --csv, 315
 --ignore_extensions, 315
 --ignore_names, 315
 --report, 315
 --vacuum-fts-db, 315
 -c, 315
 -e, 315
 -n, 315
 -r, 315

calibredb-custom_columns command line option
 --details, 314
 -d, 314

calibredb-embed_metadata command line option
 --only-formats, 316
 -f, 316

calibredb-export command line option
 --all, 310
 --dont-asciiize, 310
 --dont-save-cover, 310
 --dont-save-extra-files, 310
 --dont-update-metadata, 310
 --dont-write-opf, 310
 --formats, 310
 --progress, 310
 --replace-whitespace, 310
 --single-dir, 310
 --template, 311
 --timefmt, 311
 --to-dir, 311
 --to-lowercase, 311

calibredb-fts_index command line option
 --indexing-speed, 317
 --wait-for-completion, 317

calibredb-fts_search command line option
 --do-not-match-on-related-words, 318
 --include-snippets, 318
 --indexing-threshold, 318
 --match-end-marker, 318
 --match-start-marker, 318
 --output-format, 318
 --restrict-to, 318

calibredb-list command line option
 --ascending, 306
 --fields, 306
 --for-machine, 306
 --limit, 306
 --line-width, 306
 --prefix, 306
 --search, 306
 --separator, 306
 --sort-by, 307
 --template, 307
 --template_file, 307
 --template_heading, 307
 -f, 306
 -s, 306
 -t, 307
 -w, 306

calibredb-list_categories command line option
 --categories, 315
 --csv, 315
 --dialect, 316
 --item_count, 316
 --width, 316
 -c, 315

-i, 316
 -r, 315
 -w, 316
 calibredb-remove command line option
 --permanent, 308
 calibredb-remove_custom_column command
 line option
 --force, 314
 -f, 314
 calibredb-restore_database command li-
 ne option
 --really-do-it, 315
 -r, 315
 calibredb-search command line option
 --limit, 317
 -l, 317
 calibredb-set_custom command line op-
 tion
 --append, 314
 -a, 314
 calibredb-set_metadata command line
 option
 --field, 309
 --list-fields, 310
 -f, 309
 -l, 310
 calibredb-show_metadata command line
 option
 --as-opf, 309
 can_be_disabled (calibre.customize.conversion.InputFormatPlugin
 attributo), 258
 can_be_disabled (calibre.customize.conversion.OutputFormatPlugin
 attributo), 259
 can_be_disabled (calibre.customize.InterfaceActionBase attributo),
 277
 can_be_disabled (calibre.customize.Plugin attributo),
 249
 can_be_disabled (calibre.customize.PreferencesPlugin attributo),
 278
 CAN_DO_DEVICE_DB_PLUGBOARD (cali-
 bre.devices.interface.DevicePlugin attributo),
 261
 can_get_multiple_covers (cali-
 bre.ebooks.metadata.sources.base.Source
 attributo), 254
 can_handle() (calibre.devices.interface.DevicePlugin
 metodo), 263
 can_handle_windows() (cali-
 bre.devices.interface.DevicePlugin metodo),
 263
 can_handle_windows() (cali-
 bre.devices.usbms.device.Device metodo),
 271
 CAN_SET_METADATA (cali-
 bre.devices.interface.DevicePlugin attributo),
 261
 CAN_SET_METADATA (cali-
 bre.devices.usbms.driver.USBMS attributo),
 272
 canonicalize_internal_url() (cali-
 bre.web.feeds.news.BasicNewsRecipe metodo),
 38
 capabilities (cali-
 bre.ebooks.metadata.sources.base.Source
 attributo), 254
 card_prefix() (calibre.devices.interface.DevicePlugin
 metodo), 264
 card_prefix() (calibre.devices.usbms.device.Device
 metodo), 270
 CatalogPlugin (classe in calibre.customize), 253
 category (calibre.customize.PreferencesPlugin attributo),
 278
 category_order (calibre.customize.PreferencesPlugin
 attributo), 278
 center_navbar (cali-
 bre.web.feeds.news.BasicNewsRecipe attributo),
 42
 change_font() (nel modulo ca-
 libre.ebooks.oeb.polish.fonts),
 360
 changed_signal (cali-
 bre.gui2.preferences.ConfigWidgetBase
 attributo), 279
 changed_signal (cali-
 bre.gui2.preferences.ConfigWidgetInterface
 attributo), 278
 clean_downloaded_metadata() (cali-
 bre.ebooks.metadata.sources.base.Source
 metodo), 255
 cleanup() (calibre.web.feeds.news.BasicNewsRecipe
 metodo), 38
 CLI (classe in calibre.devices.usbms.cli), 272
 cli_main() (calibre.customize.Plugin metodo), 250
 cli_options (calibre.customize.CatalogPlugin attribu-
 to), 253
 clone_browser() (cali-
 bre.web.feeds.news.BasicNewsRecipe metodo),
 38
 close_editor() (calibre.gui2.tweak_book.boss.Boss
 metodo), 363
 command line option
 --help, 305
 --library-path, 305
 --password, 306

- timeout, 306
 - username, 306
 - version, 306
 - with-library, 305
 - h, 305
 - commit() (*calibre.ebooks.oeb.polish.container.Container* metodo), 355
 - commit() (*calibre.gui2.preferences.ConfigWidgetBase* metodo), 280
 - commit() (*calibre.gui2.preferences.ConfigWidgetInterface* metodo), 279
 - commit_all_editors_to_container() (*calibre.gui2.tweak_book.boss.Boss* metodo), 363
 - commit_item() (*calibre.ebooks.oeb.polish.container.Container* metodo), 355
 - common_options (*calibre.customize.conversion.InputFormatPlugin* attributo), 258
 - common_options (*calibre.customize.conversion.OutputFormatPlugin* attributo), 259
 - compress_covers() (*calibre.db.cache.Cache* metodo), 347
 - compress_news_images (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 42
 - compress_news_images_auto_size (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 42
 - compress_news_images_max_size (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 42
 - config_help_message (*calibre.ebooks.metadata.sources.base.Source* attributo), 254
 - config_widget (*calibre.customize.PreferencesPlugin* attributo), 278
 - config_widget() (*calibre.customize.Plugin* metodo), 249
 - config_widget() (*calibre.devices.interface.DevicePlugin* metodo della classe), 265
 - config_widget() (*calibre.ebooks.metadata.sources.base.Source* metodo), 255
 - ConfigWidgetBase (classe in *calibre.gui2.preferences*), 279
 - ConfigWidgetInterface (classe in *calibre.gui2.preferences*), 278
 - Container (classe in *calibre.ebooks.oeb.polish.container*), 354
 - contains(), 153
 - conversion_options (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 42
 - convert() (*calibre.customize.conversion.InputFormatPlugin* metodo), 258
 - convert() (*calibre.customize.conversion.OutputFormatPlugin* metodo), 260
 - copy_cover_to() (*calibre.db.cache.Cache* metodo), 347
 - copy_format_to() (*calibre.db.cache.Cache* metodo), 347
 - core_usage (*calibre.customize.conversion.InputFormatPlugin* attributo), 258
 - cover() (*calibre.db.cache.Cache* metodo), 347
 - cover_margins (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 42
 - create_action() (*calibre.gui2.tweak_book.plugin.Tool* metodo), 362
 - create_inline_toc() (*calibre.ebooks.oeb.polish.toc* modulo), 361
 - create_menu_action() (*calibre.gui2.actions.InterfaceAction* metodo), 275
 - create_widget() (*calibre.customize.PreferencesPlugin* metodo), 278
 - CSS, 369
 - current_container (*calibre.gui2.tweak_book.plugin.Tool* property), 362
 - currently_editing (*calibre.gui2.tweak_book.boss.Boss* property), 363
 - custom_field_keys() (*calibre.ebooks.metadata.book.base.Metadata* metodo), 201
 - customization_help() (*calibre.customize.Plugin* metodo), 250
 - customization_help() (*calibre.ebooks.metadata.sources.base.Source* metodo), 255
- ## D
- data_for_find_identical_books() (*calibre.db.cache.Cache* metodo), 348
 - data_for_has_book() (*calibre.db.cache.Cache* metodo), 348
 - debug_managed_device_detection() (*calibre.devices.interface.DevicePlugin* metodo), 262

