Panduan Pengguna calibre

Release 5.7.2

Kovid Goyal

Disember 12, 2020

Contents

1 Antarmuka Pengguna Bergrafik 3

2 Menambah laman sesawang berita kegemaran anda 27

3 Pelihat E-buku 45

4 E-book conversion 49

5 Menyunting e-buku 67

6 Pelayan Kandungan calibre 101

7 Membandingkan e-buku 109

8 Menyunting data meta e-buku 113

9 Soalan Kerap Ditanya 117

10 Tutorial 139

11 The calibre:// URL scheme 221

12 Penyesuaian calibre. 225

13 Antaramuka Baris Perintah 267

14 Menetapkan persekitaran pembangunan calibre 305

15 Digital Rights Management (DRM) 329

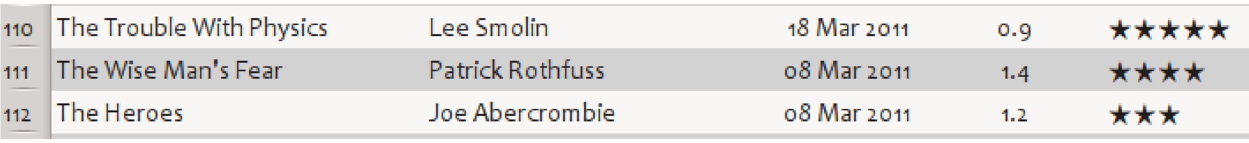
16 Glosari 331

Python Module Index 333

Index 335

calibre merupakan pengurus pustaka e-buku. Ia dapat papar, tukar format dan katalogkan e-buku dalam kebanyakan format e-buku yang popular. Ia juga dapat berkomunikasi dengan kebanyakan peranti pembaca e-buku. Ia dapat mendapatkan data meta buku menerusi sambungan Internet. Ia juga dapat memuat turun surat khabar dan tukar ia menjadi e-buku untuk pembacaan yang lebih mudah. Calibre merupakan platform silang yang dapat dijalankan dalam Linux, Windows dan OS X.

Anda baru sahaja memulakan calibre. Apa perlu dilakukan Sekarang? Sebelum calibre dapat berfungsi dan mengendali e-buku anda, mula-mula anda perlu tahu mengenainya. Seret dan lepas beberapa buah fail e-buku ke dalam calibre, atau klik butang "Tambah buku" dan layar e-buku yang dikehendaki. Selepas menambah buku, ia akan dipapar dalam paparan utama seperti ini:



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	★★★

Selepas anda menyenaraikan buku yang anda sukai, seterusnya anda membaca buku-buku tersebut. Sebelum itu, anda perlu tukar format buku tersebut ke dalam format yang pembaca ebuku anda fahami. Kali pertama menjalankan calibre, guilabel:*Bestari aluan* akan mula memasang dan menetapkan calibre untuk peranti pembaca anda. Proses pertukaran format adalah cepat dan mudah. Hanya pilih buku yang dikehendaki, kemudian klik butang "Tukar format buku". Abaikan semua pilihan buat masa ini, terus klik "OK". Ikon kecil di bucu bawah kanan akan mula berputar. Seusai berputar, buku anda sudah ditukarkan formatnya. Klik butang "Lihat" untuk membaca buku tersebut.

Jika anda mahu membaca ebuku di dalam pembaca ebuku, sambungkan dengan komputer, tunggu sehingga calibre mengesannya (10-20 saat) kemudian klik butang "Hantar ke peranti". Bilamana ikon berhenti berpusing, tanggalkan pembaca ebuku anda dan boleh terus membaca! Jika anda tidak menukar format buku dalam langkah terdahulu, calibre akan auto-tukar ebuku tersebut dalam format pembaca anda fahami.

Untuk memulakan calibre dengan penggunaan lanjutan, anda disaran membaca dahulu *Antarmuka Pengguna Bergrafik* (halaman **??**). Dan lebih mendalam, baca *Antaramuka Baris Perintah* (halaman **??**). Anda akan dapati terdapat senarai *Soalan Kerap Ditanya* (halaman **??**) yang berguna.

Jika anda ada soalan, atau mahu berbincang berkenaan calibre dengan pengguna lain ataupun mahu dapatkan bantuan perkara-perkara khusus, telah disediakan 'forum dan lain-lain sumber bantuan yang disediakan <https://calibre-ebook.com/help>`_.

Seksyen

CHAPTER 1

Antarmuka Pengguna Bergrafik

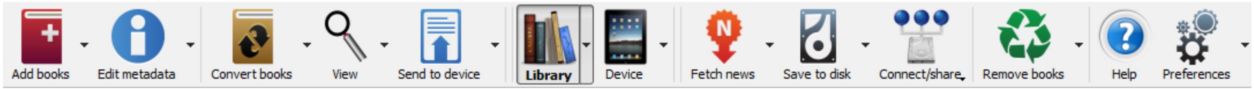
Antaramuka Pengguna Bergrafik *(GUI)* menyediakan capaian ke semua fitur pengurusan pustaka dan pertukaran format e-buku. Aliran kerja asas untuk menggunakan calibre adalah dengan menambah dahulu buku ke dalam pustaka daripada cakera keras anda. calibre akan cuba membaca data meta daripada buku secara automatik dan menambahnya ke dalam pangkalan data dalamannya. Seusai ia berada dalam pangkalan data, anda boleh lakukan pelbagai tindakan pada buku tersebut termasuklah pertukaran format dari satu format ke format buku yang lain, pemindahan ke peranti pembacaan, melihatnya dalam komputer anda, dan menyunting data meta e-buku anda. Penyuntingan melibatkan pengubahsuaian kulit buku, keterangan, dan tag serta lain-lain perincian yang ada. Perhatian, calibre mencipta satu salinan fail yang anda tambah ke dalamnya. Fail asal anda tidak akan diubahsuai.

Antaramuka terbahagi kepada beberpa bahagian:

- *Tindakan* (halaman 4)
- *Pilihan* (halaman 11)
- *Katalog* (halaman 11)
- *Gelintar & Isih* (halaman 12)
- *Antaramuka gelintar* (halaman 13)
- *Menyimpan gelintar* (halaman 16)
- *Pustaka maya* (halaman 16)
- *Meneka data meta dari nama fail* (halaman 16)
- *Perincian buku* (halaman 17)
- *Pelayar tag* (halaman 19)
- *Grid kulit buku* (halaman 21)
- *Pelayar kulit buku* (halaman 22)
- *Paparan Pantas* (halaman 22)

- *Kerja* (halaman 23)
- *Pintasan papan kekunci* (halaman 23)

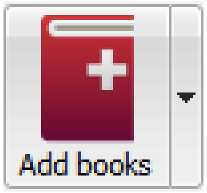
1.1 Tindakan



Palang alat tindakan menyediakan pintasan mudah ke beberapa tindakan yang kerap digunakan. Jika anda mengklik-kanan butang tersebut, anda boleh lakukan variasi pada tindakan lalai. Perhatian, palang alat tindakan akan kelihatan sedikit berbeza bilamana anda menyambungkan pembaca e-buku dengan komputer anda.

- *Tambah buku* (halaman 4)
- *Sunting data meta* (halaman 5)
- *Tukar format buku* (halaman 6)
- *Lihat* (halaman 6)
- *Hantar ke peranti* (halaman 7)
- *Dapatkan berita* (halaman 7)
- *Pustaka* (halaman 8)
- *Peranti* (halaman 9)
- *Simpan ke dalam cakera* (halaman 9)
- *Sambung/kongsi* (halaman 10)
- *Buang buku* (halaman 10)

1.1.1 Tambah buku



Tindakan *Tambah buku* mempunyai tujuh variasi yang dicapai dengan mengklik-kanan pada butang tersebut.

1. **Tambah buku dari satu direktori**: Buka dialog pemilih fail yang membolehkan anda nyatakan buku manakah dalam sesebuah direktori patut ditambah. Tindakan ini adalah *sensitif konteks*, iaitu ia bergantung pada *katalog* (halaman 11) manakah anda pilih. Jika anda telah memilih *Pustaka*, buku akan ditambah ke dalam pustaka. Jika anda telah memilih peranti pembaca e-buku, buku akan dimuat naik ke dalam peranti, dan seterusnya.
2. **Add books from directories and sub-directories**: Allows you to choose a directory. The directory and all its sub-directories 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 directory to a single book record or multiple book records. calibre assumes that each directory contains a single book. All e-book files in a directory are assumed to be the same

book in different formats. This action is the inverse of the *Save to disk* (halaman 9) action, i.e. you can *Save to disk*, delete the books and re-add them in single book per directory 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).

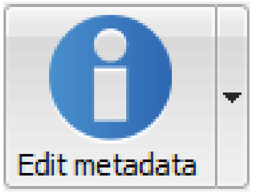
3. **Tambah buku berbilang dari arkib (ZIP/RAR)**: Membolehkan anda tambah e-buku berbilang yang tersimpan di dalam fail ZIP atau RAR yang terpilih. Ia merupakan pintasan mudah yang dapat menghindari proses menyahzip dan tambah buku menerusi salah satu dua pilihan di atas.
4. **Tambah buku kosong. (Masukan Buku tanpa format)**: Membolehkan anda cipta satu rekod buku kosong. Kemudian ia boleh digunakan dengan mengisi secara manual maklumat berkenaan buku yang tiada lagi dalam koleksi anda.
5. **Tambah dari ISBN**: Membolehkan anda tambah satu atau lebih buku dengan memasukkan ISBN mereka.
6. **Tambah fail ke rekod buku terpilih**: Membolehkan anda tambah atau kemaskini fail berkaitan dengan buku sedia yang ada dalam pustaka anda.
7. **Add an empty file to selected book records**: Allows you to add an empty file of the specified format to the selected book records.

Tindakan *Tambah buku* dapat membaca data meta dari pelbagai jenis format e-buku. Selain itu, ia cuba meneka data meta dari nama fail. Sila rujuk seksyen *Meneka data meta dari nama fail* (halaman 16), untuk mengetahui bagaimana hendak konfigurkannya.

Untuk menambah format tambahan bagi buku sedia ada anda boleh lakukan mana-mana tiga langkah berikut:

1. Seret dan lepas fail ke dalam panel perincian Buku di sebelah kanan tetingkap utama
2. Klik kanan butang *Tambah buku* dan pilih *Tambah fail ke buku terpilih*.
3. Click the *Add books* button in the top right area of the *Edit metadata* dialog, accessed by the *Sunting data meta* (halaman 5) action.

1.1.2 Sunting data meta



Tindakan *Sunting data meta* mempunyai empat variasi yang boleh dicapai dengan mengklik-kanan pada butang.

1. **Sunting data meta secara individu**: Membolehkan anda sunting data meta buku satu-demi-satu dengan pilihan mendapatkan data meta, termasuklah kulit buku, dari Internet. Ia juga membolehkan anda tambah atau buang format e-buku tertentu dari sesebuah buku.
2. **Sunting data meta secara pukal**: Membolehkan anda sunting medan data meta umum bagi sejumlah besar buku secara serentak. Ia mengendalikan semua buku yang dipilih dalam *Paparan pustaka* (halaman 12).
3. **Muat turun data meta dan kulit buku**: Muat turun data meta dan kulit buku (jika tersedia) bagi buku yang dipilih dalam senarai buku.
4. **Gabung rekod buku**: Memberikan keupayaan menggabungkan data meta dan format bagi dua atau lebih rekod buku. Anda boleh pilih sama ada hendak memadam atau kekalkan rekod yang belum diklik.

Untuk perincian lanjut sila rujuk *Menyunting data meta e-buku* (halaman 113).

1.1.3 Tukar format buku

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* (halaman **??**) *(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.

Bagi kebanyakan pengguna, pertukaran format hanya kerja sekali-klik. Jika anda mahu ketahui lebih lanjut berkenaan proses pertukaran format, sila rujuk *E-book conversion* (halaman 49).

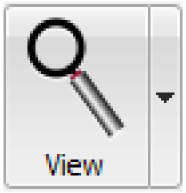
Tindakan *Tukar format buku* mempunyai tiga variasi, yang boleh dicapai dengan mengklik-kanan pada butang tersebut.

1. **Tukar format secara individu**: Membolehkan anda nyatakan pilihan pertukaran untuk membuat penyuaian bagi setiap e-buku yang telah dipilih.

2. **Tukar format pukal**: Membolehkan anda nyatakan pilihan untuk membuat pertukaran format sejumlah e-buku secara pukal.

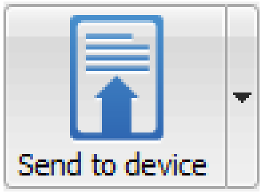
3. **Cipta satu katalog buku dalam pustaka calibre anda**: Membolehkan anda jana satu senarai lengkap buku di dalam pustaka anda, termasuklah semua data meta, dalam beberapa format seperti XML, CSV, BiBTeX, EPUB dan MOBI. Katalog akan mengandungi semua buku semasa yang ditunjukkan dalam paparan pustaka. Ia membolehkan anda guna fitur gelintar untuk membataskan buku yang dikatalogkan. Selain itu, jika anda memilih lebih dari sebuah buku menggunakan tetikus, hanya buku tersebut akan ditambah ke dalam katalog. Jika anda menjana katalog dalam format e-buku seperti EPUB, MOBI atau AZW3, dilain masa anda menyambung pembaca e-buku, katalog akan dihantar secara automatik ke dalam peranti tersebut. Untuk maklumat lanjut bagaimana katalog berfungsi, rujuk *Penciptaan Katalog AZW3 EPUB MOBI* (halaman 212).

1.1.4 Lihat



Tindakan *Lihat* akan memaparkan buku dalam program pelihat e-buku. calibre mempunyai pelihat terbina-dalam untuk kebanyakan format e-buku. Bagi lain-lain format ia menggunakan aplikasi sistem pengoperasian lalai. Anda boleh mengkonfigur format yang manakah harus dibuka dengan pelihat dalaman melalui *Keutamaan→Antaramuka→Kelakuan*. Jiika sesebuah buku mempunyai lebih dari satu format. Selain itu, anda boleh melihat format tertentu dengan mengklik-kanan pada butang tersebut.

1.1.5 Hantar ke peranti



Tindakan *Hantar ke peranti* mempunyai lapan variasi, yang boleh dicapai dengan mengklik-kanan pada butang tersebut.

1. **Hantar ke ingatan utama**: Buku terpilih dipindahkan ke dalam ingatan utama pembaca e-buku.
2. **Hantar ke kad (A)**: Buku terpilih dipindah ke kad storan (A) dalam pembaca e-buku.
3. **Hantar ke kad (B)**: Buku terpilih dipindah ke kad storan (B) dalam pembaca e-buku.
4. **Hantar format khusus ke**: Buku terpilih dipindahkan ke lokasi storan terpilih di dalam peranti, dalam format yang anda tentukan.
5. **Lenting peranti**: Tanggalkan peranti dengan calibre.
6. **Tetapkan tindakan hantar lalai ke peranti**: Membolehkan anda nyatakan pilihan-pilihan, 1 hingga 5 di atas atau 7 di bawah, akan menjadi tindakan lalai bila anda mengklik butang utama.
7. **Hantar dan padam dari pustaka**: Buku terpilih dipindahkan ke lokasi storan terpilih pada peranti dan kemudian **dipadamkan** dari Pustaka.
8. **Dapatkan Catatan (eksperimental)**: Pindah catatan yang anda buat pada e-buku dalam peranti anda ke data meta ulasan buku di dalam pustaka calibre.

Anda boleh mengawal nama fail dan folder struktur fail dihantar ke peranti dengan menetapkan templat di dalam *Keutamaan→Import/Eksport→Menghantar buku ke peranti*. Lihat juga *Bahasa templat calibre* (halaman 147).

1.1.6 Dapatkan berita

Tindakan *Dapatkan berita* akan memuat turun berita dari pelbagai sumber laman sesawang dan menukarnya menjadi e-buku yang boleh dibaca dengan pembaca e-buku anda. Kebiasaannya, e-buku yang baharu dicipta akan ditambah ke dalam pustaka e-buku anda, tetapi jika pembaca e-buku bersambung ketika muat turun selesai. berita juga akan dimuat naik ke dalam pembaca secara automatik.

Tindakan *Dapatkan berita* menggunakan resepi ringkas (10-15 baris kod) bagi setiap laman berita. Untuk mengetahui bagaimana hendak cipta resepi sumber berita anda sendiri, sila rujuk *Menambah laman sesawang berita kegemaran anda* (halaman 27).

Tindakan *Dapatkan berita* mempunyai tiga variasi, boleh dicapai dengan mengklik-kanan pada butang.

1. **Jadualkan muat turun berita**: Membolehkan anda jadualkan muat turun sumber berita terpilih anda dari senarai yang ada. Penjadualan boleh ditetapkan secara individu bagi setiap sumber berita yang anda pilih dan penjadualan yang anjal membolehkan anda pilih hari dalam seminggu atau kekerapan hari semasa memuat turun.
2. **Tambah sumber berita suai**: Membolehkan anda cipta satu resepi ringkas untuk memuat turun berita dari laman berita suai yang anda mahu capai. Penciptaan resepi adalah semudah dengan memasukkan URL suapan

berita RSS, atau anda boleh lebih teliti dengan mencipta kod berasaskan-Python bagi tugasan terperinci. Untuk maklumat lanjut sila rujuk *Menambah laman sesawang berita kegemaran anda* (halaman 27).

3. **Muat turun semua sumber berita berjadual**: Menyebabkan calibre serta-merta mula memuat turun semua sumber berita yang anda telah jadualkan.

1.1.7 Pustaka

Tindakan *Pustaka* membolehkan anda cipta, bertukar-tukar Pustaka, namakan semula atau membuang Pustaka. calibre membolehkan anda cipta sebanyak mana pustaka yang dikehendaki. Selain itu, anda boleh menghasilkan pustaka fiksyen, bukan-fiksyen, pustaka bahasa asing, projek perpustakaan, atau apa jua struktur yang memenuhi keperluan anda. Pustaka adalah struktur berorganisasi tertinggi di dalam calibre. Setiap pustaka mempunyai set buku, tag buku, kategori buku dan lokasi storan asasnya sendiri.

1. **Tukar/cipta pustaka...**: Membolehkan anda; a) sambung ke pustaka calibre pra-wujud pada lokasi lain, b) cipta satu pustaka kosong pada satu lokasi baharu, c) alih pustaka semasa ke lokasi baharu yang dinyatakan.
2. **Tukar pantas**: Membolehkan anda tukar diantara pustaka yang telah berdaftar atau dicipta di dalam calibre.
3. **Nama semula pustaka**: Membolehkan anda namakan semula Pustaka.
4. **Pick a random book**: Chooses a random book in the library for you
5. **Buang pustaka**: Membolehkan anda menyah-daftar pustaka dari calibre.
6. **Export/import all calibre data**: Allows you to either export calibre data for migration to a new computer or import previously exported data.
7. **<library name>**: Actions 7, 8 etc... give you immediate switch access between multiple libraries that you have created or attached to. This list contains only the 5 most frequently used libraries. For the complete list, use the Quick Switch menu.
8. **Penyelenggaraan pustaka**: Membolehkan anda periksa pustaka semasa jika ada masalah ketekalan data dan pulihkan pangkalan data pustaka semasa dari sandar yang ada.

Note: Data meta berkenaan e-buku anda, seperti tajuk, pengarang, dan tag tersimpan di dalam sebuah fail dalam folder pustaka calibre anda yang dikenali sebagai metadata.db. Jika fail ini menjadi rosak (sangat jarang berlaku), anda boleh kehilangan semua data meta. Sebagai langkah berjaga-jaga, calibre menyandar data meta secara automatik bagi setiap buku di dalam folder buku dalam format fail OPF. Dengan menggunakan tindakan Pulih pangkalan data di bawah Penyelenggaraan Pustaka, anda boleh membina semula fail metadata.db file melalui fail OPF tersebut.

Anda boleh salin atau alih buku diantara pustaka yang berlainan (jika anda mempunyai lebih dari satu pustaka) dengan mengklik-kanan pada buku dan pilih tindakan *Salin ke pustaka*.

1.1.8 Peranti

Tindakan *Peranti* membolehkan anda lihat buku di daam ingatan utama atau kad storan peranti anda atau lentingkan peranti tersebut (tanggalkannya dari calibre). Ikon ini ditunjukkan secara automatik dalam palang alat calibre utama ketika anda sambungkan peranti disokong. Anda boleh mengklik padanya untuk lihat senarai buku di dalam peranti anda. Anda juga boleh seret dan lepas buku dari pustaka calibre anda ke dalam ikon untuk memindahkannya ke dalam peranti anda. Sebaliknya, anda juga boleh seret dan lepas buku dari dalam peranti ke dalam ikon pustaka pada palang alat untuk memindahkan buku dari peranti ke pustaka calibre anda.

1.1.9 Simpan ke dalam cakera

Tindakan *Simpan ke cakera* mempunyai lima variasi, boleh dicapai dengan mengklik-kanan pada butang tersebut.

1. **Simpan ke cakera**: Simpan buku terpilih ke dalam cakera tersusun dalam direktori. Struktur direktori akan seperti berikut:

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

Anda boleh mengawal nama fail dan folder struktur fail tersimpan ke cakera dengan menetapkan templat di dalam *Keutamaan→Import/eksport→Menyimpan buku ke cakera*. Lihat juga *Bahasa templat calibre* (halaman 147).

2. **Simpan ke cakera dalam satu direktori**: Simpan buku terpilih ke cakera dalam satu direktori.

Bagi 1. dan 2., semua format tersedia, begitu juga sebagai data meta, disimpan ke dalam cakera bagi setiap buku terpilih. Data meta disimpan ke dalam fail OPF. Buku tersimpan boleh diimport semula ke dalam pustaka tanpa kehilangan maklumat dengan menggunakan tindakan *Tambah buku* (halaman 4).

3. **Hanya simpan format *<your preferred>* ke dalam cakera**: Simpan buku terpilih ke dalam cakera dalam struktur direktori yang ditunjukkan pada (1.) tetapi hanya dalam format e-buku yang anda kehendaki. Anda boleh tetapkan format dikehendaki dalam *Keutamaan→Antaramuka→Kelakuan→Format output dikehendaki*

4. **Hanya simpan format *<your preferred>* dalam satu direktori**: Simpan buku terpilih ke dalam cakera dalam satu direktori tetapi hanya dalam format e-buku yang anda kehendaki. Anda boleh tetapkan format yang dikehendaki dalam *Keutamaan→Antaramuka→Kelakuan→Format output dikehendaki*

5. **Simpan format tunggal ke cakera...**: Simpan buku terpilih ke struktur direktori sepertimana yang ditunjukkan pada (1.) tetapi hanya dalam format yang anda pilih dari senarai timbul.

1.1.10 Sambung/kongsi

Tindakan *Sambung/kongsi* membolehkan anda sambung secara manual ke peranti atau folder dalam komputer anda. Ia juga membolehkan anda pasang pustaka calibre anda untuk dicapai melalui pelayar sesawang atau emel.

> Tindakan *Sambung/kongsi* mempunyai empat variasi, boleh dicapai dengan mengklik-kanan pada butang tersebut.
>
> 1. **Sambung ke folder**: Membolehkan anda sambung pada mana-mana folder dalam komputer anda yang dianggap sebagai peranti dan dapat guna semua kemudahan calibre berikan untuk peranti dengan folder tersebut. Berguna jika peranti anda tidak disokong oleh calibre tetapi tersedia sebagai cakera USB.
> 2. **Mula pelayan Kandungan**: Mulakan pelayan sesawang terbina-dalam calibre. Bila dimulakan, pustaka calibre anda boleh dicapai melalui pelayar sesawang di Internet (jika anda mahu). Anda boleh konfigur bagaimana pelayan sesawang dicapai dengan menetapkan keutamaan di *Keutamaan→Perkongsian→Perkongsian melalui rangkaian*
> 3. **Persediaan emel berdasarkan perkongsian buku**: Membolehkan perkongsian buku dan suapan berita melalui emel. Selepas penetapan alamat emel untuk pilihan ini, calibre akan menghantar kemaskini berita dan kemaskini buku ke alamat emel tersebut. Anda boleh mengkonfigur bagaimana cara calibre menghantar emel melalui penetapan keutamaan di *Keutamaan→Perkongsian→Perkongsian buku melalui emel*. Seusai anda menetapkan satu atau lebih alamat emel, masukan menu ini akan diganti dengan masukan menu untuk menghantar buku ke alamat emel yang telah dikonfigurkan.

1.1.11 Buang buku

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

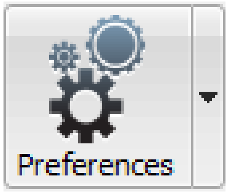
1. **Buang buku terpilih**: Membolehkan anda buang semua buku yang terpilih di dalam senarai buku secara **kekal**.
2. **Buang fail dalam format khusus dari buku terpilih...**: Membolehkan anda buang fail e-buku **secara kekal** dalam format khusus dari buku terpilih dalam senarai buku.
3. **Buang semua format dari buku terpilih, kecuali...**: Membolehkan anda buang fail ebuku **secara kekal** dari mana-mana format kecuali format tertentu buku yang terpilih dalam senarai buku.
4. **Buang semua format dari buku terpilih**: Membolehkan anda buang semua fail e-buku dari buku terpilih dalam senarai buku secara **kekal**. Hanya data meta akan dikekalkan.

5. **Buang kulit buku dari buku terpilih**: Membolehkan anda buang fail imej kulit buku dari buku terpilih di dalam senarai buku secara **kekal**.

6. **Buang buku sepadan dari peranti**: Membolehkan anda buang fail e-buku dari peranti tersambung yang sepadan dengan buku yang terpilih dalam senarai buku.

Note: Perhatian bila anda guna *Buang buku* untuk memadam buku dari pustaka calibre anda, rekod buku dipadam secara kekal, tetapi pada Windows dan OS X fail dialih ke dalam tong sampah. Ia membolehkan anda pulihkannya semula jika anda berubah fikiran.

1.2 Pilihan



Tindakan *Keutamaan* membolehkan anda ubah pelbagai cara calibre berfungsi. Ia mempunyai empat variasi, boleh dicapai dengan mengklik-kanan pada butang tersebut.

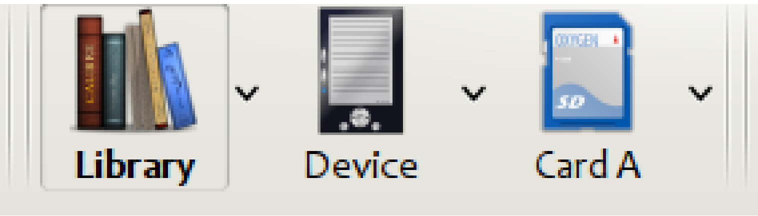
1. **Keutamaan**: Membolehkan anda ubah pelbagai aspek berkenaan calibre. Mengklik-kanan pada butang juga menjurus tindakan yang sama.

2. **Jalankan bestari aluan**: Membolehkan anda mulakan *Bestari aluan* yang muncul kali pertama anda memulakan calibre.

3. **Dapatkan pemalam untuk menambahbaik calibre**: Buka satu tetingkap yang menunjukkan pemalam untuk calibre. Pemalam ini dibangunkan oleh pihak ketiga yang dapat menambahbaik kefungsi calibre.

4. **Mula semula dalam mod nyahpepijat**: Membolehkan anda benarkan mod penyahpepijatan yang dapat membantu pembangun calibre menyelesaikan masalah yang anda hadapi ketika menggunakan program. Bagi kebanyakan pengguna ia kekal dilumpuhkan melainkan diarah oleh pembangun untuk hidupkannya.

1.3 Katalog



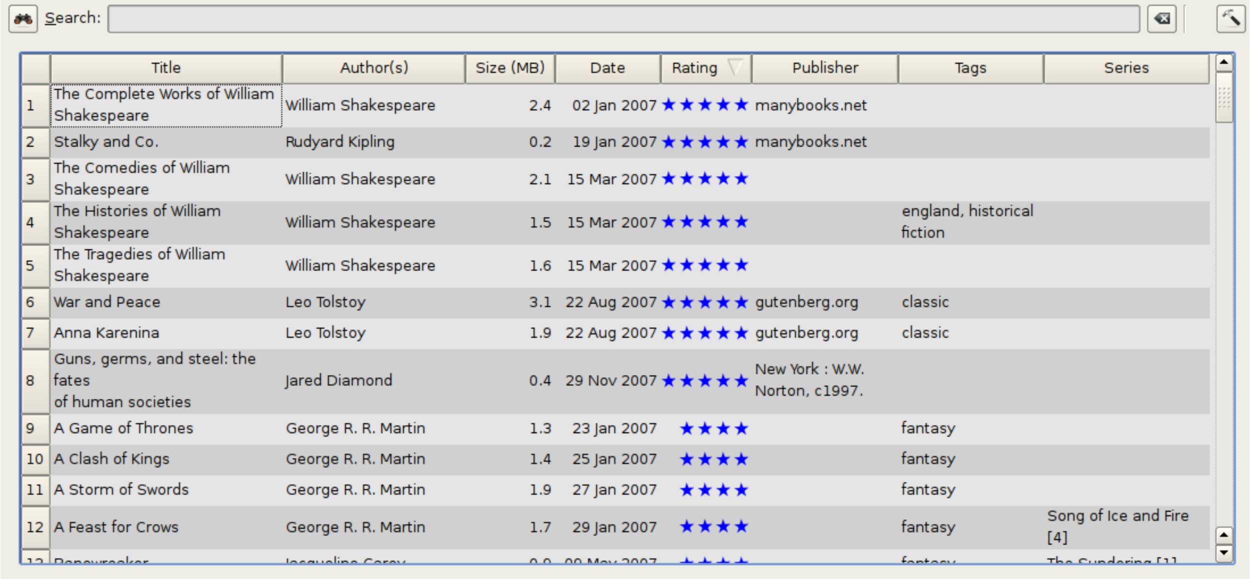
Katalog merupakan sebuah koleksi buku. calibre dapat mengurus dua jenis katalog yang berlainan:

1. **Pustaka**: Merupakan satu koleksi buku yang disimpan dalam pustaka calibre anda yang berada di dalam komputer anda.

2. **Peranti**: Merupakan satu koleksi buku yang tersimpan di dalam pembaca e-buku anda. Ia akan tersedia bila anda sambungkan pembaca dengan komputer anda.

Banyak operasi, seperti menambah, memadam, melihat buku dan lain-lain adalah sensitif konteks. Sebagai contoh, jika anda klik butang *Lihat* ketika anda mempunyai katalog **Peranti** terpilih, maka calibre akan membuka fail dalam peranti untuk melihatnya. Jika anda mempunyai katalog **Pustaka** terpilih, fail dalam pustaka calibre anda akan dibuka sebaliknya.

1.4 Gelintar & Isih



	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	Banewreaker	Jacqueline Carey	0.9	09 May 2007	★★★★		fantasy	The Sundering [1]

Seksyen Gelintar & Isih membolehkan anda lakukan beberapa tindakan pada koleksi buku anda.

- Anda boleh mengisihnya mengikut tajuk, pengarang, tarikh, penarafan, dll dengan mengklik pada tajuk lajur. Anda juga boleh sub-isih iaitu isih pada lajur berbilang. Sebagai contoh, jika anda klik pada tajuk lajur dan kemudian lajur pengarang, buku akan diisih mengikut pengarang dan kemudian semua masukan untuk pengarang yang sama akan diisih mengikut tajuk.
- Anda boleh gelintar buku tertentu atau set buku menggunakan palang Gelintar. Maklumat lanjut ada di bawah.
- Anda boleh sunting data meta secara pantas dan mudah dengan memilih masukan yang mahu diubah dalam senarai dan menekan kekunci `E`.
- Anda juga boleh lakukan *Tindakan* (halaman 4) pada set buku. Untuk memilih buku berbilang anda boleh:
  - Kekalkan kekunci `Ctrl` ditekan dan kemudian klik buku yang anda mahu pilih.
  - Kekalkan kekunci `Shift` ditekan dan klik pada buku mula dan akhir bagi sejulat buku yang anda mahu pilih.
- Anda boleh konfigur medan yang manakah mahu dipaparkan dengan menggunakan dialog konfigurasi.

1.5 Antaramuka gelintar

Anda boleh gelintar semua data meta dengan memasukkan terma gelintar dalam palang Gelintar. Penggelintaran adalah sensitif kata. Sebagai contoh:

```
Asimov Foundation format:lrf
```

Tindakan ini akan padankan semua buku dalam pustaka anda yang mempunyai perkataan `Asimov` dan `Foundation` dalam data meta mereka dan tersedia dalam format LRF. Lagi beberapa contoh:

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

Penggelintaran adalah secara lalai 'contains'. Satu item sepadan jika rentetan gelintar muncul di mana sahaja dalam data meta yang ditunjukkan. Dua jenis gelintar lain juga tersedia: gelintar sama dan gelintar menggunakan ungkapan nalar.

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". Regular expression searches are indicated by prefixing the search string with a tilde (~). Any Python-compatible regular expression can be used. Note that 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 `\\(`. Regular expression searches are 'contains' searches unless the expression contains anchors.

Patutkah anda perlu gelintar rentetan dengan awalan sama dengan atau tilde, awalkan rentetan dengan tanda miring belakang.

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

Anda boleh bina pertanyaan gelintar lanjutan menggunakan *Dialog gelintar Lanjutan* yang boleh dicapai dengan mengklik butang 🔍.

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.

Sintaks untuk menggelintar tarih ialah:

```
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, 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
```

Untuk menghindari masalah timbul disebabkan rentetan terjemahan bila menggunakan versi bukan-Inggeris calibre, rentetan `_today`, `_yesterday`, `_thismonth`, dan `_daysago` sentiasa tersedia. Mereka tidak boleh diterjemahkan.

Anda boleh gelintar buku yang mempunyai format bagi saiz tertentu seperti ini:

```
size:>1.1M Will find books with a format larger than 1.1MB
size:<=1K  Will find books with a format smaller than 1KB
```

Medan tarikh dan angka menyokong operator hubungan = (sama dengan), > (lebih besar dari), >= (lebih besar dari atau sama dengan), < (kurang dari), <= (kurang dari atau sama dengan), dan != (tidak sama dengan). Medan penarafan dianggap angka. Sebagai contoh, search `rating:>=3` akan cari semua buku yang mendapat penarafan 3 atau lebih tinggi.

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

Series indices 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.

Medan khas `gelintar` digunakan untuk gelintar tersimpan. Jadi jika anda simpan satu gelintar dengan nama "My spouse's books" anda boleh masukkan `search:"My spouse's books"` dalam palang gelintar yang gunakan semula gelintar tersimpan. Lagi perihal gelintar tersimpan di bawah.

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

Anda boleh gelintar ketidakhadiran atau kehadiran medan menggunakan nilai khas "true" dan "false". Sebagai contoh:

```
cover:false will give you all books without a cover
series:true will give you all books that belong to a series
comments:false will give you all books with an empty comment
format:false will give you all books with no actual files (empty records)
```

Yes/no custom columns are searchable. Searching 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.

Hierarchical items (e.g. A.B.C) use an extended syntax to match initial parts of the hierarchy. This is done by adding a period between the exact match indicator (=) and the text. For example, the query `tags:=.A` will find the tags *A* and *A.B*, but will not find the tags *AA* or *AA.B*. The query `tags:=.A.B` will find the tags *A.B* and *A.B.C*, but not the tag *A*.

Identifiers (e.g., ISBN, doi, lccn etc) also use an extended syntax. First, note that an identifier has the form `type:value`, as in `isbn:123456789`. The extended syntax permits you to specify independently which type and value to search for. Both the type and the value parts of the query can use *equality*, *contains*, or *regular expression* matches. Examples:

- `identifiers:true` akan cari buku dengan mana-mana pengecam.
- `identifiers:false` akan cari buku tanpa pengecam.

- `identifiers:123` akan gelintar buku dengan mana-mana jenis yang mengandungi *123*.
- `identifiers:=123456789` akan menggelintar buku apa jua jenis yang mempunyai nilai sama dengan *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` akan mencari buku dengan jenis yang mengandungi *i* daengan nilai *1*.

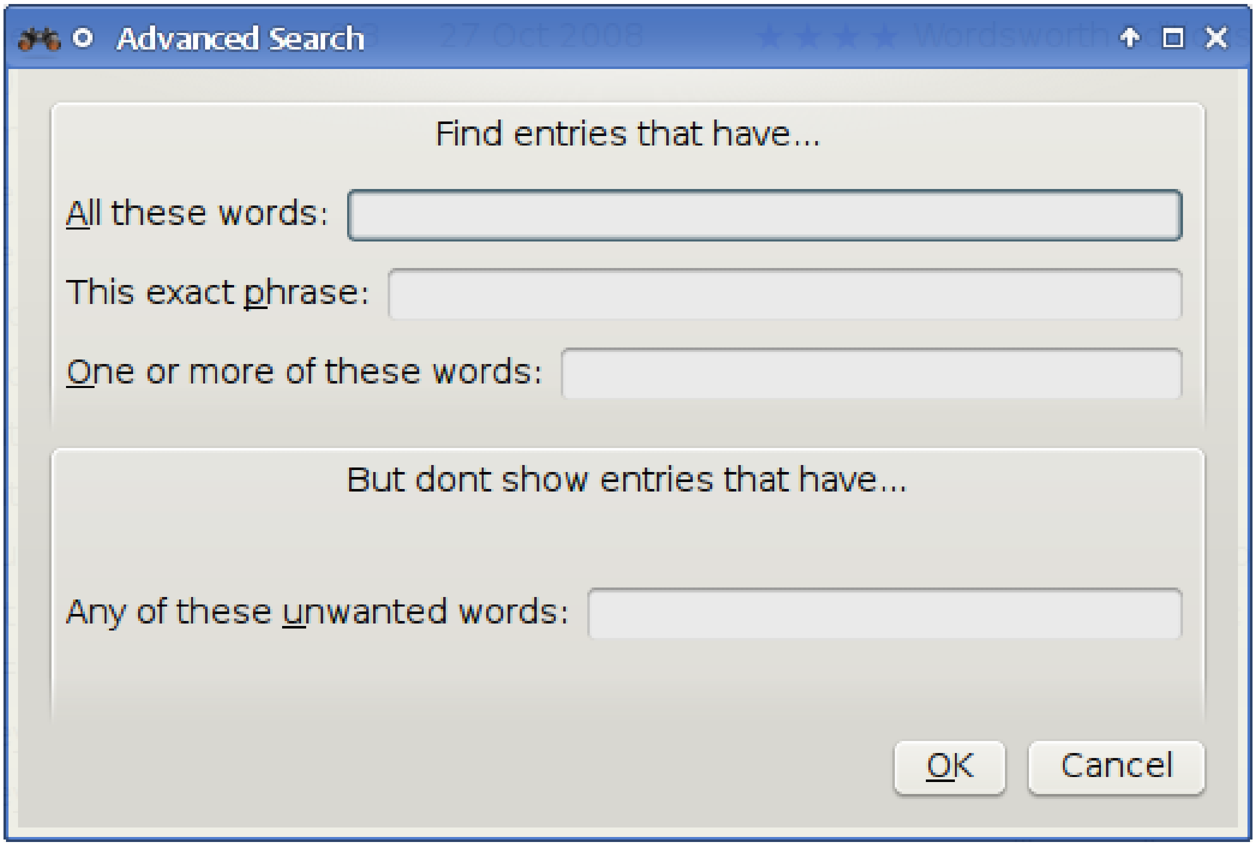

Advanced Search
Find entries that have...
All these words:
This exact phrase:
One or more of these words:
But dont show entries that have...
Any of these unwanted words:
OK
Cancel

Figure1: *Advanced search dialog*

You can search using a template in the *Bahasa templat calibre* (halaman 147) 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. It can use the special operators described above for the various search types. You must quote the entire search string if there are spaces anywhere in it.