`deepcopy()` (*calibre.ebooks.metadata.book.base.Metadata*
metodo), 201
`default_cover()` (*calibre.web.feeds.news.BasicNewsRecipe*
metodo), 38
`delay` (*calibre.web.feeds.news.BasicNewsRecipe* *attribu-*
to), 43
`delete_books()` (*calibre.devices.interface.DevicePlugin*
metodo), 265
`delete_books()` (*calibre.devices.usbms.driver.USBMS*
metodo), 273
`delete_custom_book_data()` (*calibre.db.cache.Cache*
metodo), 348
`delete_trash_entry()` (*calibre.db.cache.Cache*
metodo), 348
`description` (*calibre.customize.conversion.OutputFormatPlugin-*
property), 260
`description` (*calibre.customize.Plugin* *attributo*), 248
`description` (*calibre.customize.PreferencesPlugin* *attri-*
buto), 278
`description` (*calibre.devices.usbms.driver.USBMS* *at-*
tributo), 272
`description` (*calibre.web.feeds.news.BasicNewsRecipe*
attributo), 43
`detect_managed_devices()` (*calibre.devices.interface.DevicePlugin*
metodo), 262
`Device` (*classe in calibre.devices.usbms.device*), 269
`DevicePlugin` (*classe in calibre.devices.interface*), 260
`dirty()` (*calibre.ebooks.oeb.polish.container.Container*
metodo), 355
`do_user_config()` (*calibre.customize.Plugin* *metodo*),
249
`dont_add_to` (*calibre.gui2.actions.InterfaceAction* *attri-*
buto), 275
`dont_remove_from` (*calibre.gui2.actions.InterfaceAction*
attributo), 275
`download()` (*calibre.web.feeds.news.BasicNewsRecipe*
metodo), 38
`download_cover()` (*calibre.ebooks.metadata.sources.base.Source*
metodo), 257
`drop_event()` (*calibre.gui2.actions.InterfaceAction* *me-*
todo), 275

E

`ebook-convert` command line option
`--asciiize`, 321
`--author-sort`, 326
`--authors`, 326
`--base-font-size`, 321
`--book-producer`, 326
`--change-justification`, 321
`--chapter`, 325
`--chapter-mark`, 325
`--comments`, 326
`--cover`, 327
`--debug-pipeline`, 327
`--disable-dehyphenate`, 324
`--disable-delete-blank-paragraphs`,
324
`--disable-fix-indents`, 324
`--disable-font-rescaling`, 321
`--disable-format-scene-breaks`, 324
`--disable-italicize-common-cases`,
324
`--disable-markup-chapter-headings`,
324
`--disable-remove-fake-margins`, 325
`--disable-renumber-headings`, 324
`--disable-unwrap-lines`, 324
`--dont-split-on-page-breaks`, 320
`--duplicate-links-in-toc`, 326
`--embed-all-fonts`, 321
`--embed-font-family`, 321
`--enable-heuristics`, 324
`--epub-flatten`, 320
`--epub-inline-toc`, 320
`--epub-max-image-size`, 320
`--epub-toc-at-end`, 320
`--epub-version`, 320
`--expand-css`, 321
`--extra-css`, 322
`--extract-to`, 320
`--filter-css`, 322
`--flow-size`, 320
`--font-size-mapping`, 322
`--from-opf`, 327
`--help`, 319
`--html-unwrap-factor`, 324
`--input-encoding`, 320
`--input-profile`, 319
`--insert-blank-line`, 322
`--insert-blank-line-size`, 322
`--insert-metadata`, 325
`--isbn`, 327
`--keep-ligatures`, 322
`--language`, 327
`--level1-toc`, 326
`--level2-toc`, 326
`--level3-toc`, 326
`--line-height`, 322
`--linearize-tables`, 322
`--list-recipes`, 319
`--margin-bottom`, 322

```

--margin-left, 322
--margin-right, 322
--margin-top, 323
--max-toc-links, 326
--minimum-line-height, 323
--no-chapters-in-toc, 326
--no-default-epub-cover, 320
--no-svg-cover, 320
--output-profile, 319
--page-breaks-before, 325
--prefer-metadata-cover, 325
--preserve-cover-aspect-ratio, 321
--pretty-print, 321
--pubdate, 327
--publisher, 327
--rating, 327
--read-metadata-from-opf, 327
--remove-first-image, 325
--remove-paragraph-spacing, 323
--remove-paragraph-
    spacing-indent-size,
    323
--replace-scene-breaks, 324
--search-replace, 324
--series, 327
--series-index, 327
--smarten-punctuation, 323
--sr1-replace, 324
--sr1-search, 324
--sr2-replace, 325
--sr2-search, 325
--sr3-replace, 325
--sr3-search, 325
--start-reading-at, 325
--subset-embedded-fonts, 323
--tags, 327
--timestamp, 327
--title, 327
--title-sort, 327
--toc-filter, 326
--toc-threshold, 326
--toc-title, 321
--transform-css-rules, 323
--transform-html-rules, 323
--unsmarten-punctuation, 323
--use-auto-toc, 326
--verbose, 327
--version, 319
-d, 327
-h, 319
-m, 327
-v, 327
ebook-edit command line option
--detach, 328
--help, 328
--select-text, 328
--version, 328
-h, 328
ebook-meta command line option
--author-sort, 328
--authors, 328
--book-producer, 328
--category, 328
--comments, 329
--cover, 329
--date, 329
--from-opf, 329
--get-cover, 329
--help, 329
--identifier, 329
--index, 329
--isbn, 329
--language, 329
--lrf-bookid, 329
--publisher, 329
--rating, 329
--series, 329
--tags, 329
--title, 329
--title-sort, 329
--to-opf, 329
--version, 330
-a, 328
-c, 329
-d, 329
-h, 329
-i, 329
-k, 328
-l, 329
-p, 329
-r, 329
-s, 329
-t, 329
ebook-polish command line option
-H, 330
-U, 331
--add-soft-hyphens, 330
--compress-images, 330
--cover, 330
--embed-fonts, 330
--help, 330
--jacket, 330
--opf, 330
--remove-jacket, 330
--remove-soft-hyphens, 330
--remove-unused-css, 330
--smarten-punctuation, 331
--subset-fonts, 331