Examples:

- **`template:"program: connected_device_name('main')#@#:t:kindle"` -- is true when the `kindle` device**

  - `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

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*. This can be added to the toolbars or menus via *Preferences→Toolbars & menus*. It can also be assigned a keyboard shortcut via *Preferences→Shortcuts*.

1.6 Menyimpan gelintar

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 Pustaka maya

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: *Pustaka maya* (halaman 217).

1.8 Meneka data meta dari nama fail

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

You can also control how metadata is read from the filename using regular expressions (see *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman **??**)). 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
```

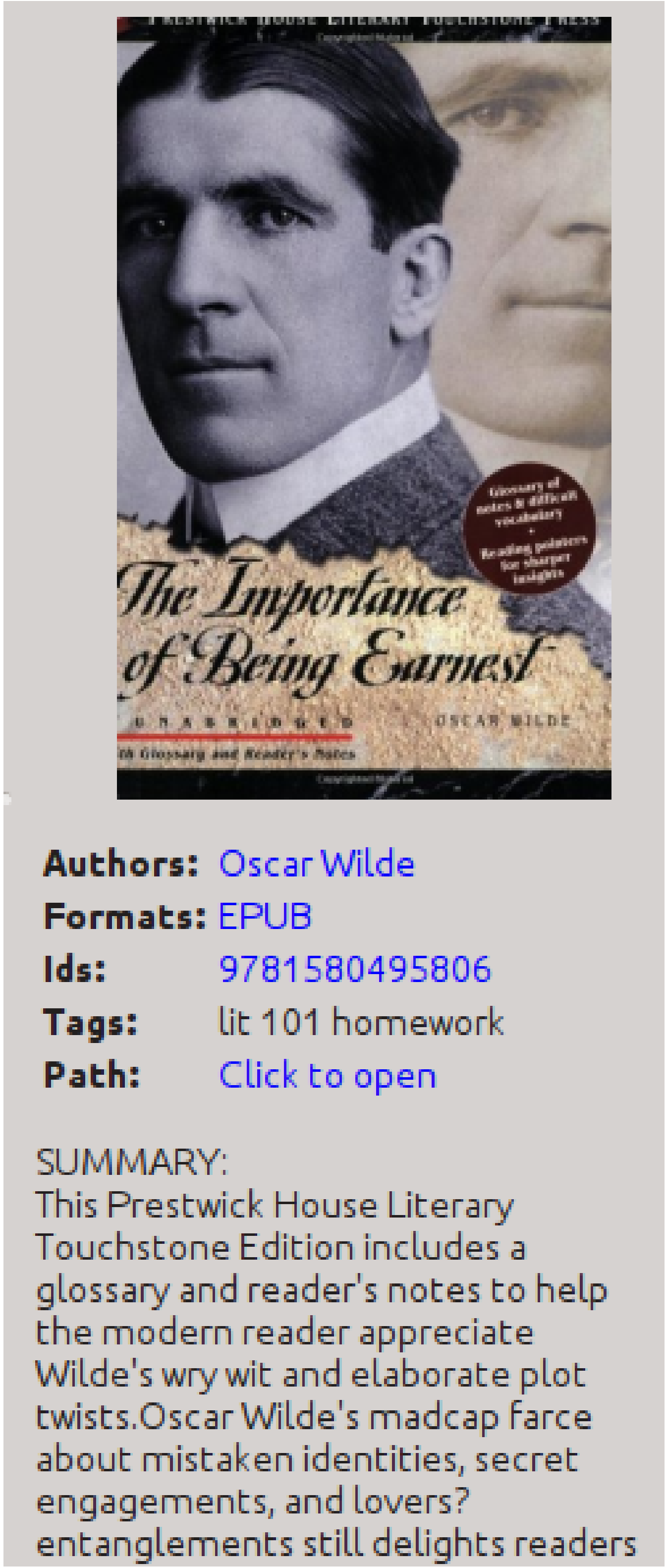
iaitu, ia menganggap semua aksara sebelum tanda − merupakan tajuk buku dan berikutnya adalah pengarang buku. Sebagai contoh, nama fail:

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

akan ditafsir untuk mempunyai tajuk: Foundation and Earth dan pengarang: Isaac Asimov

Tip: Jika nama fail tidak mengandungi tanda sengkang, ungkapan nalar di atas akan gagal berfungsi.

1.9 Perincian buku


The Importance of Being Earnest
UNABRIDGED
OSCAR WILDE
Glossary of notes & difficult vocabulary
Reading pointers for sharper insights

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.

Samalah, jika anda muat turun data meta buku tersebut, panel perincian Buku akan ditunjukkan secara automatik pautan yang berkenaan laman sesawang pada amazon, worlcat, dll yang mana data meta dimuat turunkan.

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.

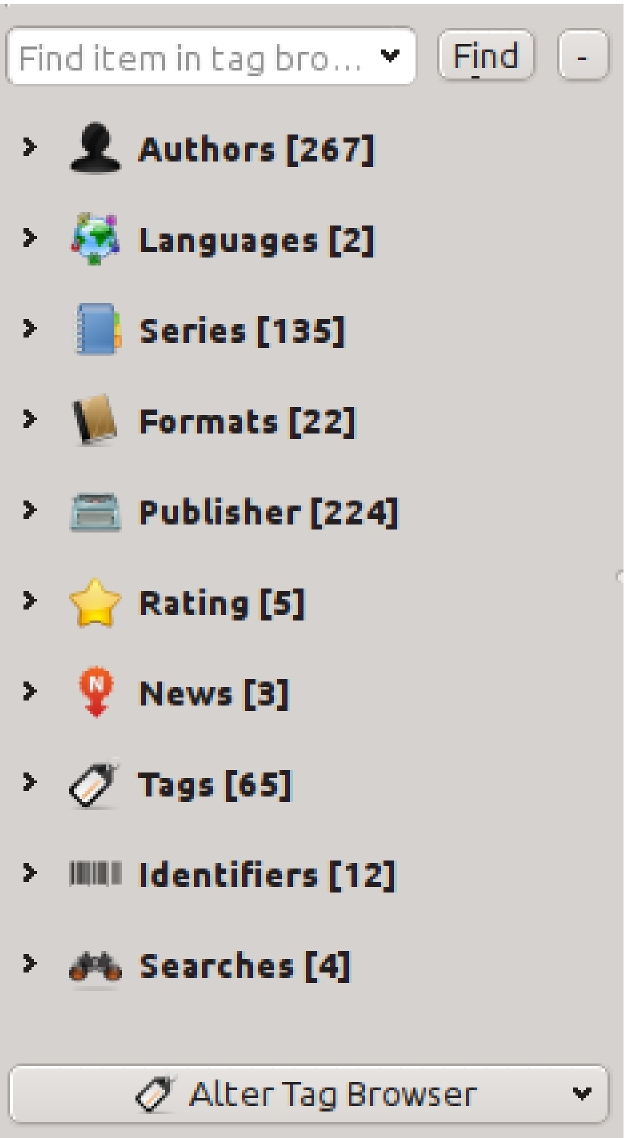
You can change the cover of the book by simply drag and dropping an image onto the Book details panel. If you wish to edit the cover image in an external program, simply right click on it and choose *Open with*.

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*.

1.10 Pelayar 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, series, 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 cyc

1. "segalanya sepadan dengan item dalam kategori" ditanda dengan tanda tambah berwarna hijau.
2. "segalanya sepadan dengan item dalam kategori atau sub-kategorinya" ditanda dengan tanda dua tambah berwarna hijau.
3. "segalanya tidak sepadan dengan item dalam kategori" ditanda dengan tanda tolak berwarna merah.
4. "segalanya tidak sepadan dengan item dalam kategori atau sub-kategorinya" ditanda dengan tanda dua tolak berwarna merah.

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"* (halaman 139) for more information.

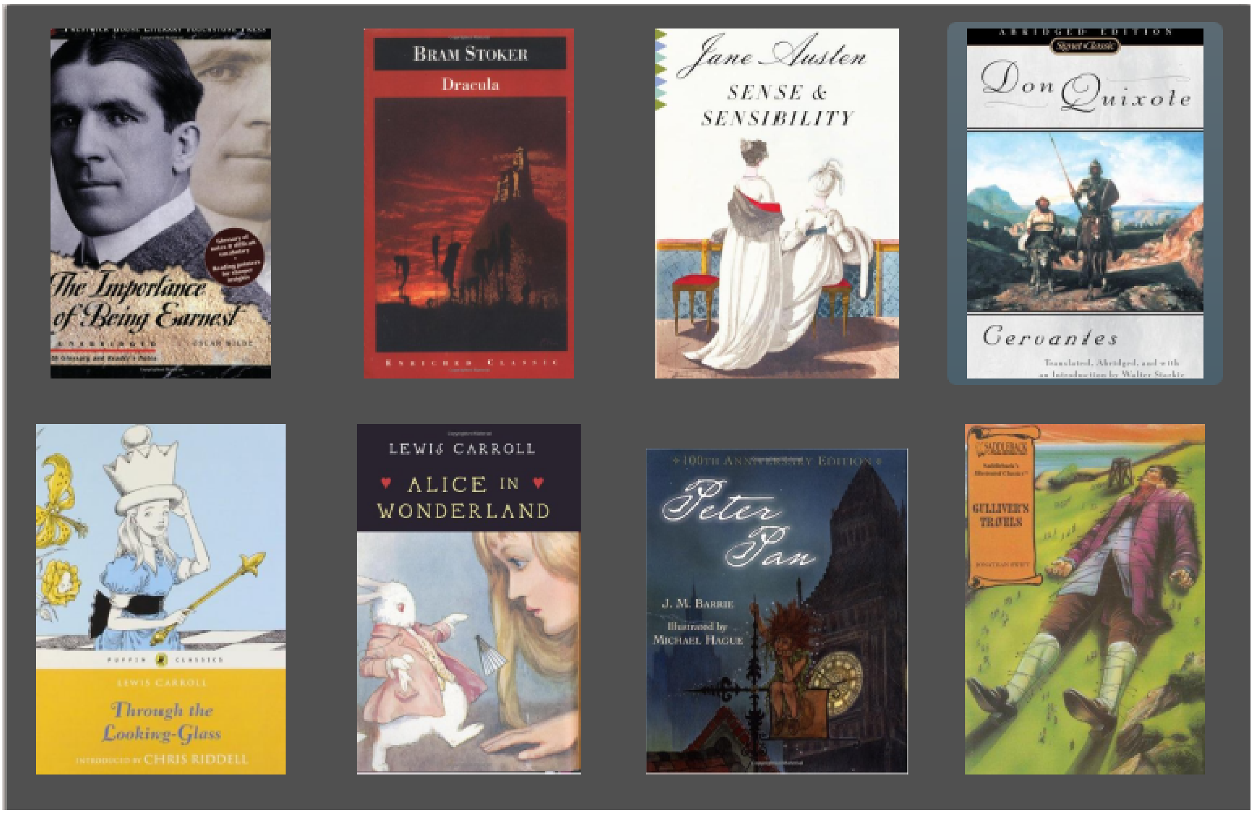
Hierarchical items (items with children) use the same four 'click-on' searches as User categories. Items that do not have children use two of the searches: "everything matching" and "everything not matching".

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.

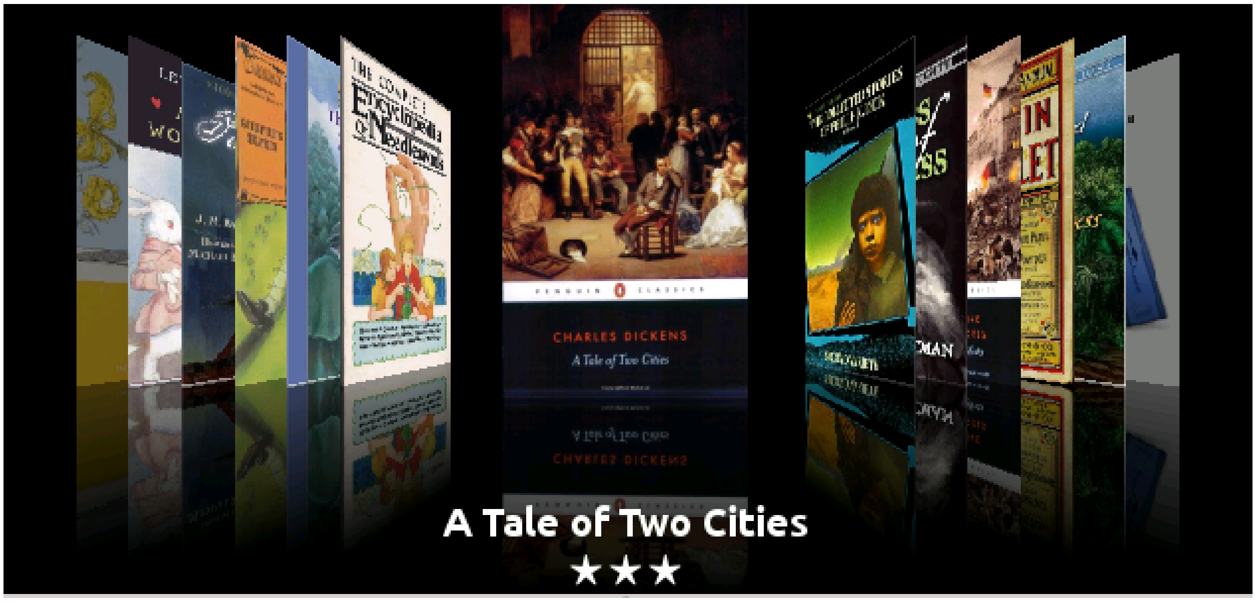
You can control how items are sorted in the Tag browser via the *Configure* button at the lower-left of the Tag browser. You can choose to sort by name, average rating or popularity (popularity is the number of books with an item in your library; for example, the popularity of Isaac Asimov is the number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

1.11 Grid kulit buku



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.

1.12 Pelayar kulit buku



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.

1.13 Paparan Pantas

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.

Beberapa contoh penggunaan Paparanpantas: lihat secara pantas lain-lain buku:

- have some tag(s) applied to the currently selected book,
- adalah berada dalam siri yang sama dengan buku semasa
- mempunyai nilai yang sama dalam lajur suai seperti buku semasa
- adalah ditulis oleh salah seorang pengarang yang sama dengan buku semasa
- kongsi nilai dalam satu lajur suai

Terdapat dua cara supaya maklumat Paparan Pantas muncul:

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.

Dwi-klik pada buku dalam tetingkap paparan Pantas untuk memilih buku tersebut di dalam paparan pustaka. Tindakan ini juga mengubah item yang dipapar dalam tetingkap Paparan Pantas (anak tetingkap sebelah kiri) yang dapat menunjukkan item buku yang baru dipilih.

`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.

Anda dapati satu lajur boleh dipapar pantas dengan letak kursor tetikus di atas tajuk lajur dan melihatnya di bahagian petua alat bagi tajuk tersebut. Anda juga boleh mengklik-kanan pada tajuk lajur untuk melihat pilihan "Paparan Pantas" ditunjukkan pada menu, yang mana memilih pilihan Paparan Pantas adalah sama dengan mengetik 'Q' dalam sel semasa.

Pilihan (dalam *Keutamaan→Penampilan→Paparan Pantas*):

- Respect (or not) the current Virtual library. If checked then Quickview shows only books in the current Virtual library. Default: respect Virtual libraries
- Ubah kandungan Paparan Pantas ketika lajur berubah di dalam senarai buku menggunakan kekunci kursor. Lalai: jangan ikuti perubahan yang dibuat oleh kekunci kursor
- Change the column being "quickview'ed" when a cell in the Quickview window is double-clicked. Otherwise the book is changed but the column being examined is not. Default: change the column
- Change the column being "quickview'ed" to the current column when the return key is pressed in the Quickview panel. Otherwise the book is changed but the column being examined is not. Default: change the column
- Choose which columns are shown in the Quickview window/panel.

1.14 Kerja

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.

1.15 Pintasan papan kekunci

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.

Note: Perhatian: Pintasan papan kekunci calibre tidak memerlukan kekunci pengubahsuai (Perintah, Pilihan, Kawalan, dll.), melainkan telah ditentukan. Anda hanya perlu tekan kekunci abjad, contohnya E untuk menyunting.

Table1: Pintasan papan kekunci untuk program calibre utama

Pintasan papan kekunci	Tindakan
`F2 (Enter dalam macOS)`	Sunting data meta medan terpilih semasa di dalam senarai buku.
`A`	Tambah buku
`Shift+A`	Add formats to the selected books
`C`	Convert selected books
`D`	Hantar ke peranti
`Del`	Buang buku terpilih
`E`	Sunting data meta buku terpilih
`G`	Dapatkan buku
`I`	Tunjuk perincian Buku
`K`	Sunting Senarai Kandungan
`M`	Gabung rekod terpilih
`Alt+M`	Gabung rekod terpilih, tetapi rekod asal masih kekal
`O`	Buka folder yang dikandungi
`P`	Gilap buku
`S`	Simpan ke dalam cakera
`T`	Sunting buku
`V`	Lihat
`Shift+V`	View last read book
`Alt+V/Cmd+V dalam macOS`	Lihat format tertentu
`Alt+Shift+J`	Togol senarai kerja
`Alt+Shift+B`	Togol pelayar Kulit Buku
`Alt+Shift+D`	Togol panel perincian Buku
`Alt+Shift+T`	Togol pelayar Tag
`Alt+Shift+G`	Togol grid Kulit Buku
`Alt+A`	Tunjuk buku mengikut pengarang yang sama sebagai buku semasa
`Alt+T`	Tunjuk buku mengikut tag yang sama sebagai buku semasa
`Alt+P`	Tunjuk buku mengikut penerbit yang sama sebagai buku semasa
`Alt+Shift+S`	Tunjuk buku mengikut siri yang sama sebagai buku semasa
`/, Ctrl+F`	Fokuskan palang Gelintar
`Shift+Ctrl+F`	Buka dialog gelintar lanjutan
`Shift+Alt+F`	Toggle the Search bar
`Esc`	Kosongkan gelintar semasa
`Shift+Esc`	Fokus senarai buku
`Ctrl+Esc`	Clear the Virtual library
`Alt+Esc`	Kosongkan sekatan tambahan

continues on next page

Table 1 – continued from previous page

Pintasan papan kekunci	Tindakan
`Ctrl+*`	Create a temporary Virtual library based on the current search
`Ctrl+Right`	Select the next Virtual library tab
`Ctrl+Left`	Select the previous Virtual library tab
`N` atau `F3`	Cari buku berikutnya yang sepadan dengan gelintar semasa (hanya berfungsi jika penyorotan dihidupkan dalam keutamaan gelintar)
`Shift+N` atau `Shift+F3`	Cari buku terdahulu yang sepadan dengan gelintar semasa (hanya berfungsi jika penyorotan dihidupkan dalam keutamaan gelintar)
`Ctrl+D`	Download metadata and covers
`Ctrl+R`	Restart calibre
`Ctrl+Shift+R`	Mula semula calibre dalam mod nyahpepijat
`Shift+Ctrl+E`	Tambah buku kosong ke pustaka
`Ctrl+M`	Togol status tanda/tidak bertanda untuk buku terpilih
`Q`	Buka dialog timbul Paparan Pantas untuk melihat buku dalam siri/tag/lain-lain yang berkaitan.
`Shift+Q`	Fokuskan panel Paparan Pantas yang terbuka
`Shift+S`	Buat gelintar dalam panel Paparan Pantas
`F5`	Re-apply the current sort
`Ctrl+Q`	Keluar dari calibre

CHAPTER 2

Menambah laman sesawang berita kegemaran anda

calibre mempunyai bingkai kerja yang hebat, fleksibel dan mudah digunakan untuk memuat turun berita dari Internet dan menukar ia menjadi e-buku. Yang berikut akan menunjukkan kepada anda, dengan contoh bagaimana hendak dapatkan berita dari pelbagai laman sesawang.

Untuk memahami bagaimana hendak guna bingkai kerja, ikuti contoh yang tersenarai tertib di bawah:

- *Pemerolehan automatik sepenuhnya* (halaman 28)
  - *Blog calibre* (halaman 28)
  - *bbc.co.uk* (halaman 30)
- *Penyuaian proses mendapatkan artikel* (halaman 30)
  - *Menggunakan versi cetak bbc.co.uk* (halaman 30)
  - *Menggantikan gaya artikel* (halaman 31)
  - *Menghiris dan mencincang* (halaman 32)
  - *Contoh sebenar* (halaman 32)
- *Petua untuk membangunkan resepi baharu* (halaman 35)
- *Bacaan lanjutan* (halaman 36)
- *Dokumentasi API* (halaman 36)

2.1 Pemerolehan automatik sepenuhnya

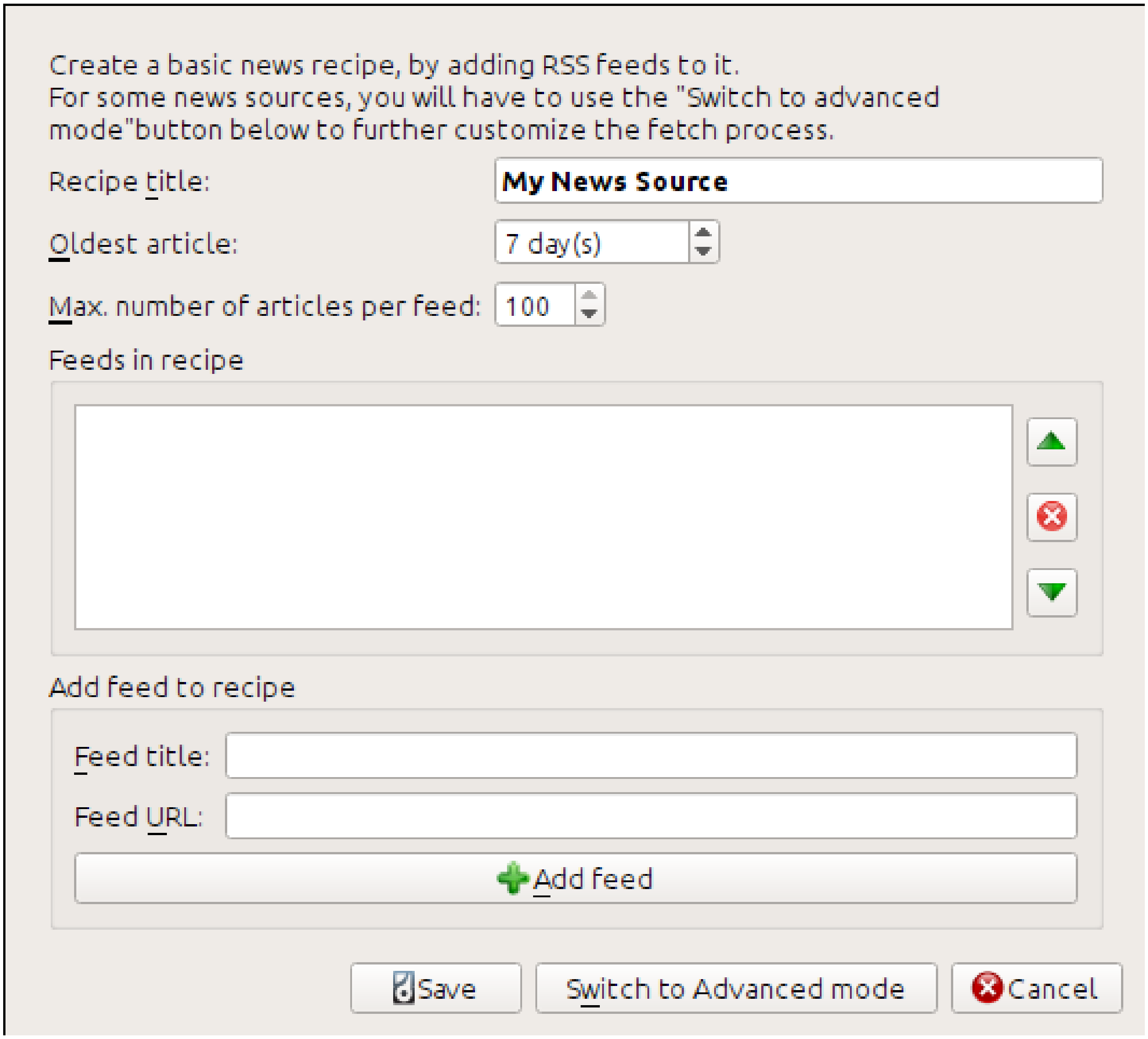
Jika sumber berita anda adalah ringkas, calibre boleh mendapatkannya dengan sepenuhnya secara automatik, apa yang anda perlu sediakan adalah URL. calibre menghimpun semua maklumat yang diperlukan untuk memuat turun sumber berita dalam bentuk *resepi*. Untuk memberitahu calibre berkenaan sumber beritah, anda perlu cipta *resepi* untuknya. Ok, mari kita lihat beberapa contoh yang disediakan:

2.1.1 Blog calibre

Blog calibre adalah blog pos yang menjelaskan banyak fitur calibre yang berguna dalam cara yang mudah dan boleh dicapai untuk pengguna baru calibre. Untuk memuat turun blog ini menjadi e-buku, kami bergantung pada suapan *RSS* blog:

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

Saya perolehi URL RSS dengan meneliti "Subscribe to" di bahagian bawah laman blog dan pilih *Posts→Atom*. Supaya calibre dapat memuat turun suapan dan menukarkanya menjadi e-buku, anda boleh klik-kanan butang *Dapatkan berita* kemudian item menu *Tambah sumber berita suai* dan kemudian butang *Resepi Baharu*. Satu dialog serupa dengan yang telah ditunjukkan akan dibuka.



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: My News Source

Oldest article: 7 day(s)

Max. number of articles per feed: 100

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.

Dua medan berikutnya (*Oldest article* dan *Max. number of articles*) membolehkan anda kawal berapa banyak artikel yang patut dimuat turun dari setiap suapan.

Untuk menambah suapan ke dalam resepi, masukkan tajuk suapan dan URL suapan kemudian klik butang *Tambah suapan*. Selepas itu, hanya klik butang *Simpan* dan kerja anda telah selesai! Tutup dialog.

Untuk menguji *resepi* baharu anda, klik butang *Dapatkan berita* dan pada sub-menu *Suai sumber baharu* klik *calibre Blog*. Selepas beberapa minit, e-buku baharu dimuat turun akan muncul dalam paparan pustaka utama (jika anda memiliki pembaca yang telah bersambung, ia akan dimasukkan ke dalam pembaca selain dari ke dalam pustaka). Pilih e-buku tersebut dan tekan butang :guilabel:`Lihat' untuk membaca!

Sebab apa proses ini menghasilkan e-buku yang baik, kerana blog ini menyediakan suapan *RSS* dengan *kandungan-lengkap* iaitu, kandungan artikel sebenar terbenam dalam suapan itu sendiri. Bagi kebanyakan sumber berita yang menyediakan berita dengan suapan dengan *kandungan-lengkap*, anda tidak perlu buat lebih kerja untuk menukar ia menjadi e-buku. Sekarang kita lihat sumber berita yang tidak menyediakan suapan dengan kandungan-lengkap. Suapan sebegini, artikel lengkap berada dalam laman sesawang dan suapan hanya menyediakan pautan ke laman sesawang dengan ringkasan pendek artikel berkenaan.

2.1.2 bbc.co.uk

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

1. Laman Muka Hadapan Berita: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_page/rss.xml
2. Sains/Alam: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/science/nature/rss.xml

Ikuti prosedur yang digariskan dalam *Blog calibre* (halaman 28) seperti di atas untuk menghasilkan resepi untuk *The BBC* (menggunakan suapan tersebut). Berdasarkan pada e-buku yang dimuat turun, didapati calibre telah melakukan kerja yang bagus hanya mengekstrak kandungan yang dikehendaki bagi setiap artikel dalam laman sesawang. Walaubagaimanapun, proses pengekstrakan ini tidaklah seratus peratus sempurna. Kadangkala ia meninggalkan kandungan yang tidak dikehendaki seperti menu dan bantuan navigasi atau ia telah buang kandungan yang sepatutnya diambil seperti tajuk artikel. Untuk memastikan pengekstrakan kandungan menepati kehendak anda, kita perlu suaikan proses mendapatkan artikel, yang dijelaskan pada seksyen yang berikutnya.

2.2 Penyuaian proses mendapatkan artikel

Bila anda mahukan proses muat turun yang sempurna, atau muat turun kandungan dari laman sesawang yang kompleks, anda perlu biasakan dengan bingkai kerja *resepi* yang fleksibel. Untuk membuatnya, pada dialog *Tambah sumber berita suai*, hanya klik butang *Tukar ke mod Lanjutan*.

Penyuaian termudah dan paling produktif ialah dengan menggunakan versi cetak artikel atas-talian. Versi cetak biasanya kurang kompleks dan mudah diterjemah menjadi e-buku. Cuba guna versi cetak daripada *The BBC*.

2.2.1 Menggunakan versi cetak bbc.co.uk

Langkah pertama adalah melihat e-buku yang telah dimuat turun sebelum ini daripada *bbc.co.uk* (halaman 30). Dipenghujung setiap artikel, dalam e-buku terdapat lokasi artikel dimuat turun. Salin dan tampal URL tersebut ke dalam pelayar. Sekarang laman sesawang artikel akan menuju ke "Versi Boleh Cetak". Klik padanya untuk melihat versi cetak artikel. Ia kelihatan lebih kemas! Sekarang bandingkan kedua-dua URL yang mana:

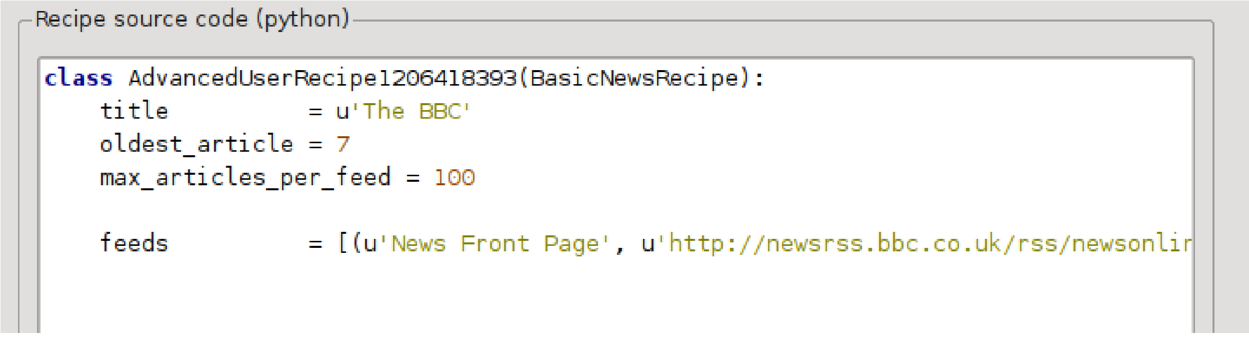
URL Artikel https://news.bbc.co.uk/2/hi/science/nature/7312016.stm

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

Jika mahukan versi cetak, kita perlu tambah setiap URL artikel dengan:

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):



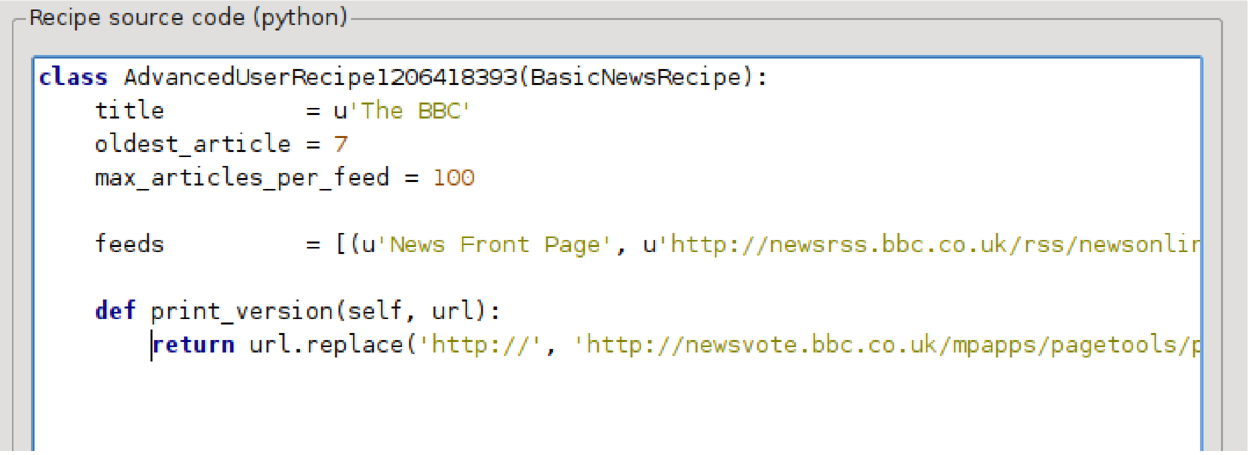
```
class AdvancedUserRecipe1206418393(BasicNewsRecipe):
    title          = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100