```

--upgrade-book, 331
 --verbose, 331
 --version, 331
 -c, 330
 -e, 330
 -f, 331
 -h, 330
 -i, 330
 -j, 330
 -o, 330
 -p, 331
 -u, 330
 ebook-viewer command line option
 --continue, 331
 --detach, 331
 --force-reload, 331
 --full-screen, 331
 --fullscreen, 331
 --help, 331
 --open-at, 331
 --raise-window, 332
 --version, 332
 -f, 331
 -h, 331
 edit_file() (*calibre.gui2.tweak_book.boss.Boss metodo*), 363
 eject() (*calibre.devices.interface.DevicePlugin metodo*), 263
 eject() (*calibre.devices.usbms.device.Device metodo*), 271
 embed_metadata() (*calibre.db.cache.Cache metodo*), 348
 encoding (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 43
 exists() (*calibre.ebooks.oeb.polish.container.Container metodo*), 355
 expire_old_trash() (*calibre.db.cache.Cache metodo*), 348
 extra_css (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 43
 extract_readable_article() (*calibre.web.feeds.news.BasicNewsRecipe metodo*), 38
F
 fast_field_for() (*calibre.db.cache.Cache metodo*), 348
 feeds (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 43
 fetch-ebook-metadata command line option
 -I, 332
 --allowed-plugin, 332
 --authors, 332
 --cover, 332
 --help, 332
 --identifier, 332
 --isbn, 332
 --opf, 332
 --timeout, 332
 --title, 332
 --verbose, 333
 --version, 333
 -a, 332
 -c, 332
 -d, 332
 -h, 332
 -i, 332
 -o, 332
 -p, 332
 -t, 332
 -v, 333
 field_for() (*calibre.db.cache.Cache metodo*), 348
 field_ids_for() (*calibre.db.cache.Cache metodo*), 348
 file_type (*calibre.customize.conversion.OutputFormatPlugin attributo*), 259
 file_types (*calibre.customize.CatalogPlugin attributo*), 253
 file_types (*calibre.customize.conversion.InputFormatPlugin attributo*), 258
 file_types (*calibre.customize.FileTypePlugin attributo*), 250
 file_types (*calibre.customize.MetadataReaderPlugin attributo*), 252
 file_types (*calibre.customize.MetadataWriterPlugin attributo*), 253
 filename_callback() (*calibre.devices.usbms.device.Device metodo*), 272
 filesize() (*calibre.ebooks.oeb.polish.container.Container metodo*), 355
 FileTypePlugin (*classe in calibre.customize*), 250
 filter_css() (*nel modulo calibre.ebooks.oeb.polish.css*), 361
 filter_regexp (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 43
 find_identical_books() (*calibre.db.cache.Cache metodo*), 348
 fix_all_html() (*nel modulo calibre.ebooks.oeb.polish.pretty*), 359
 fix_html() (*nel modulo calibre.ebooks.oeb.polish.pretty*), 359

for_viewer(*calibre.customize.conversion.InputFormatPlugin* *attributo*), 258
 format(*calibre.db.cache.Cache* *metodo*), 348
 format_abspath(*calibre.db.cache.Cache* *metodo*), 349
 format_added(*calibre.db.cache.Cache.EventType* *attributo*), 345
 format_field(*calibre.ebooks.metadata.book.base.Metadata* *metodo*), 202
 format_hash(*calibre.db.cache.Cache* *metodo*), 349
 format_metadata(*calibre.db.cache.Cache* *metodo*), 349
 FORMATS(*calibre.devices.interface.DevicePlugin* *attributo*), 260
 FORMATS(*calibre.devices.usbms.driver.USBMS* *attributo*), 272
 formats(*calibre.db.cache.Cache* *metodo*), 349
 formats_removed(*calibre.db.cache.Cache.EventType* *attributo*), 346
 free_space(*calibre.devices.interface.DevicePlugin* *metodo*), 264
 free_space(*calibre.devices.usbms.device.Device* *metodo*), 270
 from_files(*nel modulo calibre.ebooks.oeb.polish.toc*), 361
 from_links(*nel modulo calibre.ebooks.oeb.polish.toc*), 361
 from_xpaths(*nel modulo calibre.ebooks.oeb.polish.toc*), 361

G

generate_item(*calibre.ebooks.oeb.polish.container.Container* *metodo*), 355
 genesis(*calibre.gui2.actions.InterfaceAction* *metodo*), 276
 genesis(*calibre.gui2.preferences.ConfigWidgetInterface* *metodo*), 279
 get_all_link_maps_for_book(*calibre.db.cache.Cache* *metodo*), 349
 get_all_standard_metadata(*calibre.ebooks.metadata.book.base.Metadata* *metodo*), 202
 get_all_user_metadata(*calibre.ebooks.metadata.book.base.Metadata* *metodo*), 202
 get_annotations(*calibre.devices.usbms.device.Device* *metodo*), 272
 get_article_url(*calibre.web.feeds.news.BasicNewsRecipe* *metodo*), 38
 get_author_tokens(*calibre.ebooks.metadata.sources.base.Source* *metodo*), 255
 get_book_url(*calibre.ebooks.metadata.sources.base.Source* *metodo*), 256
 get_book_url_name(*calibre.ebooks.metadata.sources.base.Source* *metodo*), 256
 get_book_urls(*calibre.ebooks.metadata.sources.base.Source* *metodo*), 256
 get_browser(*calibre.web.feeds.news.BasicNewsRecipe* *metodo*), 38
 get_cached_cover_url(*calibre.ebooks.metadata.sources.base.Source* *metodo*), 256
 get_categories(*calibre.db.cache.Cache* *metodo*), 349
 get_collections(*calibre.devices.interface.BookList* *metodo*), 268
 get_cover_url(*calibre.web.feeds.news.BasicNewsRecipe* *metodo*), 39
 get_custom_book_data(*calibre.db.cache.Cache* *metodo*), 349
 get_device_information(*calibre.devices.interface.DevicePlugin* *metodo*), 264
 get_device_information(*calibre.devices.usbms.driver.USBMS* *metodo*), 272
 get_device_uid(*calibre.devices.interface.DevicePlugin* *metodo*), 266
 get_driveinfo(*calibre.devices.interface.DevicePlugin* *metodo*), 264
 get_extra_css(*calibre.web.feeds.news.BasicNewsRecipe* *metodo*), 39
 get_feeds(*calibre.web.feeds.news.BasicNewsRecipe* *metodo*), 39
 get_file(*calibre.devices.interface.DevicePlugin* *metodo*), 265
 get_file_path_for_processing(*calibre.ebooks.oeb.polish.container.Container* *metodo*), 355
 get_id_map(*calibre.db.cache.Cache* *metodo*), 349

- `get_identifiers()` (*calibre.ebooks.metadata.book.base.Metadata metodo*), 201
- `get_ids_for_custom_book_data()` (*calibre.db.cache.Cache metodo*), 349
- `get_images()` (*calibre.customize.conversion.InputFormatPlugin metodo*), 258
- `get_item_id()` (*calibre.db.cache.Cache metodo*), 350
- `get_item_ids()` (*calibre.db.cache.Cache metodo*), 350
- `get_item_name()` (*calibre.db.cache.Cache metodo*), 350
- `get_link_map()` (*calibre.db.cache.Cache metodo*), 350
- `get_masthead_title()` (*calibre.web.feeds.news.BasicNewsRecipe metodo*), 39
- `get_masthead_url()` (*calibre.web.feeds.news.BasicNewsRecipe metodo*), 39
- `get_metadata()` (*calibre.customize.MetadataReaderPlugin metodo*), 252
- `get_metadata()` (*calibre.db.cache.Cache metodo*), 350
- `get_next_series_num_for()` (*calibre.db.cache.Cache metodo*), 350
- `get_obfuscated_article()` (*calibre.web.feeds.news.BasicNewsRecipe metodo*), 39
- `get_open_popup_message()` (*calibre.devices.interface.DevicePlugin metodo della classe*), 262
- `get_option()` (*calibre.devices.interface.DevicePlugin metodo*), 267
- `get_proxy_metadata()` (*calibre.db.cache.Cache metodo*), 350
- `get_recommended_folders()` (*nel modulo calibre.ebooks.oeb.polish.replace*), 358
- `get_standard_metadata()` (*calibre.ebooks.metadata.book.base.Metadata metodo*), 201
- `get_title_tokens()` (*calibre.ebooks.metadata.sources.base.Source metodo*), 255
- `get_url_specific_delay()` (*calibre.web.feeds.news.BasicNewsRecipe metodo*), 39
- `get_usage_count_by_id()` (*calibre.db.cache.Cache metodo*), 350
- `get_user_blacklisted_devices()` (*calibre.devices.interface.DevicePlugin metodo*), 266
- `get_user_metadata()` (*calibre.ebooks.metadata.book.base.Metadata metodo*), 202
- `gui` (*calibre.gui2.tweak_book.plugin.Tool property*), 362
- `gui_category` (*calibre.customize.PreferencesPlugin attributo*), 278
- `gui_configuration_widget()` (*calibre.customize.conversion.InputFormatPlugin metodo*), 259
- `gui_configuration_widget()` (*calibre.customize.conversion.OutputFormatPlugin metodo*), 260
- `gui_layout_complete()` (*calibre.gui2.actions.InterfaceAction metodo*), 277
- `gui_name` (*calibre.customize.PreferencesPlugin attributo*), 278
- `guide_type_map` (*calibre.ebooks.oeb.polish.container.Container property*), 355
- ## H
- `handle_gzip` (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 43
- `has_book()` (*calibre.db.cache.Cache metodo*), 350
- `has_format()` (*calibre.db.cache.Cache metodo*), 350
- `has_html_comments` (*calibre.ebooks.metadata.sources.base.Source attributo*), 254
- `has_id()` (*calibre.db.cache.Cache metodo*), 350
- `has_name()` (*calibre.ebooks.oeb.polish.container.Container metodo*), 355
- `href_to_name()` (*calibre.ebooks.oeb.polish.container.Container metodo*), 355
- HTML, **369**
- ## I
- `icon` (*calibre.customize.PreferencesPlugin attributo*), 278
- `icon` (*calibre.devices.interface.DevicePlugin attributo*), 261
- `id_from_url()` (*calibre.ebooks.metadata.sources.base.Source metodo*), 256
- `identify()` (*calibre.ebooks.metadata.sources.base.Source metodo*), 256
- `identify_results_keygen()` (*calibre.ebooks.metadata.sources.base.Source metodo*), 256

- `ignore_connected_device()` (*calibre.devices.interface.DevicePlugin* metodo), 266
- `ignore_duplicate_articles` (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 43
- `ignore_ssl_errors` (*calibre.ebooks.metadata.sources.base.Source* attributo), 254
- `image_url_processor()` (*calibre.web.feeds.news.BasicNewsRecipe* della classe), 39
- `index_to_soup()` (*calibre.web.feeds.news.BasicNewsRecipe* metodo), 39
- `indexing_progress_changed` (*calibre.db.cache.Cache.EventType* attributo), 346
- `init()` (*calibre.db.cache.Cache* metodo), 350
- `initial_tab_changed()` (*calibre.gui2.preferences.ConfigWidgetInterface* metodo), 279
- `initialization_complete()` (*calibre.gui2.actions.InterfaceAction* metodo), 277
- `initialize()` (*calibre.customize.CatalogPlugin* metodo), 253
- `initialize()` (*calibre.customize.Plugin* metodo), 249
- `initialize()` (*calibre.gui2.preferences.ConfigWidgetBase* metodo), 280
- `initialize()` (*calibre.gui2.preferences.ConfigWidgetInterface* metodo), 279
- `InputFormatPlugin` (classe in *calibre.customize.conversion*), 258
- `insert_into_xml()` (*calibre.ebooks.oeb.polish.container.Container* metodo), 355
- `installation_type` (*calibre.customize.Plugin* attributo), 249
- `InterfaceAction` (classe in *calibre.gui2.actions*), 274
- `InterfaceActionBase` (classe in *calibre.customize*), 277
- `InternalMetadataCompareKeyGen` (classe in *calibre.ebooks.metadata.sources.base*), 257
- `is_configured()` (*calibre.ebooks.metadata.sources.base.Source* metodo), 255
- `is_dir` (*calibre.ebooks.oeb.polish.container.Container* attributo), 355
- `is_dynamically_controllable()` (*calibre.devices.interface.DevicePlugin* metodo), 267
- `is_image_collection` (*calibre.customize.conversion.InputFormatPlugin* attributo), 258
- `is_link_wanted()` (*calibre.web.feeds.news.BasicNewsRecipe* metodo), 39
- `is_null()` (*calibre.ebooks.metadata.book.base.Metadata* metodo), 201
- `is_running()` (*calibre.devices.interface.DevicePlugin* metodo), 267
- `is_usb_connected()` (*calibre.devices.interface.DevicePlugin* metodo), 262
- `items_removed` (*calibre.db.cache.Cache.EventType* attributo), 346
- `items_renamed` (*calibre.db.cache.Cache.EventType* attributo), 346
- `iterlinks()` (*calibre.ebooks.oeb.polish.container.Container* metodo), 355
- ## K
- `keep_only_tags` (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 44
- ## L
- `language` (*calibre.web.feeds.news.BasicNewsRecipe* attributo), 44
- `library_about_to_change()` (*calibre.gui2.actions.InterfaceAction* metodo), 276
- `library_changed()` (*calibre.gui2.actions.InterfaceAction* metodo), 277
- `list_extra_files()` (*calibre.db.cache.Cache* metodo), 350
- `load_actual_plugin()` (*calibre.customize.InterfaceActionBase* metodo), 277
- `load_resources()` (*calibre.customize.Plugin* metodo), 249
- `load_resources()` (*calibre.gui2.actions.InterfaceAction* metodo), 276
- `location_selected()` (*calibre.gui2.actions.InterfaceAction* metodo), 276
- LRF, 369**
- `lrf2lrs` command line option
- `--dont-output-resources`, 333
 - `--help`, 333
 - `--output`, 333
 - `--verbose`, 333
 - `--version`, 333