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

Anda dapati medan dari *Mod asas* telah diterjemah dalam kod bahasa Python. Kita perlu tambah arahan ke dalam resepi ini supaya dapat guna versi cetak artikel. Apa yang diperlukan adalah dengan menambah dua baris berikut:

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

Ini adalah bahasa Python, oleh itu penggunaan indentasi adalah penting. Selepas anda menambah baris, ia sepatutnya kelihatan seperti berikut:



Recipe source code (python)

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

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

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

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

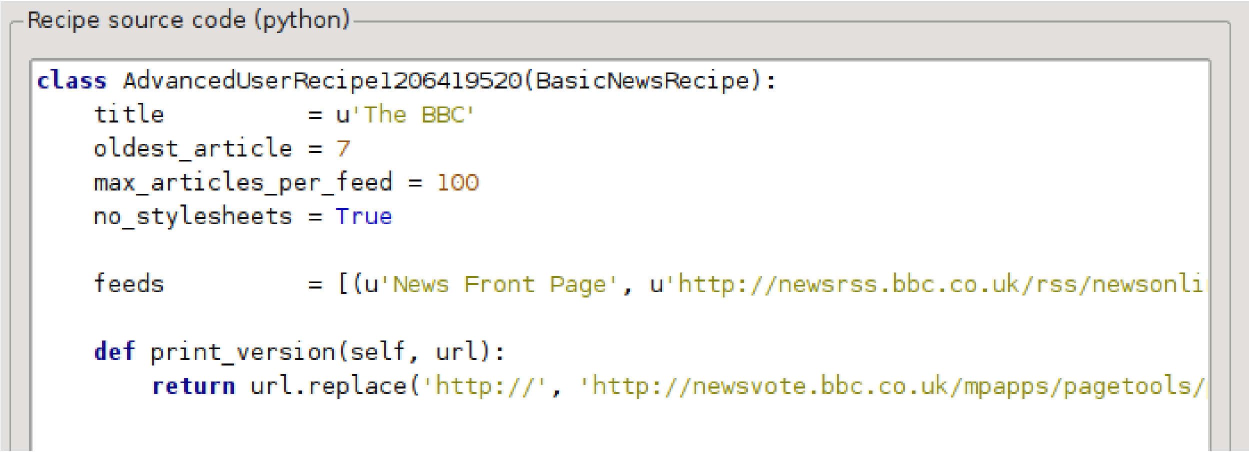
Sekarang, klik butang *Tambah/kemaskini resepi* dan perubahan yang anda buat akan disimpan. Muat semula e-buku. E-buku tersebut telah diperkemaskan. Salah satu masalah dengan versi baharu ialah fon yang digunakan pada versi cetak terlalu kecil. Ia dibaiki secara automatik bila menukar menjadi e-buku, tetapi selepas proses pembaikan, saiz fon bagi menu dan palang navigasi menjadi terlalu besar berbanding teks artikel. Untuk membaikinya, kita perlu lakukan lagi penyuaian, pada seksyen berikutnya.

2.2.2 Menggantikan gaya artikel

Dalam seksyen terdahulu, kita dapati saiz fon artikel dalam versi cetak *The BBC* terlalu kecil. Dalam kebanyakan laman sesawang, *The BBC* juga, saiz fon ini ditetapkan melalui lembaran gaya *CSS*. Kita boleh lumpuhkan pengambilan lembaran gaya tersebut dengan menambah baris:

```
no_stylesheets = True
```

Resepi kini kelihatan seperti:



```
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/
```

Versi baharu kelihatan menarik. Jika anda mementingkan kesempurnaan, anda perlu baca seksyen berikutnya yang melibatkan pengubahsuaian sebenar kandungan muat turun.

2.2.3 Menghiris dan mencincang

calibre memiliki keupayaan yang hebat dan fleksibel bila memanipulasikan kandungan muat turun. Sebagai contoh, kita rujuk kembali *The BBC* (halaman 31) sekali lagi. Berdasarkan pada kod sumber (*HTML*) beberapa artikel (versi cetak), didapati mereka mempunyai pengkaki yang tidak diperlukan, terkandung didalamnya

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

Ia boleh dibuang dengan menambah:

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

ke dalam resepi. Akhir sekali, gantikan beberapa *CSS* yang telah dilumpuhkan sebelum ini, dengan *CSS* anda sendiri yang sesuai dengan penukaran sesebuah e-buku:

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

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

Resepi ini hanya menjelajah sebahagian kecil sahaja bila dibandingkan kelebihan yang ada pada calibre. Untuk menjelajah lebih mendalam keupayaan calibre kita perlu menyelidiki contoh sebenar yang lebih kompleks pada seksyen berikutnya.