-h, 333
 -o, 333
 lrfviewer command line option
 --disable-hyphenation, 334
 --help, 334
 --profile, 334
 --verbose, 334
 --version, 334
 --visual-debug, 334
 --white-background, 334
 -h, 334
 lrs2lrf command line option
 --help, 334
 --lrs, 334
 --output, 334
 --verbose, 334
 --version, 334
 -h, 334
 -o, 334

M

make_name_unique() (calibre.ebooks.oeb.polish.container.Container metodo), 356
 MANAGES_DEVICE_PRESENCE (calibre.devices.interface.DevicePlugin attributo), 261
 manifest_has_name() (calibre.ebooks.oeb.polish.container.Container metodo), 356
 manifest_id_map (calibre.ebooks.oeb.polish.container.Container property), 356
 manifest_items_of_type() (calibre.ebooks.oeb.polish.container.Container metodo), 356
 manifest_items_with_property() (calibre.ebooks.oeb.polish.container.Container metodo), 356
 manifest_type_map (calibre.ebooks.oeb.polish.container.Container property), 356
 mark_as_cover() (nel modulo calibre.ebooks.oeb.polish.cover), 360
 mark_as_titlepage() (nel modulo calibre.ebooks.oeb.polish.cover), 360
 masthead_url (calibre.web.feeds.news.BasicNewsRecipe attributo), 44
 match_regexps (calibre.web.feeds.news.BasicNewsRecipe attributo), 44

max_articles_per_feed (calibre.web.feeds.news.BasicNewsRecipe attributo), 44
 MAX_PATH_LEN (calibre.devices.usbms.device.Device attributo), 270
 merge() (nel modulo calibre.ebooks.oeb.polish.split), 359
 merge_extra_files() (calibre.db.cache.Cache metodo), 351
 Metadata (classe in calibre.ebooks.metadata.book.base), 201
 metadata_changed (calibre.db.cache.Cache.EventType attributo), 346
 metadata_for_field() (calibre.ebooks.metadata.book.base.Metadata metodo), 201
 MetadataReaderPlugin (classe in calibre.customize), 252
 MetadataWriterPlugin (classe in calibre.customize), 252
 mi (calibre.ebooks.oeb.polish.container.Container property), 356
 minimum_calibre_version (calibre.customize.Plugin attributo), 249
 module
 calibre.customize, 248
 calibre.customize.conversion, 258
 calibre.db.cache, 345
 calibre.devices.interface, 260
 calibre.ebooks.metadata.book.base, 201
 calibre.ebooks.metadata.sources.base, 254
 calibre.ebooks.oeb.polish.container, 354
 calibre.ebooks.oeb.polish.cover, 360
 calibre.ebooks.oeb.polish.css, 360
 calibre.ebooks.oeb.polish.jacket, 359
 calibre.ebooks.oeb.polish.pretty, 359
 calibre.ebooks.oeb.polish.replace, 358
 calibre.ebooks.oeb.polish.split, 359
 calibre.ebooks.oeb.polish.toc, 361
 calibre.gui2.tweak_book.boss, 363
 calibre.utils.formatter_functions, 181
 calibre.web.feeds.news, 37
 move_book_from_trash() (calibre.db.cache.Cache metodo), 351
 move_format_from_trash() (calibre.db.cache.Cache metodo),

351
 multisort() (*calibre.db.cache.Cache metodo*), 351
 multisplit() (*nel modulo calibre.ebooks.oeb.polish.split*), 359

N

name (*calibre.customize.Plugin attributo*), 248
 name (*calibre.gui2.actions.InterfaceAction attributo*), 274
 name (*calibre.gui2.tweak_book.plugin.Tool attributo*), 362
 name(), **153**
 name_order (*calibre.customize.PreferencesPlugin attributo*), 278
 name_to_abspath() (*calibre.ebooks.oeb.polish.container.Container metodo*), 356
 name_to_href() (*calibre.ebooks.oeb.polish.container.Container metodo*), 356
 names_that_must_not_be_changed (*calibre.ebooks.oeb.polish.container.Container property*), 356
 names_that_must_not_be_removed (*calibre.ebooks.oeb.polish.container.Container property*), 356
 names_that_need_not_be_manifested (*calibre.ebooks.oeb.polish.container.Container property*), 356
 needs_subscription (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 44
 NEWS_IN_FOLDER (*calibre.devices.usbms.device.Device attributo*), 270
 no_stylesheets (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 44
 normalize_path() (*calibre.devices.usbms.driver.USBMS metodo della classe*), 274
 NUKE_COMMENTS (*calibre.devices.interface.DevicePlugin attributo*), 261

O

oldest_article (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 44
 on_import (*calibre.customize.FileTypePlugin attributo*), 250
 on_postconvert (*calibre.customize.FileTypePlugin attributo*), 250
 on_postdelete (*calibre.customize.FileTypePlugin attributo*), 251
 on_postimport (*calibre.customize.FileTypePlugin attributo*), 250

on_postprocess (*calibre.customize.FileTypePlugin attributo*), 251
 on_preprocess (*calibre.customize.FileTypePlugin attributo*), 251
 open() (*calibre.devices.interface.DevicePlugin metodo*), 263
 open() (*calibre.devices.usbms.device.Device metodo*), 271
 open() (*calibre.ebooks.oeb.polish.container.Container metodo*), 356
 open_book() (*calibre.gui2.tweak_book.boss.Boss metodo*), 363
 OPEN_FEEDBACK_MESSAGE (*calibre.devices.interface.DevicePlugin attributo*), 261
 opf (*calibre.ebooks.oeb.polish.container.Container property*), 356
 opf_get_or_create() (*calibre.ebooks.oeb.polish.container.Container metodo*), 356
 opf_version (*calibre.ebooks.oeb.polish.container.Container property*), 356
 opf_version_parsed (*calibre.ebooks.oeb.polish.container.Container property*), 356
 opf_xpath() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357
 options (*calibre.customize.conversion.InputFormatPlugin attributo*), 258
 options (*calibre.customize.conversion.OutputFormatPlugin attributo*), 259
 options (*calibre.ebooks.metadata.sources.base.Source attributo*), 254
 OSX_MAIN_MEM_VOL_PAT (*calibre.devices.usbms.device.Device attributo*), 270
 output_encoding (*calibre.customize.conversion.InputFormatPlugin attributo*), 258
 OutputFormatPlugin (*classe in calibre.customize.conversion*), 259

P

parse_feeds() (*calibre.web.feeds.news.BasicNewsRecipe metodo*), 40
 parse_index() (*calibre.web.feeds.news.BasicNewsRecipe metodo*), 40
 parsed() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357
 path_sep (*calibre.devices.interface.DevicePlugin attributo*), 261
 Plugin (*classe in calibre.customize*), 248

- poblare_article_metadata() (calibre.web.feeds.news.BasicNewsRecipe metodo), 40
-
- popup_type (calibre.gui2.actions.InterfaceAction attributo), 275
-
- post_yank_cleanup() (calibre.devices.interface.DevicePlugin metodo), 263
-
- post_yank_cleanup() (calibre.devices.usbms.device.Device metodo), 271
-
- postadd() (calibre.customize.FileTypePlugin metodo), 252
-
- postconvert() (calibre.customize.FileTypePlugin metodo), 251
-
- postdelete() (calibre.customize.FileTypePlugin metodo), 251
-
- postimport() (calibre.customize.FileTypePlugin metodo), 251
-
- postprocess_book() (calibre.customize.conversion.InputFormatPlugin metodo), 259
-
- postprocess_book() (calibre.web.feeds.news.BasicNewsRecipe metodo), 40
-
- postprocess_html() (calibre.web.feeds.news.BasicNewsRecipe metodo), 40
-
- pref() (calibre.db.cache.Cache metodo), 351
-
- prefer_results_with_isbn (calibre.ebooks.metadata.sources.base.Source attributo), 255
-
- PreferencesPlugin (classe in calibre.customize), 278
-
- prepare_addable_books() (calibre.devices.interface.DevicePlugin metodo), 266
-
- preprocess_html() (calibre.web.feeds.news.BasicNewsRecipe metodo), 41
-
- preprocess_image() (calibre.web.feeds.news.BasicNewsRecipe metodo), 41
-
- preprocess_raw_html() (calibre.web.feeds.news.BasicNewsRecipe metodo), 41
-
- preprocess_regexp (calibre.web.feeds.news.BasicNewsRecipe attributo), 44
-
- pretty_all() (nel modulo calibre.ebooks.oeb.polish.pretty), 359
-
- pretty_css() (nel modulo calibre.ebooks.oeb.polish.pretty), 359
-
- pretty_html() (nel modulo calibre.ebooks.oeb.polish.pretty), 359
-
- pretty_xml() (nel modulo calibre.ebooks.oeb.polish.pretty), 359
-
- print_version() (calibre.web.feeds.news.BasicNewsRecipe metodo della classe), 41
-
- priority (calibre.customize.Plugin attributo), 249
-
- priority (calibre.gui2.actions.InterfaceAction attributo), 274
-
- PRODUCT_ID (calibre.devices.interface.DevicePlugin attributo), 261
-
- PRODUCT_ID (calibre.devices.usbms.device.Device attributo), 269
-
- publication_date() (calibre.web.feeds.news.BasicNewsRecipe metodo), 41
-
- publication_type (calibre.web.feeds.news.BasicNewsRecipe attributo), 44
-
- R**
-
- raw_data() (calibre.ebooks.oeb.polish.container.Container metodo), 357
-
- re:test(), 153
-
- read_backup() (calibre.db.cache.Cache metodo), 351
-
- recipe_disabled (calibre.web.feeds.news.BasicNewsRecipe attributo), 45
-
- recommendations (calibre.customize.conversion.InputFormatPlugin attributo), 258
-
- recommendations (calibre.customize.conversion.OutputFormatPlugin attributo), 259
-
- recursions (calibre.web.feeds.news.BasicNewsRecipe attributo), 45
-
- refresh_gui() (calibre.gui2.preferences.ConfigWidgetInterface metodo), 279
-
- regex, 369
-
- register() (calibre.gui2.preferences.ConfigWidgetBase metodo), 280
-
- register_shortcut() (calibre.gui2.tweak_book.plugin.Tool metodo), 362
-
- relpath() (calibre.ebooks.oeb.polish.container.Container metodo), 357
-
- remove_attributes (calibre.web.feeds.news.BasicNewsRecipe attributo), 45

remove_book() (*calibre.devices.interface.BookList metodo*), 268

remove_books() (*calibre.db.cache.Cache metodo*), 351

remove_books_from_metadata() (*calibre.devices.interface.DevicePlugin metodo della classe*), 265

remove_books_from_metadata() (*calibre.devices.usbms.driver.USBMS metodo*), 273

remove_empty_feeds (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 45

remove_formats() (*calibre.db.cache.Cache metodo*), 351

remove_from_spine() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357

remove_from_xml() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357

remove_item() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357

remove_items() (*calibre.db.cache.Cache metodo*), 351

remove_jacket() (*nel modulo calibre.ebooks.oeb.polish.jacket*), 359

remove_javascript (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 45

remove_stale_user_metadata() (*calibre.ebooks.metadata.book.base.Metadata metodo*), 202