2.2.4 Contoh sebenar

Contoh sebenar yang benar-benar kompleks yang mendedahkan lebih *API* `BasicNewsRecipe` adalah *resepi* untuk *The New York Times*

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

class NYTimes(BasicNewsRecipe):
```

(continues on next page)

(continued from previous page)

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

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

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

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

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

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

(continues on next page)

(continued from previous page)

```
                 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'))
```

Kami melihat beberapa fitur baharu dalam *resepi* ini. Mula-mula, kita perlu:

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

Ini tetapkan masa paparan pada muka hadapan e-buku yang dihasilkan dalam format, `Day, Day_Number Month, Year`. Sila rujuk *`timefmt`* (halaman 44).

Kemudian kita lihat sekumpulan arahan untuk membersihkan *HTML* dimuat turun:

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

Ia membuang segalanya sebelum `<h1>` tag pertama dan segalanya selepas tag pertama yang mana id ialah `footer`. Sila rujuk *`remove_tags`* (halaman 43), *`remove_tags_before`* (halaman 43), *`remove_tags_after`* (halaman 43).

Fitur menarik berikutnya ialah:

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

`needs_subscription = True` memberitahui calibre bahawa resepi ini perlukan nama pengguna dan kata laluan supaya ia dapat capai kandungan tersebut. Oleh itu, calibre meminta nama pengguna dan kata laluan bila anda cuba guna resepi ini. Kod di dalam *`calibre.web.feeds.news.BasicNewsRecipe.get_browser()`* (halaman 37) sebenarnya mendaftar masuk ke laman sesawang the NYT. Seusai mendaftar masuk, calibre akan menggunakannya semula untuk dapatkan semua kandungan. Sila rujuk mechanize untuk memahami kod di dalam `get_browser`.

The next new feature is the *`calibre.web.feeds.news.BasicNewsRecipe.parse_index()`* (halaman 38) method. Its job is to go to https://www.nytimes.com/pages/todayspaper/index.html and fetch the list of articles that appear in *todays* paper. While more complex than simply using *RSS*, the recipe creates an e-book that corresponds very closely to the days paper. `parse_index` makes heavy use of BeautifulSoup to parse the daily paper webpage. You can also use other, more modern parsers if you dislike BeautifulSoup. calibre comes with lxml and html5lib, 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)
```

Fitur baharu terakhir ialah kaedah *`calibre.web.feeds.news.BasicNewsRecipe.preprocess_html()`* (halaman 39). Ia boleh digunakan untuk melakukan transformasi arbitari pada setiap halaman HTML dimuat turun. Di sini ia digunakan untuk melangkaui iklan yang mana nytimes paparkan sebelum setiap artikel.

2.3 Petua untuk membangunkan resepi baharu

Cara terbaik membangunkan resepi baharu adalah dengan penggunaan antaramuka baris perintah. Cipta resepi mengguna penyunting Python kegemaran anda dan simpan ia ke dalam fail seperti `resepisaya.recipe`. Sambungan *.recipe* diperlukan. Kemudian anda boleh muat turun kandungan menggunakan resepi dengan perintah:

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

Perintah **`ebook-convert`** akan muat turun semua laman sesawang dan simpan ia ke dalam fail EPUB `resepisaya.epub`. Pilihan `-vv` menjadikan ebook-convert buang banyak maklumat berkenaan kerjanya. Pilihan `--test` menyuruh ia muat turun hanya beberapa pasangan artikel paling banyak dari dua suapan. Selain itu, ebook-convert akan masukkan HTML dimuat turun ke direktori `debug/input`, iaitu `debug` adalah direktori yang anda nyatakan dalam pilihan *`ebook-convert --debug-pipeline`* (halaman 296).

Selepas muat turun selesai, anda boleh lihat *HTML* dimuat turun dengan membuka fail `debug/input/index.html` di dalam pelayar. Jika anda puashati dengan muat turun dan pra-pemprosesan yang berlaku sepertimana yang diharapkan, anda boleh jana e-buku dalam format berlainan seperti di bawah:

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

Jika anda puashati dengan resepi anda, dan anda dapati masih terdapat ruang untuk ruang penambahbaikan dalam set resepi terbina-dalam, poskan resepi anda ke forum resepi calibre recipes untuk berkongsi ia dengan lain-lain pengguna calibre.

Note: Pada MacOS, alat baris perintah disediakan dalam berkas calibre, contohnya, jika anda memasang calibre di dalam `/Applications` alat baris perintah berada di dalam `/Applications/calibre.app/Contents/MacOS/`.

See also:

***ebook-convert* (halaman ??)** Antaramuka baris perintah untuk semua pertukaran format e-buku.

2.4 Bacaan lanjutan

Untuk mengetahui lebih lanjut berkenaan penulisan resepi lanjutan dengan menggunakan beberapa fasiliti, yang tersedia dalam `BasicNewsRecipe` anda perlu ke beberapa sumber berikut:

***Dokumentasi API* (halaman 36)** Dokumentasi kelas `BasicNewsRecipe` dan semua kaedah dan medan pentingnya.

BasicNewsRecipe Kod sumber `BasicNewsRecipe`

Resepi terbina-dalam Kod sumber untuk resepi terbina-dalam yang disertakan dengan calibre

Forum resepi calibre Lebih banyak penulis resepi calibre yang berpengetahuan datang ke sini.

2.5 Dokumentasi API

2.5.1 API documentation for recipes

API untuk menulis resepi ditakrif oleh *`BasicNewsRecipe`* (halaman 36)

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 *Menambah laman sesawang berita kegemaran anda* (halaman **??**).

abort_article(*msg=None*)

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

abort_recipe_processing(*msg*)

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

add_toc_thumbnail(*article*, *src*)

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

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()`* (halaman 39).

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.

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

cleanup()

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

clone_browser(*br*)

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

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

default_cover (*cover_file*)
Create a generic cover for recipes that don't have a cover

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

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

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

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

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

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

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

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

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

get_masthead_title ()
Override in subclass to use something other than the recipe title

get_masthead_url ()
Return a *URL* to the masthead image for this issue or *None*. By default it returns the value of the member

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

get_obfuscated_article(*url*)

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

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

classmethod image_url_processor(*baseurl*, *url*)

Perform some processing on image urls (perhaps removing size restrictions for dynamically generated images, etc.) and return the precessed URL.

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 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.

Parameters

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

parse_feeds()

Create a list of articles from the list of feeds returned by `BasicNewsRecipe.get_feeds()` (halaman 37). 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()` (halaman 38).

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()` (halaman 36).

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).

Parameters

- **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.

Parameters

- **oeb** -- An OEBBook object
- **opts** -- Conversion options

postprocess_html(*soup*, *first_fetch*)

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

Parameters

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

preprocess_html(*soup*)

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

soup: A BeautifulSoup 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:

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

skip_ad_pages(*soup*)

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

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

sort_index_by(*index*, *weights*)

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

index: A list of titles.

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

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

Convenience method to take a BeautifulSoup *Tag* and extract the text from it recursively, including any CDATA sections and alt tag attributes. Return a possibly empty unicode string.

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

tag: BeautifulSoup *Tag*

articles_are_obfuscated = False

Set to True and implement *get_obfuscated_article()* (halaman 38) to handle websites that try to make it difficult to scrape content.

auto_cleanup = False

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

auto_cleanup_keep = None

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

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

center_navbar = True

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

compress_news_images = False

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

compress_news_images_auto_size = 16

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

compress_news_images_max_size = None

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

conversion_options = {}

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

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

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

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

delay = 0

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

description = ''

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

encoding = None

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

extra_css = None

Specify any extra *CSS* that should be added to downloaded *HTML* files. It will be inserted into *<style>* tags, just before the closing *</head>* tag thereby overriding all *CSS* except that which is declared using the style attribute on individual *HTML* tags. Note that if you want to programmatically generate the extra_css override the *get_extra_css()* (halaman 37) method instead. For example:

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

feeds = None

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

filter_regexps = []

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

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

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

Only one of *BasicNewsRecipe.match_regexps* (halaman 42) or *BasicNewsRecipe.filter_regexps* (halaman 41) should be defined.

handle_gzip = False

Set to True if you want to use gziped transfers. Note that some old servers flake out with this so it is off by default.

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* (halaman 43). 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* (halaman 42) or *BasicNewsRecipe.filter_regexps* (halaman 41) 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* (halaman 42)

needs_subscription = False

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

no_stylesheets = False

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

oldest_article = 7.0

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

preprocess_regexps = []

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

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

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

publication_type = 'unknown'

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

recipe_disabled = None

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

recursions = 0

Number of levels of links to follow on article webpages

remove_attributes = []

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

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

remove_empty_feeds = False

If True empty feeds are removed from the output. This option has no effect if parse_index is overridden in the sub class. It is meant only for recipes that return a list of feeds using *feeds* or `get_feeds()` (halaman 37). 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': 'advertisment'}
}
```

All keys are optional. For a full explanation of the search criteria, see Beautiful Soup 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` (halaman 43). 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` (halaman 43). For example:

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

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

`requires_version = (0, 6, 0)`

Minimum calibre version needed to use this recipe

`resolve_internal_links = False`

If set to True then links in downloaded articles that point to other downloaded articles are changed to point to the downloaded copy of the article rather than its original web URL. If you set this to True, you might also need to implement `canonicalize_internal_url()` (halaman 36) to work with the URL scheme of your particular website.

`reverse_article_order = False`

Reverse the order of articles in each feed

`scale_news_images = None`

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

`scale_news_images_to_device = True`

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

`simultaneous_downloads = 5`

Number of simultaneous downloads. Set to 1 if the server is picky. Automatically reduced to 1 if `BasicNewsRecipe.delay` (halaman 41) > 0

`summary_length = 500`

Max number of characters in the short description

`template_css = '\n .article_date {\n color:  gray; font-family:  monospace;\n }\n\n .a`

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 = 'Sumber Beritah Tidak Diketahui'`

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.

CHAPTER 3

Pelihat E-buku

calibre menyertakan pelihat e-buku terbina-dalam yang dapat melihat semua format e-buku utama. Pelihat E-buku ini boleh diubahsuai dan mempunyai banyak fitur lanjutan.

- *Memulakan pelihat E-buku* (halaman 45)
- *Menavigasi e-buku* (halaman 46)
- *Highlighting text* (halaman 47)
- *Read aloud* (halaman 47)
- *Penyuaian penampilan bagi pembacaan anda* (halaman 48)
- *Carian kamus* (halaman 48)
- *Penyalinan teks dan imej* (halaman 48)
- *Zooming in on images* (halaman 48)
- *Non re-flowable content* (halaman 48)

3.1 Memulakan pelihat E-buku

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

3.2 Menavigasi e-buku

You can "turn pages" in a book by either:

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

You can access the viewer controls by either:

- Right clicking on the text
- Pressing the `Esc` or `Menu` keys
- On a touchscreen by tapping the top 1/3rd of the screen

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

3.2.1 Tanda buku

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

3.2.2 Senarai Kandungan

If the book you are reading defines a Table of Contents, you can access it by pressing the *Table of Contents* button. This will bring up a list of sections in the book. You can click on any of them to jump to that portion of the book.

3.2.3 Navigasi mengikut lokasi

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

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

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

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

3.2.4 Mod rujukan

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

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

3.3 Highlighting text

When you select text in the viewer, a little popup bar appears next to the selection. You can click the highlight button in that bar to create a highlight. You can add notes and change the color of the highlight. On a touch screen, long tap a word to select it and show the popup bar. Once in highlight mode you can tap the *Adjust selection* button to change what text is selected, using touch screen friendly selection handles. Drag the handles to the top or bottom margins to scroll while selecting.

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

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

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

3.4 Read aloud

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

You can also read aloud highlighted passages by adding the *Read aloud* button to the selection bar in the viewer preferences under *Selection behavior*.

Note: Support for text-to-speech in browsers is very incomplete and bug-ridden so how well *Read aloud* will work in the in-browser viewer is dependent on how well the underlying browser supports text-to-speech. In particular, highlighting of current word does not work, and changing speed or voice will cause reading to start again from the beginning.

Note: On Linux, *Read aloud* requires Speech Dispatcher to be installed and working.

3.5 Penyuaian penampilan bagi pembacaan anda

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

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

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

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

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

3.6 Carian kamus

You can look up the meaning of words in the current book by opening the *Lookup/search word panel* via the viewer controls. Then simply double click on any word and its definition will be displayed in the Lookup panel.

3.7 Penyalinan teks dan imej

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

3.8 Zooming in on images

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

3.9 Non re-flowable content

Some books have very wide content that content be broken up at page boundaries. For example tables or `<pre>` tags. In such cases, you should switch the viewer to *flow mode* by pressing `Ctrl+m` to read this content. Alternately, you can also add the following CSS to the *Styles* section of the viewer preferences to force the viewer to break up lines of text in `<pre>` tags:

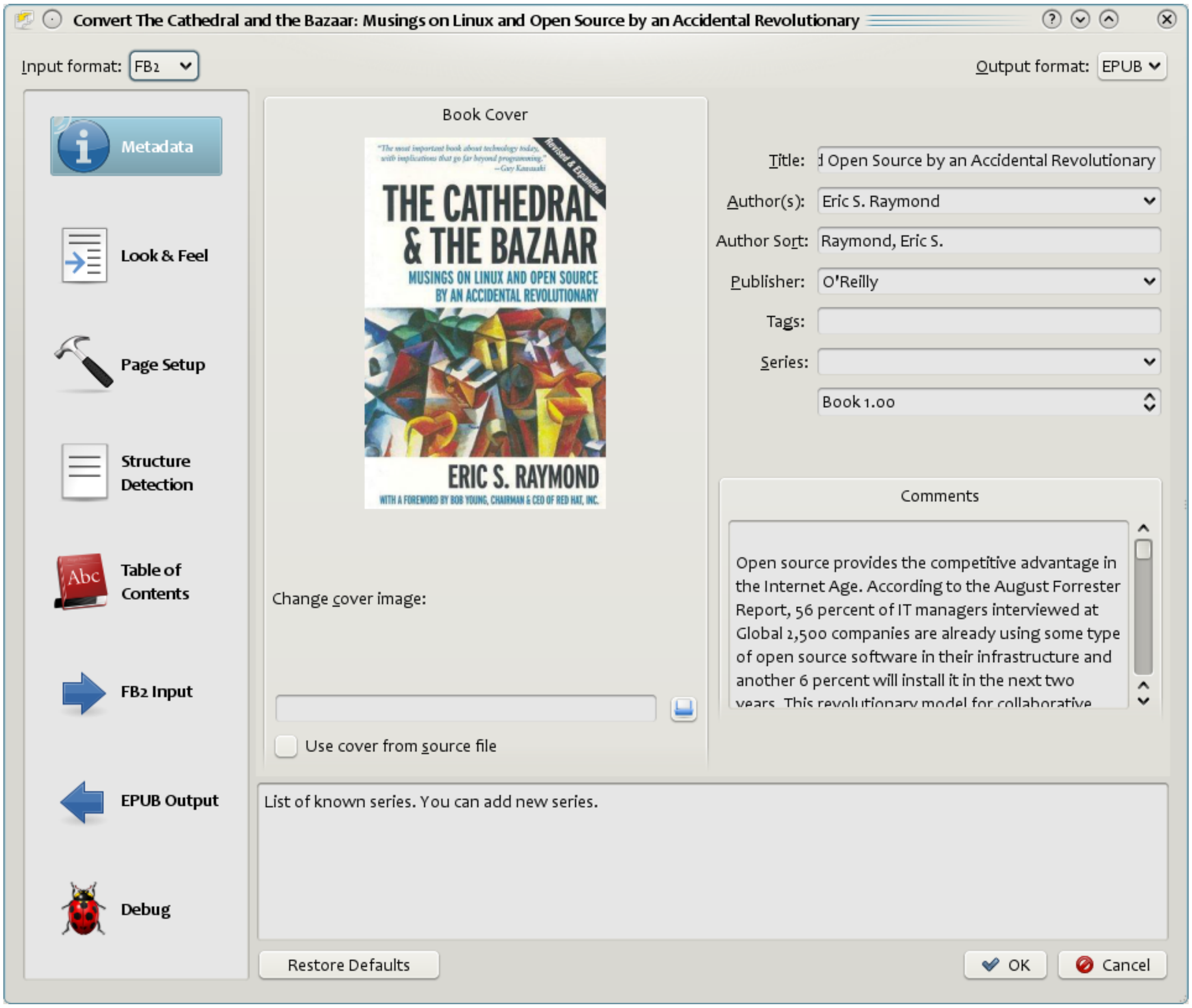
```
code, pre { white-space: pre-wrap }
```

CHAPTER 4

E-book conversion

calibre has a conversion system that is designed to be very easy to use. Normally, you just add a book to calibre, click convert and calibre will try hard to generate output that is as close as possible to the input. However, calibre accepts a very large number of input formats, not all of which are as suitable as others for conversion to e-books. In the case of such input formats, or if you just want greater control over the conversion system, calibre has a lot of options to fine tune the conversion process. Note however that calibre's conversion system is not a substitute for a full blown e-book editor. To edit e-books, I recommend first converting them to EPUB or AZW3 using calibre and then using the *Edit book* feature to get them into perfect shape. You can then use the edited e-book as input for conversion into other formats in calibre.

This document will refer mainly to the conversion settings as found in the conversion dialog, pictured below. All these settings are also available via command line interface to conversion, documented at generated/en/ebook-convert. In calibre, you can obtain help on any individual setting by holding your mouse over it, a tooltip will appear describing the setting.



Convert The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary

Input format: FB2

Output format: EPUB

Title: d Open Source by an Accidental Revolutionary

Author(s): Eric S. Raymond

Author Sort: Raymond, Eric S.

Publisher: O'Reilly

Tags:

Series:

Book 1.00

☐ Use cover from source file

Open source provides the competitive advantage in the Internet Age. According to the August Forrester Report, 56 percent of IT managers interviewed at Global 2,500 companies are already using some type of open source software in their infrastructure and another 6 percent will install it in the next two years. This revolutionary model for collaborative

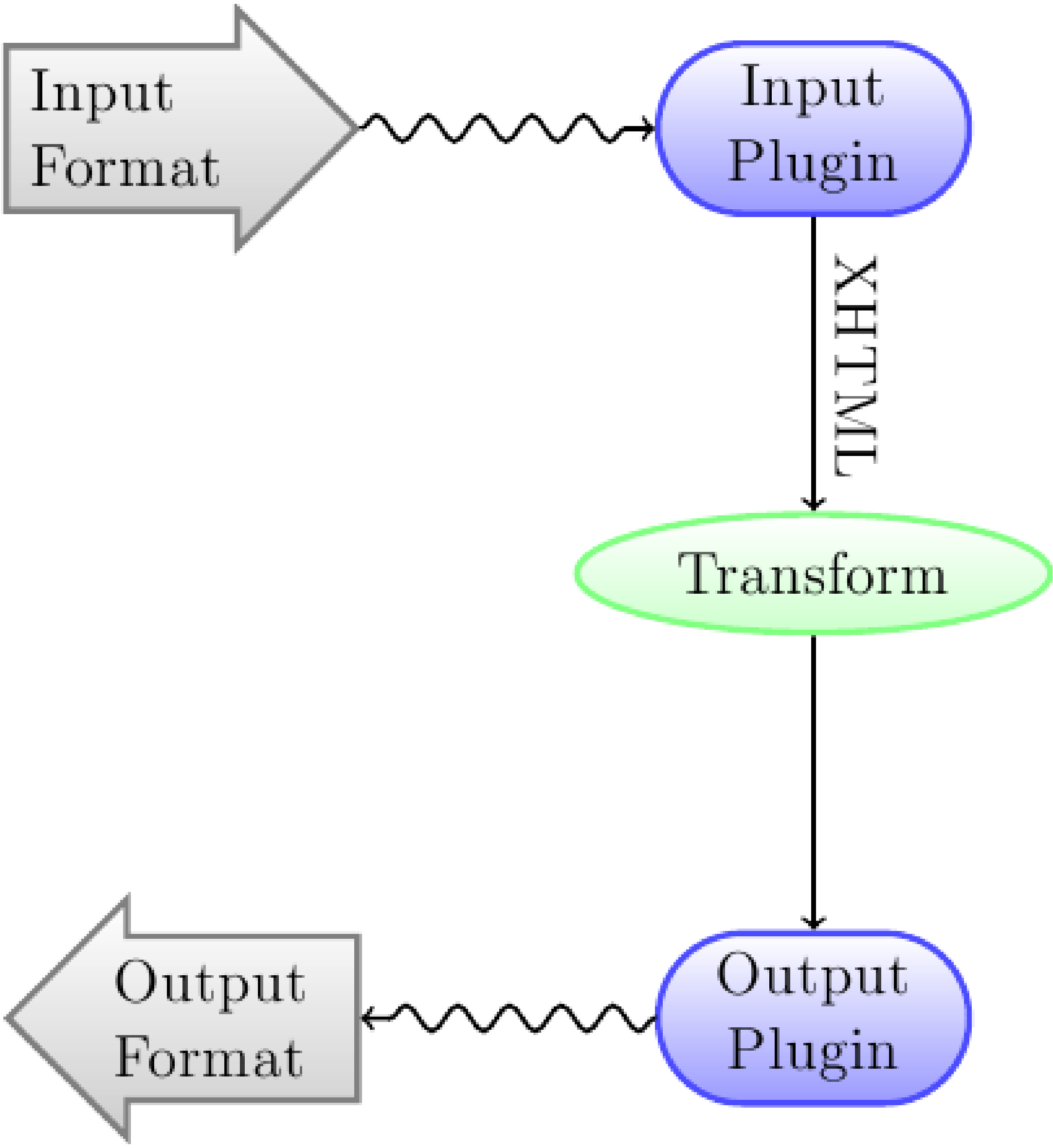
List of known series. You can add new series.

Kandungan

- *Pengenalan* (halaman 51)
- *Penampilan* (halaman 52)
- *Persediaan halaman* (halaman 55)
- *Pemprosesen heuristik* (halaman 55)
- *Search & replace* (halaman 56)
- *Structure detection* (halaman 57)
- *Senarai Kandungan* (halaman 58)
- *Using images as chapter titles when converting HTML input documents* (halaman 59)
- *Using tag attributes to supply the text for entries in the Table of Contents* (halaman 60)
- *How options are set/saved for Conversion* (halaman 60)
- *Format specific tips* (halaman 61)

4.1 Pengenalan

Perkara pertama untuk memahami sistem pertukaran format ialah ia direka sebagai satu talian paip. Secara skematik, ia kelihata sebegini:



The input format is first converted to XHTML by the appropriate *Input plugin*. This HTML is then *transformed*. In the last step, the processed XHTML is converted to the specified output format by the appropriate *Output plugin*. The results of the conversion can vary greatly, based on the input format. Some formats convert much better than others. A list of the best source formats for conversion is available *here* (halaman 118).

The transforms that act on the XHTML output are where all the work happens. There are various transforms, for example, to insert book metadata as a page at the start of the book, to detect chapter headings and automatically create a Table of Contents, to proportionally adjust font sizes, et cetera. It is important to remember that all the transforms act on the XHTML output by the *Input plugin*, not on the input file itself. So, for example, if you ask calibre to convert an RTF file to EPUB, it will first be converted to XHTML internally, the various transforms will be applied to the XHTML and then the *Output plugin* will create the EPUB file, automatically generating all metadata, Table of Contents, et cetera.

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

Table1: Tahap talian paip penukaran format

Direktori	Keterangan
input	This contains the HTML output by the Input plugin. Use this to debug the Input plugin.
terhurai	The result of pre-processing and converting to XHTML the output from the Input plugin. Use to debug structure detection.
struktur	Pengesanan struktur pasca, tetapi sebelum meratakan CSS dan penukaran saiz fon. Digunakan untuk menyahpepijat penukaran saiz fon dan penjelmaan CSS.
diproses	Just before the e-book is passed to the Output plugin. Use to debug the Output plugin.

If you want to edit the input document a little before having calibre convert it, the best thing to do is edit the files in the `input` sub-directory, 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 Penampilan

Kandungan

- *Fon* (halaman 52)
- *Teks* (halaman 53)
- *Bentangan* (halaman 54)
- *Pengayaan* (halaman 54)
- *Gaya jelma* (halaman 54)

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

4.2.1 Fon

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 *Persediaan halaman* (halaman 55)). 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.

Let's illustrate with an example. Suppose the source document we are converting was produced by someone with excellent eyesight and has a base font size of 8pt. That means the bulk of the text in the document is sized at 8pts, while headings are somewhat larger (say 10 and 12pt) and footnotes somewhat smaller at 6pt. Now if we use the following settings:

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

The output document will have a base font size of 12pt, headings of 14 and 16pt and footnotes of 8pt. Now suppose we want to make the largest heading size stand out more and make the footnotes a little larger as well. To achieve this, the font key should be changed to:

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

The largest headings will now become 18pt, while the footnotes will become 9pt. You can play with these settings to try and figure out what would be optimum for you by using the font rescaling wizard, which can be accessed by clicking the little button next to the *Font size key* setting.

Semua penskalaan semula saiz fon dalam pertukaran format boleh juga dilumpuhkan di sini, jika anda mahu kekalkan saiz fon seperti dalam dokumen 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* (halaman 54).

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 Teks

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

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

Finally, there is *Input character encoding*. Older documents sometimes don't specify their character encoding. When converted, this can result in non-English characters or special characters like smart quotes being corrupted. calibre tries to auto-detect the character encoding of the source document, but it does not always succeed. You can force it to assume a particular character encoding by using this setting. *cp1252* is a common encoding for documents produced using Windows software. You should also read *How do I convert my file containing non-English characters, or smart quotes?* (halaman 118) for more on encoding issues.

4.2.3 Bentangan

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* (halaman 54):