remove_tags (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 45

remove_tags_after (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 45

remove_tags_before (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 45

remove_unused_css() (*nel modulo calibre.ebooks.oeb.polish.css*), 360

rename() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357

rename_extra_files() (*calibre.db.cache.Cache metodo*), 351

rename_files() (*nel modulo calibre.ebooks.oeb.polish.replace*), 358

rename_items() (*calibre.db.cache.Cache metodo*), 352

replace() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357

replace_links() (*calibre.ebooks.oeb.polish.container.Container metodo*), 357

replace_links() (*nel modulo calibre.ebooks.oeb.polish.replace*), 358

requires_version (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 45

reset() (*calibre.devices.interface.DevicePlugin metodo*), 262

reset() (*calibre.devices.usbms.device.Device metodo*), 270

resolve_internal_links (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 45

restart_critical (*calibre.gui2.preferences.ConfigWidgetBase attributo*), 279

restart_critical (*calibre.gui2.preferences.ConfigWidgetInterface attributo*), 279

restore_book() (*calibre.db.cache.Cache metodo*), 352

restore_defaults() (*calibre.gui2.preferences.ConfigWidgetBase metodo*), 280

restore_defaults() (*calibre.gui2.preferences.ConfigWidgetInterface metodo*), 279

restore_defaults_desc (*calibre.gui2.preferences.ConfigWidgetInterface attributo*), 279

restore_original_format() (*calibre.db.cache.Cache metodo*), 352

reverse_article_order (*calibre.web.feeds.news.BasicNewsRecipe attributo*), 46

rewind_savepoint() (*calibre.gui2.tweak_book.boss.Boss metodo*), 364

Ricetta, **369**

RSS, **369**

run() (*calibre.customize.CatalogPlugin metodo*), 253

run() (*calibre.customize.FileTypePlugin metodo*), 251

S

safe_read_lock (*calibre.db.cache.Cache property*), 352

<code>sanitize_callback()</code>	(<i>calibre.devices.usbms.device.Device</i> metodo), 271	<code>set_link_map()</code>	(<i>calibre.db.cache.Cache</i> metodo), 353
<code>sanitize_path_components()</code>	(<i>calibre.devices.usbms.device.Device</i> metodo), 272	<code>set_metadata()</code>	(<i>calibre.customize.MetadataWriterPlugin</i> metodo), 253
<code>save_book()</code>	(<i>calibre.gui2.tweak_book.boss.Boss</i> metodo), 364	<code>set_metadata()</code>	(<i>calibre.db.cache.Cache</i> metodo), 353
<code>save_original_format()</code>	(<i>calibre.db.cache.Cache</i> metodo), 352	<code>set_modified()</code>	(<i>calibre.gui2.tweak_book.boss.Boss</i> metodo), 364
<code>save_settings()</code>	(<i>calibre.customize.Plugin</i> metodo), 249	<code>set_option()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo), 267
<code>save_settings()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo della classe), 266	<code>set_plugboards()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo), 266
<code>save_settings()</code>	(<i>calibre.ebooks.metadata.sources.base.Source</i> metodo), 255	<code>set_pref()</code>	(<i>calibre.db.cache.Cache</i> metodo), 353
<code>scale_news_images</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 46	<code>set_progress_reporter()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo), 263
<code>scale_news_images_to_device</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 46	<code>set_progress_reporter()</code>	(<i>calibre.devices.usbms.device.Device</i> metodo), 270
<code>search()</code>	(<i>calibre.db.cache.Cache</i> metodo), 352	<code>set_spine()</code>	(<i>calibre.ebooks.oeb.polish.container.Container</i> metodo), 357
<code>serialize_item()</code>	(<i>calibre.ebooks.oeb.polish.container.Container</i> metodo), 357	<code>set_user_blacklisted_devices()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo), 266
<code>set_all_user_metadata()</code>	(<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 202	<code>set_user_metadata()</code>	(<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 202
<code>set_conversion_options()</code>	(<i>calibre.db.cache.Cache</i> metodo), 352	<code>settings()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo della classe), 266
<code>set_cover()</code>	(<i>calibre.db.cache.Cache</i> metodo), 352	<code>show_current_diff()</code>	(<i>calibre.gui2.tweak_book.boss.Boss</i> metodo), 364
<code>set_cover()</code>	(<i>nel modulo calibre.ebooks.oeb.polish.cover</i>), 360	<code>show_editor()</code>	(<i>calibre.gui2.tweak_book.boss.Boss</i> metodo), 364
<code>set_driveinfo_name()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo), 266	<code>shutdown()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo), 266
<code>set_driveinfo_name()</code>	(<i>calibre.devices.usbms.driver.USBMS</i> metodo), 272	<code>shutting_down()</code>	(<i>calibre.gui2.actions.InterfaceAction</i> metodo), 277
<code>set_field()</code>	(<i>calibre.db.cache.Cache</i> metodo), 352	<code>simultaneous_downloads</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 46
<code>set_identifier()</code>	(<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 201	<code>skip_ad_pages()</code>	(<i>calibre.web.feeds.news.BasicNewsRecipe</i> metodo), 41
<code>set_identifiers()</code>	(<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 201	<code>SLOW_DRIVEINFO</code>	(<i>calibre.devices.interface.DevicePlugin</i> attributo), 262
<code>set_library_info()</code>	(<i>calibre.devices.interface.DevicePlugin</i> metodo), 267	<code>smart_update()</code>	(<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 202