```
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-directory).

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 Pengayaan

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 }
```

atau jika anda mahu ubah indentasi bagi semua perenggan:

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

Extra CSS is a very powerful option, but you do need an understanding of how CSS works to use it to its full potential. You can use the debug pipeline option described above to see what CSS is present in your input document.

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

4.2.5 Gaya jelma

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.

4.3 Persediaan halaman

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. In particular, for MOBI output files, you should choose the Kindle, for LIT the Microsoft Reader and for EPUB the Sony Reader. In the case of EPUB, the Sony Reader profile will result in EPUB files that will work everywhere. However, it has some side effects, like inserting artificial section breaks to keep internal components below the size threshold, needed for SONY devices. In particular for the iPhone/Android phones, choose the SONY output profile. If you know your EPUB files will not be read on a SONY or similar device, use the default output profile. If you are producing MOBI files that are not intended for the Kindle, choose the Mobipocket books 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 Pemprosesen heuristik

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

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

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

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

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

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

The inserted headings are not formatted, to apply formatting use the *Extra CSS* option under the Look and Feel conversion settings. For example, to center heading tags, use the following:

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

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

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

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

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

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

Example image tag (place the image within an 'Images' folder inside the EPUB after conversion): <img style="width:10%" src="../Images/scenebreak.png" />

Contoh peraturan mengufuk dengan gaya: <hr style="width:20%;padding-top: 1px;border-top: 2px ridge black;border-bottom: 2px groove black;"/>

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

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

Replace entity indents with CSS indents Some documents use a convention of defining text indents using non-breaking space entities. When this option is enabled calibre will attempt to detect this sort of formatting and convert them to a 3% text indent using CSS.

4.5 Search & replace

These options are useful primarily for conversion of PDF documents or OCR conversions, though they can also be used to fix many document specific problems. As an example, some conversions can leaves behind page headers and footers in the text. These options use regular expressions to try and detect headers, footers, or other arbitrary text and remove or replace them. Remember that they operate on the intermediate XHTML produced by the conversion pipeline. There is a wizard to help you customize the regular expressions for your document. Click the magic wand beside the expression box, and click the 'Test' button after composing your search expression. Successful matches will be highlighted in Yellow.

The search works by using a Python regular expression. All matched text is simply removed from the document or replaced using the replacement pattern. The replacement pattern is optional, if left blank then text matching the search pattern will be deleted from the document. You can learn more about regular expressions and their syntax at *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman 184).

4.6 Structure detection

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 Chapters and page breaks

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* (halaman 145) 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 *Pengenalan* (halaman 51) 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 chapter detection:

```
//*[((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.

The default setting for detecting page breaks is:

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

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

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

4.6.2 Pelbagai

Terdapat sedikit lagi pilihan dalam seksyen ini.

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.

Remove first image 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 Senarai Kandungan

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.

Note: 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 *Structure detection* (halaman 57) section above. If you do not want to include detected chapters in the generated table of contents, check the *Do not add detected chapters* option.

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

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

```
Next|Previous
```

The *Level 1,2,3 TOC* options allow you to create a sophisticated multi-level Table of Contents. They are XPath expressions that match tags in the intermediate XHTML produced by the conversion pipeline. See the *Pengenalan* (halaman 51) for how to get access to this XHTML. Also read the *Tutorial XPath* (halaman 145), to learn how to construct XPath expressions. Next to each option is a button that launches a wizard to help with the creation of basic XPath expressions. The following simple example illustrates how to use these options.

Suppose you have an input document that results in XHTML that look like this:

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Sample document</title>
    </head>
    <body>
        <h1>Chapter 1</h1>
        ...
        <h2>Section 1.1</h2>
        ...
        <h2>Section 1.2</h2>
        ...
        <h1>Chapter 2</h1>
        ...
        <h2>Section 2.1</h2>
        ...
    </body>
</html>
```

Then, we set the options as:

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

This will result in an automatically generated two level Table of Contents that looks like:

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

Warning: 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 Using images as chapter titles when converting HTML input documents

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

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

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

4.9 Using tag attributes to supply the text for entries in the Table of Contents

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

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

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

4.10 How options are set/saved for Conversion

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

You can also change settings in the conversion dialog for each book conversion. When you convert a book, calibre remembers the settings you used for that book, so that if you convert it again, the saved settings for the individual book will take precedence over the defaults set in *Preferences*. You can restore the individual settings to defaults by using the *Restore defaults* button in the individual book conversion dialog. You can remove the saved settings for a group of books by selecting all the books and then clicking the *Edit metadata* button to bring up the bulk metadata edit dialog, near the bottom of the dialog is an option to remove stored conversion settings.

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

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

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

Note: 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 Format specific tips

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 Tukar format dokumen Microsoft Word

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

Note: There is a demo .docx file 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.

Older .doc files

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

Another alternative is to use the free OpenOffice. Open your .doc file in OpenOffice and save it in OpenOffice's format .odt. calibre can directly convert .odt files.

4.11.2 Tukar format dokumen 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 Assumes that every line is a paragraph:

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

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

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

  This is the
third.
```

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

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

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

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

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

4.11.3 Convert PDF documents

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 *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman 184).

Some limitations of PDF input are:

- Complex, multi-column, and image based documents are not supported.
- Extraction of vector images and tables from within the document is also not supported.
- Some PDFs use special glyphs to represent ll or ff or fi, etc. Conversion of these may or may not work depending on just how they are represented internally in the PDF.
- Links 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 OCRed text behind them. In such cases calibre uses the OCRed text, which can be very different from what you see when you view the PDF file

- PDFs that are used to display complex text, like right to left languages and math typesetting will not convert correctly

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

4.11.4 Comic book collections

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

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

The .cbc file will then contain:

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

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

4.11.5 EPUB advanced formatting demo

Various advanced formatting for EPUB files is demonstrated in this demo file. 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. 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 Convert ODT documents

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 build-in styles called 'Heading 1', 'Heading 2', ..., 'Heading 6' ('Heading 1' equates to the HTML tag <h1>, 'Heading 2' to <h2> etc). When you convert in calibre you can enter which style you used into the 'Detect chapters at' box. Example:

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

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

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

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

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

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

4.11.7 Converting to 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.

Headers and Footers

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

Note: 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.

Senarai Kandungan Boleh Cetak

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;
}
```

(continues on next page)

(continued from previous page)

```
.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>
```

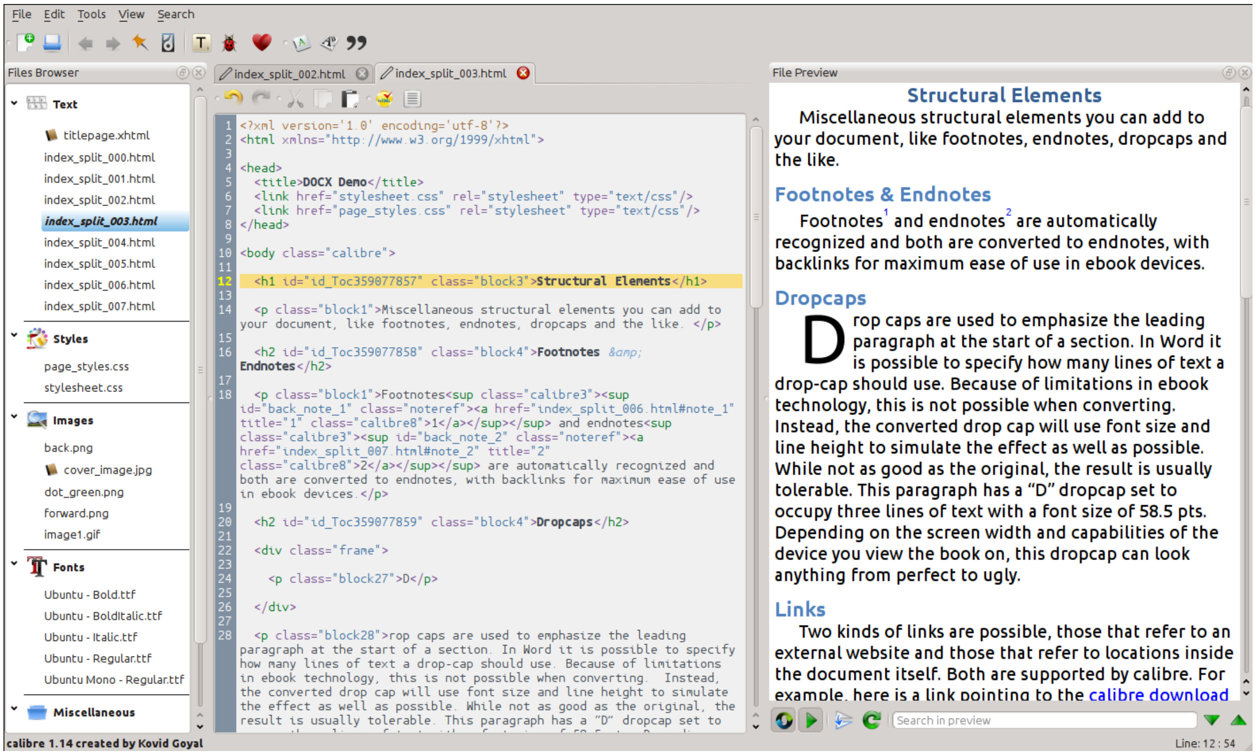
Then, in the PDF output section of the conversion dialog, turn on the option to *Use page margins from the document being converted*. Now all pages generated from this HTML file will have `10pt` margins.

CHAPTER 5

Menyunting e-buku

calibre mempunyai penyunting e-buku bersepadu yang boleh digunakan untuk menyunting buku dalam format EPUB dan AZW3 (Kindle). Penyunting dapat menunjukkan HTML dan CSS yang digunakan secara dalaman di dalam fail buku, dengan pratonton langsung bilamana anda membuat perubahan. Ia juga mengandungi pelbagai alatan berautomatik yang dapat melakukan tugas-tugas pembersihan dan pembaikan.

Anda boleh guna penyunting ini dengan mengklik-kanan pada mana-mana buku di dalam calibre kemudian memilih *Sunting buku*.



```
<?xml version='1.0' encoding='utf-8'?>
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <title>DOCX Demo</title>
  <link href="stylesheet.css" rel="stylesheet" type="text/css"/>
  <link href="page_styles.css" rel="stylesheet" type="text/css"/>
</head>

<body class="calibre">

  <h1 id="id_Toc359077857" class="block3">Structural Elements</h1>

  <p class="block1">Miscellaneous structural elements you can add to
your document, like footnotes, endnotes, dropcaps and the like. </p>

  <h2 id="id_Toc359077858" class="block4">Footnotes &amp;
Endnotes</h2>

  <p class="block1">Footnotes<sup class="calibre3"><sup
id="back_note_1" class="noteref"><a href="index_split_006.html#note_1"
title="1" class="calibre8">1</a></sup></sup> and endnotes<sup
class="calibre3"><sup id="back_note_2" class="noteref"><a
href="index_split_007.html#note_2" title="2"
class="calibre8">2</a></sup></sup> are automatically recognized and
both are converted to endnotes, with backlinks for maximum ease of use
in ebook devices.</p>

  <h2 id="id_Toc359077859" class="block4">Dropcaps</h2>

  <div class="frame">

    <p class="block27">D</p>

  </div>

  <p class="block28">rop caps are used to emphasize the leading
paragraph at the start of a section. In Word it is possible to specify
how many lines of text a drop-cap should use. Because of limitations
in ebook technology, this is not possible when converting.  Instead,
the converted drop cap will use font size and line height to simulate
the effect as well as possible. While not as good as the original, the
result is usually tolerable. This paragraph has a “D” dropcap set to
```

Structural Elements

Miscellaneous structural elements you can add to your document, like footnotes, endnotes, dropcaps and the like.

Footnotes & Endnotes

Footnotes[1] and endnotes[2] are automatically recognized and both are converted to endnotes, with backlinks for maximum ease of use in ebook devices.

Dropcaps

Drop caps are used to emphasize the leading paragraph at the start of a section. In Word it is possible to specify how many lines of text a drop-cap should use. Because of limitations in ebook technology, this is not possible when converting. Instead, the converted drop cap will use font size and line height to simulate the effect as well as possible. While not as good as the original, the result is usually tolerable. This paragraph has a “D” dropcap set to occupy three lines of text with a font size of 58.5 pts. Depending on the screen width and capabilities of the device you view the book on, this dropcap can look anything from perfect to ugly.

Links

Two kinds of links are possible, those that refer to an external website and those that refer to locations inside the document itself. Both are supported by calibre. For example, here is a link pointing to the calibre download

Kandungan

- *Aliran kerja asas* (halaman 69)
- *The File browser* (halaman 71)
  - *Menamakan semula fail* (halaman 72)
  - *Merging files* (halaman 72)
  - *Changing text file order* (halaman 73)
  - *Marking the cover* (halaman 73)
  - *Deleting files* (halaman 73)
  - *Exporting files* (halaman 73)
  - *Adding new images/fonts/etc. or creating new blank files* (halaman 73)
  - *Replacing files* (halaman 74)
  - *Linking stylesheets to HTML files efficiently* (halaman 74)
- *Search & replace* (halaman 74)
  - *Gelintar tersimpan* (halaman 74)
  - *Function mode* (halaman 75)
  - *Search ignoring HTML tags* (halaman 75)
- *Automated tools* (halaman 75)
  - *Editing the Table of Contents* (halaman 75)
  - *Checking the Book* (halaman 76)
  - *Adding a cover* (halaman 77)
  - *Membenamkan fon rujukan* (halaman 77)
  - *Subsetkan fon terbenam* (halaman 77)
  - *Memintarkan tanda baca* (halaman 78)
  - *Transforming CSS properties* (halaman 78)
  - *Membuang peraturan CSS yang tidak digunakan* (halaman 78)
  - *Fixing HTML* (halaman 78)
  - *Beautifying files* (halaman 78)
  - *Inserting an inline Table of Contents* (halaman 79)
  - *Setting Semantics* (halaman 79)
  - *Filtering style information* (halaman 79)
  - *Upgrading the book's internals* (halaman 79)
- *Titik semak* (halaman 79)
- *The Live preview panel* (halaman 81)
  - *Splitting HTML files* (halaman 82)
- *The Live CSS panel* (halaman 83)

- *Miscellaneous tools* (halaman 84)
  - *The Table of Contents view* (halaman 84)
  - *Checking the spelling of words in the book* (halaman 84)
  - *Inserting special characters* (halaman 86)
  - *The code inspector view* (halaman 86)
  - *Checking external links* (halaman 87)
  - *Downloading external resources* (halaman 87)
  - *Arranging files into folders by type* (halaman 87)
  - *Importing files in other e-book formats as EPUB* (halaman 87)
  - *The Reports tool* (halaman 98)
- *Special features in the code editor* (halaman 98)
  - *Syntax highlighting* (halaman 98)
  - *Bantuan peka konteks* (halaman 99)
  - *Auto-lengkap* (halaman 99)
  - *Snippet* (halaman 99)

5.1 Aliran kerja asas

Note: A video tour of the calibre E-book editor is available here.

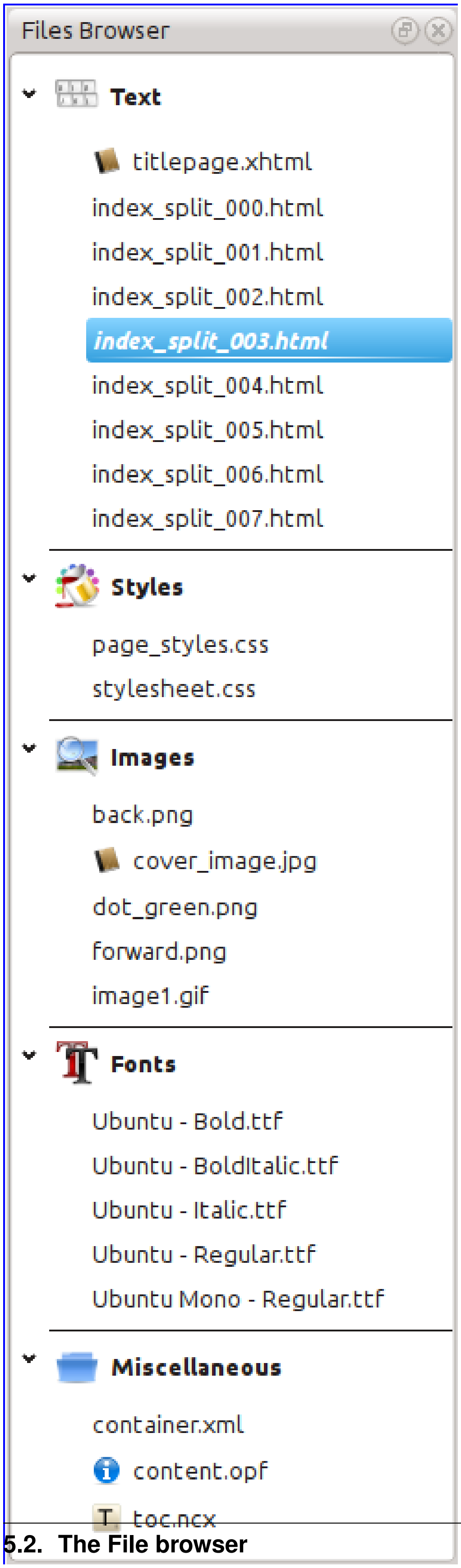
Bila anda mula-mula membuka buku dengan Alat sunting buku, anda akan dipersembahkan dengan satu senarai fail di sebelah kiri. Ia merupakan fail HTML secara individu, lembaran gaya-lembaran gaya, imej dan lain-lain item yang membentuk kandungan buku. Hanya dwi-klik pada fail tersebut untuk memulakan kerja-kerja penyuntingan. Perhatian, jika anda mahu lakukan kerja-kerja yang lebih bersifat teknikal, anda perlu ketahui Tutorial HTML dan Tutorial CSS.

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 useful for when changes are spread over multiple files in the book.

Itulah aliran kerja asas untuk penyuntingan buku -- Buka satu fail, buat perubahan pada buku tersebut, pratonton dan simpan. Selebihnya panduan ini akan membincangkan pelbagai jenis alatan dan fitur yang ada supaya anda dapat membuat tugasan tertentu dengan efisyen.

5.2 The File browser



Text

titlepage.xhtml

index_split_000.html

index_split_001.html

index_split_002.html

index_split_003.html

index_split_004.html

index_split_005.html

index_split_006.html

index_split_007.html

Styles

page_styles.css

stylesheet.css

Images

back.png

cover_image.jpg

dot_green.png

forward.png

image1.gif

Fonts

Ubuntu - Bold.ttf

Ubuntu - BoldItalic.ttf

Ubuntu - Italic.ttf

Ubuntu - Regular.ttf

Ubuntu Mono - Regular.ttf

Miscellaneous

container.xml

content.opf

toc.ncx

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.

Anda boleh membuat banyak tindakan pada sesebuah fail, dengan mengklik kanan padanya.

5.2.1 Menamakan semula fail

Anda boleh menamakan semula sesebuah fail dengan mengklik-kanan padanya dan pilih *Nama Semula*. Menamakan semula fail secara automatik akan kemaskinikan semua pautan dan rujukan pada buku tersebut. Oleh itu, anda perlu berikan nama baharu, calibre seterusnya menguruskan kerja-kerja berikutnya.

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 Merging files

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.

5.2.3 Changing text file order

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 Files browser. 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.

5.2.4 Marking the cover

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 Deleting files

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 Exporting files

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 Adding new images/fonts/etc. or creating new blank files

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 Replacing files

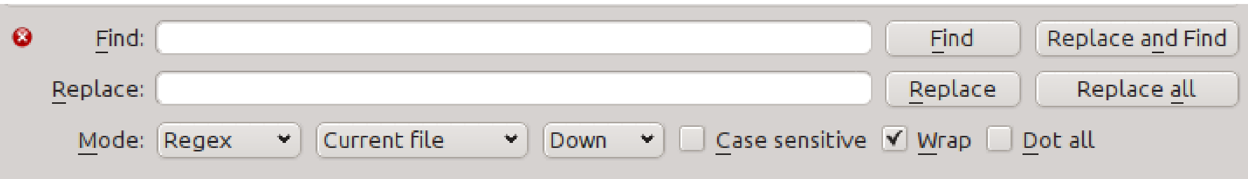
You can easily replace existing files in the book, by right clicking on the file and choosing replace. This will automatically update all links and references, in case the replacement file has a different name than the file being replaced.

5.2.9 Linking stylesheets to HTML files efficiently

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 Search & replace

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 *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman 184).



Find:
Replace:
Mode: Regex | Current file | Down | □ Case sensitive | ☑ Wrap | □ Dot all

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

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

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

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

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

Note: Remember, to harness the full power of search and replace, you will need to use regular expressions. See *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman 184).

5.3.1 Gelintar tersimpan

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 Function mode

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 *Mod Fungsi untuk Gelintar & Ganti di dalam Penyunting* (halaman **??**).

5.3.3 Search ignoring HTML tags

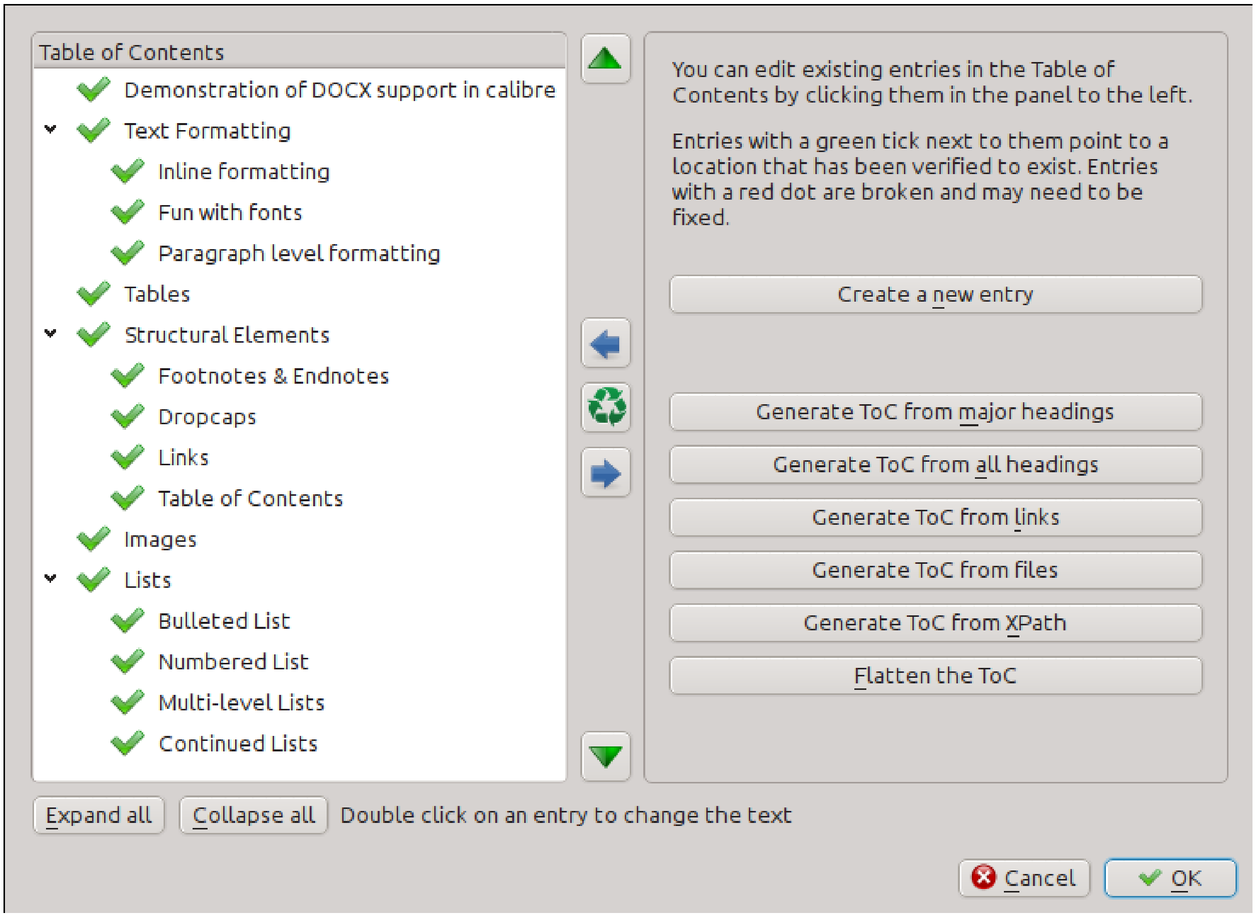
There is also a dedicated tool for searching for text, ignoring any HTML tags in between. For example, if the book has the HTML `Empahisis on a <i>word</i>.` you can search for `on a word` and it will be found even though there is an `<i>` tag in the middle. Use this tool via the *Search→Search ignoring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

5.4 Automated tools

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

5.4.1 Editing the Table of Contents

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



You can edit existing entries in the Table of Contents by clicking them in the panel to the left.

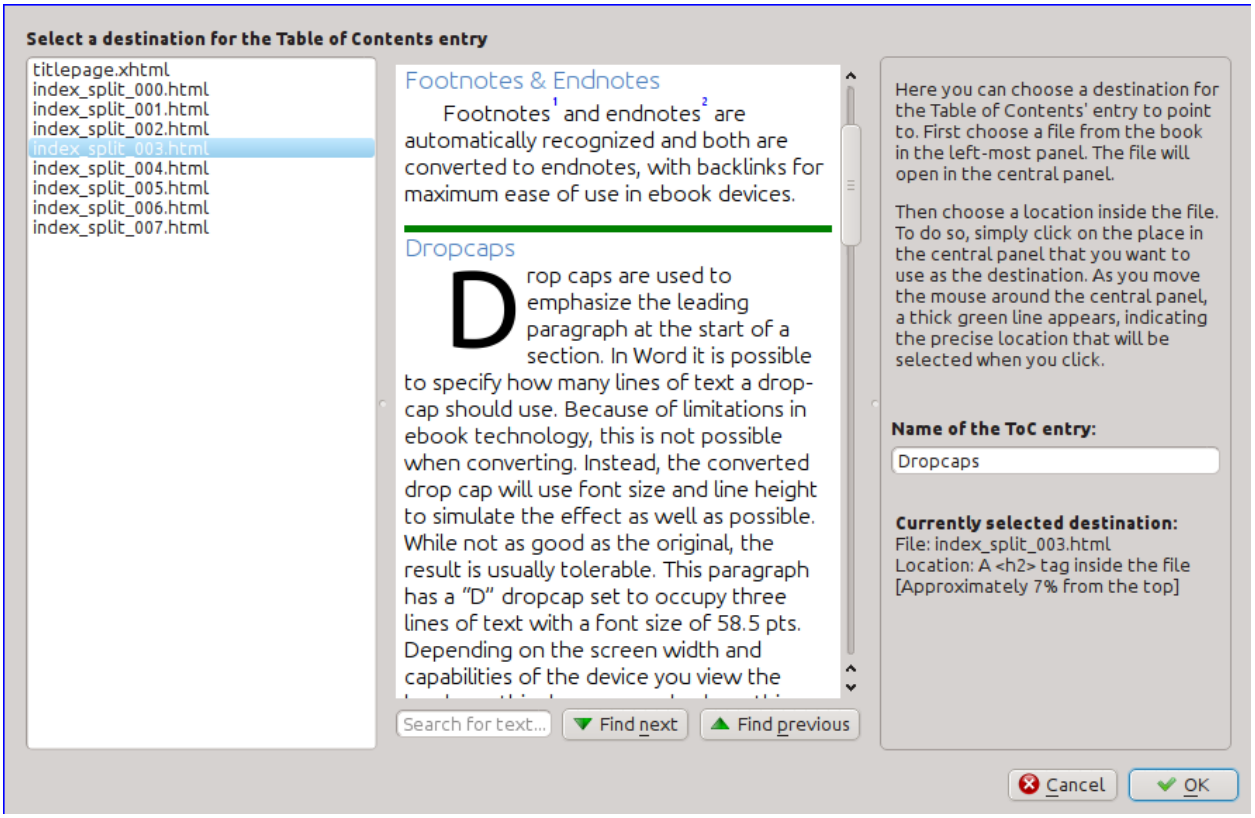
Entries with a green tick next to them point to a location that has been verified to exist. Entries with a red dot are broken and may need to be fixed.

Double click on an entry to change the text

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.



Select a destination for the Table of Contents entry

titlepage.xhtml
index_split_000.html
index_split_001.html
index_split_002.html
index_split_003.html
index_split_004.html
index_split_005.html
index_split_006.html
index_split_007.html

Footnotes & Endnotes

Footnotes[1] and endnotes[2] are automatically recognized and both are converted to endnotes, with backlinks for maximum ease of use in ebook devices.

Dropcaps

Drop caps are used to emphasize the leading paragraph at the start of a section. In Word it is possible to specify how many lines of text a drop-cap should use. Because of limitations in ebook technology, this is not possible when converting. Instead, the converted drop cap will use font size and line height to simulate the effect as well as possible. While not as good as the original, the result is usually tolerable. This paragraph has a "D" dropcap set to occupy three lines of text with a font size of 58.5 pts. Depending on the screen width and capabilities of the device you view the

Here you can choose a destination for the Table of Contents' entry to point to. First choose a file from the book in the left-most panel. The file will open in the central panel.

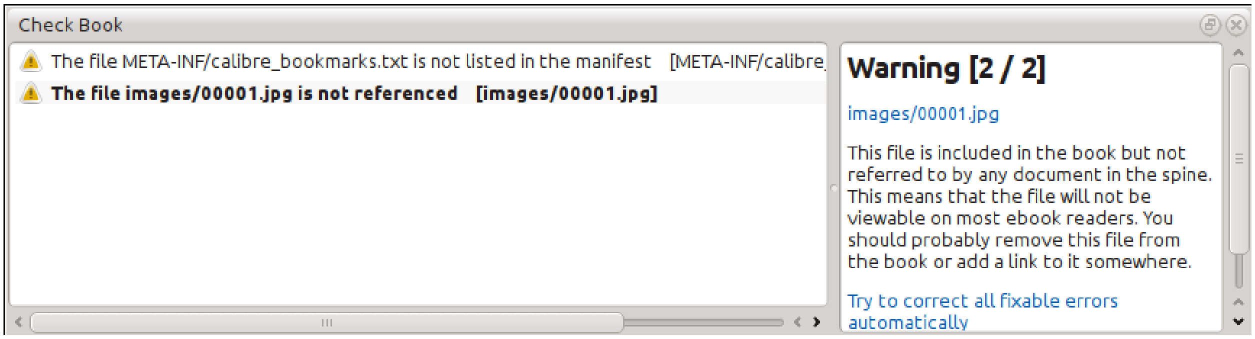
Then choose a location inside the file. To do so, simply click on the place in the central panel that you want to use as the destination. As you move the mouse around the central panel, a thick green line appears, indicating the precise location that will be selected when you click.

Name of the ToC entry: Dropcaps

Currently selected destination:
File: index_split_003.html
Location: A <h2> tag inside the file
[Approximately 7% from the top]

5.4.2 Checking the Book

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



Check Book

⚠ The file META-INF/calibre_bookmarks.txt is not listed in the manifest [META-INF/calibre
⚠ **The file images/00001.jpg is not referenced [images/00001.jpg]**

Warning [2 / 2]

images/00001.jpg

This file is included in the book but not referred to by any document in the spine. This means that the file will not be viewable on most ebook readers. You should probably remove this file from the book or add a link to it somewhere.

Try to correct all fixable errors automatically

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 Adding a cover

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 Membenamkan fon rujukan

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 Subsetkan fon terbenam

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 Memintarkan tanda baca

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 Membuang peraturan CSS yang tidak digunakan

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 Fixing 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 Beautifying files

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*.

Note: 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.

5.4.11 Inserting an inline Table of Contents

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 Setting Semantics

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 Filtering style information

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 Upgrading the book's internals

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 Titik semak

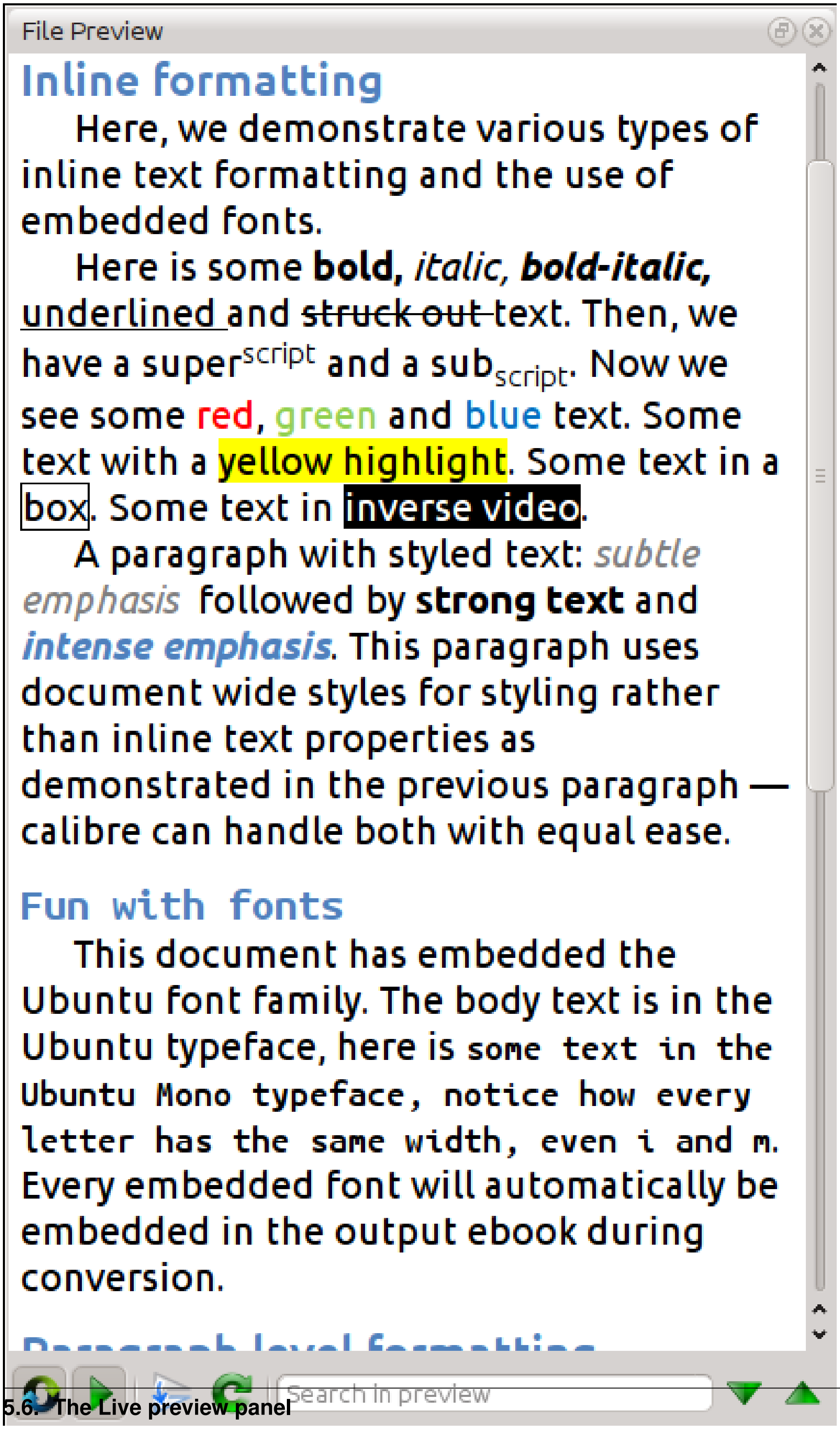
Checkpoints are a way to mark the current state of the book as "special". You can then go on to do whatever changes you want to the book and if you don't like the results, return to the checkpointed state. Checkpoints are automatically created every time you run any of the automated tools described in 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 particularly useful 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 *Membandingkan e-buku* (halaman 109) 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 The Live preview panel



Inline formatting

Here, we demonstrate various types of inline text formatting and the use of embedded fonts.

Here is some **bold**, *italic*, ***bold-italic***, underlined and ~~struck out~~ text. Then, we have a super$^{\text{script}}$ and a sub$_{\text{script}}$. Now we see some red, green and blue text. Some text with a yellow highlight. Some text in a box. Some text in inverse video.

A paragraph with styled text: *subtle emphasis* followed by **strong text** and ***intense emphasis***. This paragraph uses document wide styles for styling rather than inline text properties as demonstrated in the previous paragraph — calibre can handle both with equal ease.

Fun with fonts

This document has embedded the Ubuntu font family. The body text is in the Ubuntu typeface, here is `some text in the Ubuntu Mono typeface, notice how every letter has the same width, even i and m.` Every embedded font will automatically be embedded in the output ebook during conversion.

Paragraph level formatting

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

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

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

Note: 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.

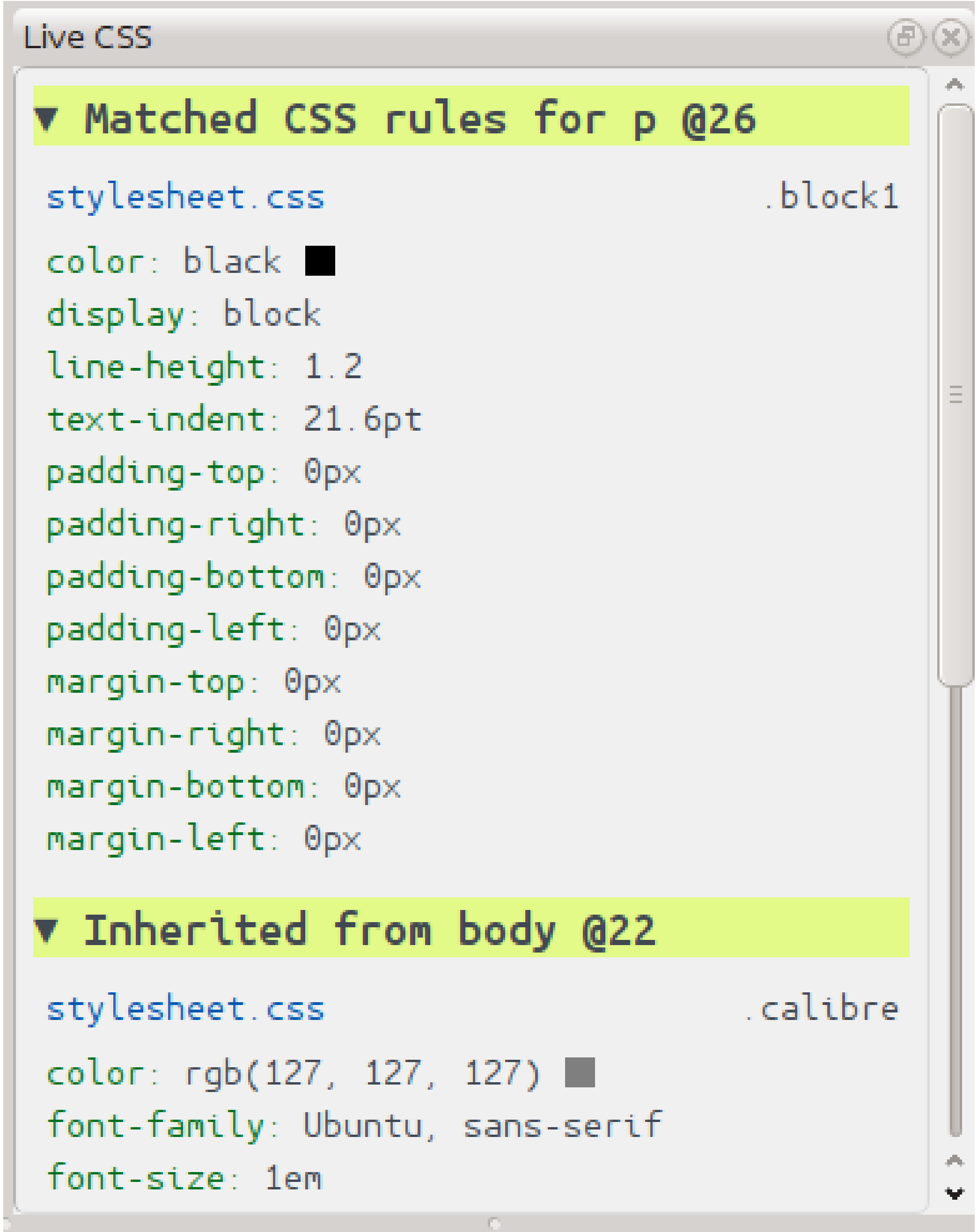
5.6.1 Splitting HTML files

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 The Live CSS panel



▼ Matched CSS rules for p @26

stylesheet.css .block1