<code>sort_index_by()</code> (<i>calibre.web.feeds.news.BasicNewsRecipe</i> metodo), 41	<code>supported_platforms</code> (<i>calibre.customize.MetadataReaderPlugin</i> attributo), 252
<code>Source</code> (classe in <i>calibre.ebooks.metadata.sources.base</i>), 254	<code>supported_platforms</code> (<i>calibre.customize.MetadataWriterPlugin</i> attributo), 253
<code>specialize()</code> (<i>calibre.customize.conversion.InputFormatPlugin</i> metodo), 259	<code>supported_platforms</code> (<i>calibre.customize.Plugin</i> attributo), 248
<code>specialize_css_for_output()</code> (<i>calibre.customize.conversion.OutputFormatPlugin</i> metodo), 260	<code>supported_platforms</code> (<i>calibre.customize.PreferencesPlugin</i> attributo), 278
<code>specialize_global_preferences()</code> (<i>calibre.devices.interface.DevicePlugin</i> metodo), 267	<code>supported_platforms</code> (<i>calibre.devices.usbms.driver.USBMS</i> attributo), 272
<code>specialize_options()</code> (<i>calibre.customize.conversion.OutputFormatPlugin</i> metodo), 260	<code>supported_platforms</code> (<i>calibre.ebooks.metadata.sources.base.Source</i> attributo), 254
<code>spine_items</code> (<i>calibre.ebooks.oeb.polish.container.Container</i> property), 358	<code>supports_collections()</code> (<i>calibre.devices.interface.BookList</i> metodo), 268
<code>spine_iter</code> (<i>calibre.ebooks.oeb.polish.container.Container</i> property), 358	<code>supports_gzip_transfer_encoding</code> (<i>calibre.ebooks.metadata.sources.base.Source</i> attributo), 254
<code>spine_names</code> (<i>calibre.ebooks.oeb.polish.container.Container</i> property), 358	<code>supports_restoring_to_defaults</code> (<i>calibre.gui2.preferences.ConfigWidgetBase</i> attributo), 279
<code>split()</code> (nel modulo <i>calibre.ebooks.oeb.polish.split</i>), 359	<code>supports_restoring_to_defaults</code> (<i>calibre.gui2.preferences.ConfigWidgetInterface</i> attributo), 279
<code>split_if_is_multiple_composite()</code> (<i>calibre.db.cache.Cache</i> metodo), 353	<code>sync_booklists()</code> (<i>calibre.devices.interface.DevicePlugin</i> metodo), 265
<code>split_jobs()</code> (<i>calibre.ebooks.metadata.sources.base.Source</i> metodo), 255	<code>sync_booklists()</code> (<i>calibre.devices.usbms.driver.USBMS</i> metodo), 274
<code>standard_field_keys()</code> (<i>calibre.ebooks.metadata.book.base.Metadata</i> metodo), 201	<code>sync_preview_to_editor()</code> (<i>calibre.gui2.tweak_book.boss.Boss</i> metodo), 364
<code>STANDARD_METADATA_FIELDS</code> (nel modulo <i>calibre.ebooks.metadata.book.base</i>), 202	<code>synchronize_with_db()</code> (<i>calibre.devices.interface.DevicePlugin</i> metodo), 267
<code>start_plugin()</code> (<i>calibre.devices.interface.DevicePlugin</i> metodo), 267	T
<code>startup()</code> (<i>calibre.devices.interface.DevicePlugin</i> metodo), 266	
<code>stop_plugin()</code> (<i>calibre.devices.interface.DevicePlugin</i> metodo), 267	
<code>summary_length</code> (<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 46	
<code>supported_platforms</code> (<i>calibre.customize.conversion.InputFormatPlugin</i> attributo), 258	
<code>supported_platforms</code> (<i>calibre.customize.conversion.OutputFormatPlugin</i> attributo), 259	
<code>supported_platforms</code> (<i>calibre.customize.InterfaceActionBase</i> attributo), 277	
	<code>tag_browser_context_action()</code> (<i>calibre.gui2.actions.InterfaceAction</i> metodo), 277
	<code>tag_to_string()</code> (<i>calibre.web.feeds.news.BasicNewsRecipe</i> metodo della classe), 41
	<code>tags_older_than()</code> (<i>calibre.db.cache.Cache</i> metodo), 353
	<code>template_css</code> (<i>calibre.web.feeds.news.BasicNewsRecipe</i> attributo), 46

- template_to_attribute() (calibre.ebooks.metadata.book.base.Metadata metodo), 202
- temporary_file() (calibre.customize.Plugin metodo), 250
- test_fields() (calibre.ebooks.metadata.sources.base.Source metodo), 255
- THUMBNAIL_COMPRESSION_QUALITY (calibre.devices.interface.DevicePlugin attributo), 261
- THUMBNAIL_HEIGHT (calibre.devices.interface.DevicePlugin attributo), 261
- timefmt (calibre.web.feeds.news.BasicNewsRecipe attributo), 46
- timeout (calibre.web.feeds.news.BasicNewsRecipe attributo), 46
- title (calibre.web.feeds.news.BasicNewsRecipe attributo), 46
- to_html() (calibre.ebooks.metadata.book.base.Metadata metodo), 202
- Tool (classe in calibre.gui2.tweak_book.plugin), 361
- toolbar_button_popup_mode (calibre.gui2.tweak_book.plugin.Tool attributo), 362
- total_space() (calibre.devices.interface.DevicePlugin metodo), 264
- total_space() (calibre.devices.usbms.device.Device metodo), 270
- touched_fields (calibre.ebooks.metadata.sources.base.Source attributo), 254
- type (calibre.customize.CatalogPlugin attributo), 253
- type (calibre.customize.conversion.InputFormatPlugin attributo), 258
- type (calibre.customize.conversion.OutputFormatPlugin attributo), 259
- type (calibre.customize.FileTypePlugin attributo), 251
- type (calibre.customize.InterfaceActionBase attributo), 277
- type (calibre.customize.MetadataReaderPlugin attributo), 252
- type (calibre.customize.MetadataWriterPlugin attributo), 253
- type (calibre.customize.Plugin attributo), 249
- type (calibre.customize.PreferencesPlugin attributo), 278
- type (calibre.devices.interface.DevicePlugin attributo), 260
- type (calibre.ebooks.metadata.sources.base.Source attributo), 254
- upload_books() (calibre.devices.usbms.driver.USBMS metodo), 273
- upload_cover() (calibre.devices.usbms.driver.USBMS metodo), 273
- URL, 369
- USBMS (classe in calibre.devices.usbms.driver), 272
- use_embedded_content (calibre.web.feeds.news.BasicNewsRecipe attributo), 46
- user_categories_for_books() (calibre.db.cache.Cache metodo), 353
- user_feedback_after_callback (calibre.devices.interface.DevicePlugin attributo), 262
- UserAnnotation (calibre.devices.interface.DevicePlugin attributo), 261
- ## V
- VENDOR_ID (calibre.devices.interface.DevicePlugin attributo), 261
- VENDOR_ID (calibre.devices.usbms.device.Device attributo), 269
- version (calibre.customize.MetadataReaderPlugin attributo), 252
- version (calibre.customize.MetadataWriterPlugin attributo), 253
- version (calibre.customize.Plugin attributo), 248
- VIRTUAL_BOOK_EXTENSION_MESSAGE (calibre.devices.interface.DevicePlugin attributo), 261
- VIRTUAL_BOOK_EXTENSIONS (calibre.devices.interface.DevicePlugin attributo), 261
- ## W
- WANTS_UPDATED_THUMBNAILS (calibre.devices.interface.DevicePlugin attributo), 261
- web2disk command line option
- base-dir, 335
 - delay, 335
 - dont-download-stylesheets, 335
 - encoding, 335
 - filter-regexp, 335
 - help, 335
 - match-regexp, 335
 - max-files, 335
 - max-recursions, 335
 - timeout, 335
 - verbose, 335
 - version, 335

U

-d, 335
 -h, 335
 -n, 335
 -r, 335
 -t, 335
 WINDOWS_CARD_A_MEM (cali-
 bre.devices.usbms.device.Device attributo),
 269
 WINDOWS_CARD_B_MEM (cali-
 bre.devices.usbms.device.Device attributo),
 270
 WINDOWS_MAIN_MEM (cali-
 bre.devices.usbms.device.Device attributo),
 269
 windows_sort_drives() (cali-
 bre.devices.usbms.device.Device metodo),
 271