```
color: black ■
display: block
line-height: 1.2
text-indent: 21.6pt
padding-top: 0px
padding-right: 0px
padding-bottom: 0px
padding-left: 0px
margin-top: 0px
margin-right: 0px
margin-bottom: 0px
margin-left: 0px
```

▼ Inherited from body @22

stylesheet.css .calibre

```
color: rgb(127, 127, 127) ■
font-family: Ubuntu, sans-serif
font-size: 1em
```

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 Miscellaneous tools

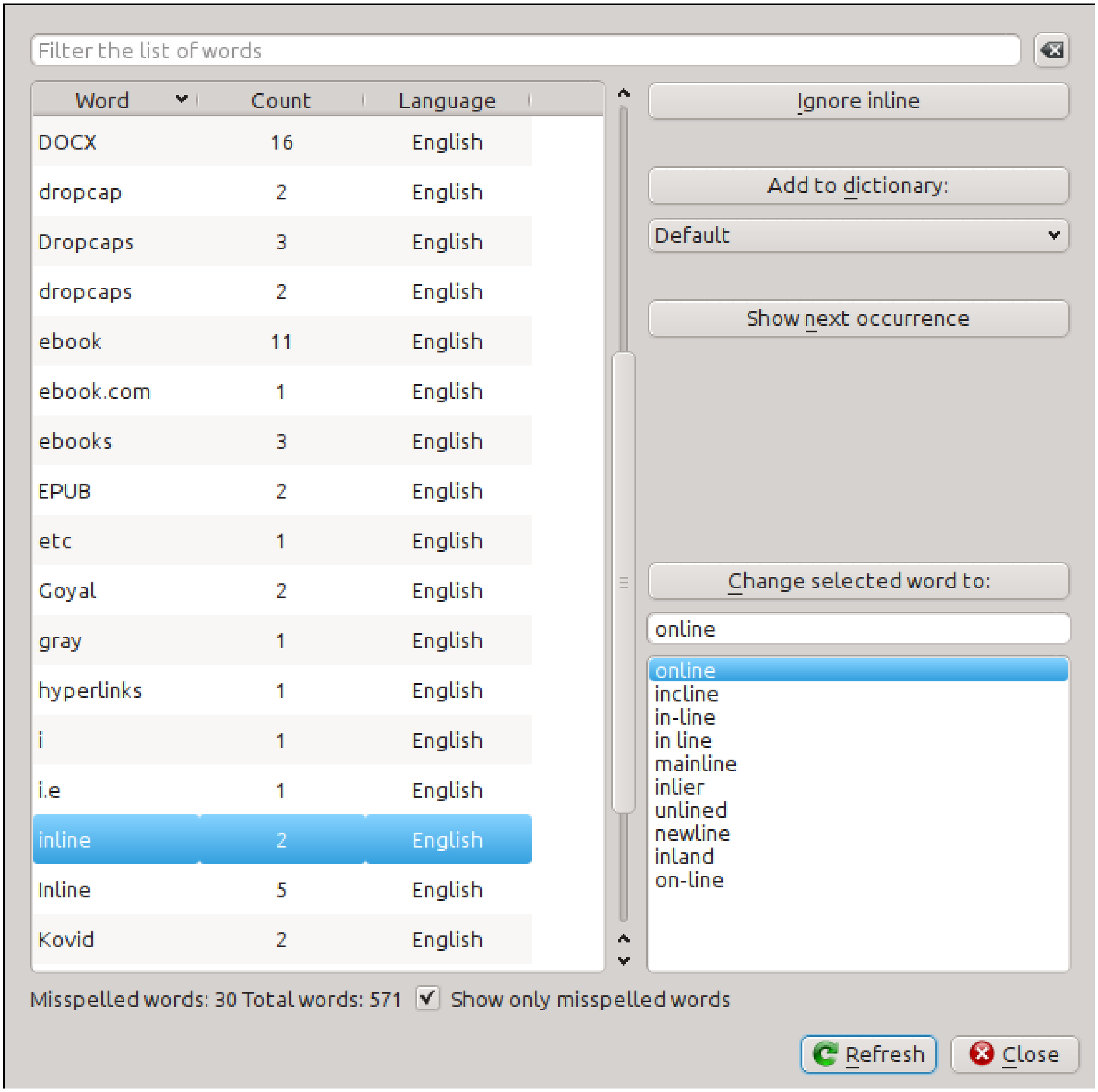
There are a few more tools that can be useful while you edit the book.

5.8.1 The Table of Contents view

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 Checking the spelling of words in the book

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



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

Default

online

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

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

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>
```

Note: 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.

Note: 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.

Adding new dictionaries

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

5.8.3 Inserting special characters

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 ß 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 &nbsp; will be replaced by a non breaking space when you type the semi-colon. The replacement happens only when typing the semi-colon.

5.8.4 The code inspector view

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 Checking external links

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 Downloading external resources

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 Arranging files into folders by type

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

5.8.8 Importing files in other e-book formats as EPUB

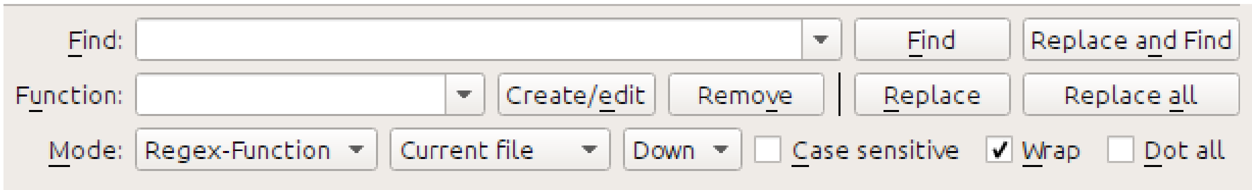
The editor includes the ability to import files in some other e-book formats directly as a new EPUB, without going through a full conversion. This is particularly useful to directly create EPUB files from your own hand-edited HTML files. You can do this via *File→Import an HTML or DOCX file as a new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

Mod Fungsi untuk Gelintar & Ganti di dalam Penyunting

Alat Gelintar & Ganti dalam penyunting menyokong *mod fungsi*. Dalam mod ini, anda boleh gabungkan ungkapan nalar (rujuk *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman **??**)) dengan fungsi hebat Python secara arbitrari untuk membuat semua bentuk pemprosesan teks lanjutan.

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

Teknik penggunaan mod fungsi dan sintaks akan dijelaskan bersama-sama dengan contoh, memudahkan penjelasan bagaimana hendak cipta fungsi yang dapat melakukan tugas yang lebih kompleks.



Membaiki huruf (case) tajuk dokumen secara automatik

Di sini, kita akan guna salah satu fungsi terbina-dalam di dalam penyunting untuk mengubah huruf semua teks dalam tag tajuk menjadi title case iaitu huruf besar pertama dalam setiap perkataan:

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

Untuk fungsi, hanya pilih fungsi terbina-dalam *teks Title-case (abai tag)*. Ia akan mengubah tajuk menjadi seperti berikut: `<h1>some TITLE</h1>` menjadi `<h1>Some Title</h1>`. Ia masih berfungsi walaupun masih terdapat tag HTML lain di dalam tag pengepala.

Fungsi suai pertama anda - hypen pintar

Kehebatan mod fungsi sebenarnya berasal dari keupayaan mencipta fungsi anda sendiri untuk memproses teks mengikut kehendak sendiri. Alat Tanda Baca Pintar di dalam penyunting membolehkan tanda sempang ditinggalkan, supaya anda boleh guna fungsi ini untuk menggantikannya dengan tanda sengkang-em.

Untuk mencipta satu fungsi baharu, hanya klik butang *Cipta/sunting* untuk mencipta satu fungsi baharu dan salin kod Python dari bawah.

```
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 wont worry about all the different arguments to `replace()` function. Just focus on the `match` argument. It represents a match when running a search and replace. Its full documentation in available here. `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.

Guna fungsi ini dengan ungkapan nalar find:

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

Dan ia akan gantikan semua tanda sengkang dengan sengkang-em, tetapi hanya dalam teks sebenar dan tidak di dalam takrifan tag HTML/

Kehebatan mod fungsi - menggunakan kamus ejaan untuk baiki perkataan tertinggal-sempang

Biasanya, e-buku dicipta dari imbasan buku bercetak yang menganfungi perkataan tertinggal-sempang -- perkataan yang mana dipisah pada akhir baris pada halaman bercetak. Kami akan hasilkan fungsi ringkas untuk cari dan baiki perkataan tersebut secara automatik.

```
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)
```

(continues on next page)

(continued from previous page)

```
        if dictionaries.recognized(without_hyphen):
            return without_hyphen
        return wmatch.group()

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

Guna fungsi ini dengan ungkapan find yang sama sebelum ini, iaitu:

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

Dan ia akan membaiki secara magik semua perkataan yang tiada-sempang dalam teks buku. Caranya adalah guna salah satu argumen tambahan untuk menggantikan fungsi, `dictionaries`. Ini merujuk pada kamus penyunting itu sendiri menggunakan semakan ejaan teks di dalam buku tersebut. Fungsi ini mencari perkataan yang diasing dengan tanda sempang, buang tanda sempang tersebut dan periksa jika kamus mengenalpasti perkataan komposit itu, jika ada, perkataan asal akan digantikan dengan perkataan yang tidak menggunakan tanda sempang tersebut.

Perhatian satu kekangan bagi teknik ini adalah ia hanya berfungsi baik dengan buku satu bahasa, kerana, secara lalai, `dictionaries.recognized()` menggunakan bahasa utama buku.

Seksyen pernomboran automatik

Sekarang kita dapati ada beberapa perkara berlainan. Anggaplah fail HTML anda mempunyai banyak seksyen, setiapnya dengan pengepala dengan `<h2>` tag seperti `<h2>Some text</h2>`. Anda boleh hasilkan fungsi suai yang dapat menomborkan pengepala ini dengan nombor seksyen yang berturuan, supaya ia kelihatan seperti `<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'
```

Guna ia dengan ungkapan cari:

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

Letak kursor di bahagian atas fail dan klik *Ganti semua*.

Fungsi ini menggunakan argumen tambahan yang berguna pada `replace()`: argumen `number`. Ketika membuat *Replace All* nombor secara automatik ditokok bagi setiap padanan yang berjaya.

Satu fitur baharu lain yang boleh digunakan ialah penggunaan `replace.file_order` -- menetapkan `'spine'` bermaksud jika gelintar ini dijalankan pada fail HTML berbilang, fail diproses dengan tertib mengikut kemunculannya di dalam buku. Sila rujuk *Pilih tertib fail bila dijalankan pada fail HTML berbilang* (halaman 93) untuk perinciannya.

Auto jana Senarai Kandungan

Akhir sekali, cuba lakukan yang lebih sukar. Anggap buku anda mempunyai pengepala dengan tag `h1` dan `h2` seperi `<h1 id="someid">Some Text</h1>`. Kita akan auto-janakan Senarai Kandungan HTML berdasarkan pengeapala ini. Cipta fungsi suai seperti di bawah:

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

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

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

Guna ia dengan ungkapan cari:

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

Jalankan penggelintaran pada *All text files* dan dipenghujung gelintar, satu tetingkap timbul dengan "Debug output from your function" yang mana akan mempunyai Senarai Kandungan HTML, sedia untuk ditampal ke dalam `toc.html`.

Fungsi di atas boleh diberikan ulasan, supaya ia mudah diikuti. Fitur utama baharu ini ialah penggunaan argumen tambahan lain yang berguna pada fungsi `replace()`, objek `data`. Objek `data` ialah *dic* Python yang berterusan diantara semua invokasi `replace()` yang berjaya semasa satu operasi *Replace All*.

Lain-lain fitur baharu yang digunakan ialah `call_after_last_match` -- menetapkannya `True` pada fungsi `replace()` bermaksud penyunting akan memanggil `replace()` sekali laig selepas semua padanan telah ditemui. Untuk panggilan ekstra ini, objek padanan akan menjadi `None`.

Ini hanyalah demonstrasi untuk menunjukkan kehebatan mod fungsi, jika anda perlu jana Senarai Kandungan dari pengepala dalam buku anda, anda digalakkan guna alat Senarai Kandungan yang disediakan dalam *Alat→Senarai Kandungan*.

API untuk mod fungsi

Semua fungsi mod fungsi mestilah fungsi Python yang bernama replace, dengan tandatangan berikut:

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

Bila cari/ganti dijalankan, bagi setiap padanan yang ditemui, fungsi `replace()` akan dipanggilkan, ia mesti kembalikan rentetan penggantian untuk padanan tersebut. Jika tiada penggantian dibuat, ia seharusnya kembalikan `match.group()` yang mana merupakan rentetan asal. Pelbagai dokumentasi argumen berkenaan fungsi `replace()` seperti di bawah.

Argumen `match`

The `match` argument represents the currently found match. It is a Python Match object. 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.

Argumen `number`

Argumen `number` ialah bilangan padanan semasa. Ketika anda menjalankan *Ganti Semua*, setiap padanan yang berjaya akan menyebabkan `replace()` diganti dengan bilangan menaik. Padanan pertama mempunyai nilai 1.

Argumen `file_name`

Ia merupaakn nama fail bagi fail yang mana padanan semasa ditemui. Ketika menggelintar di dalam teks bertanda, `file_name` adalah kosong. `file_name` adalah dalam bentuk canonical, iaitu laluan berkaitan dengan root buku, menggunakan / sebagai pemisah laluan.

Argumen `metadata`

Ia mewakili data meta bagi sesebuah buku, seperti tajuk, pengarang, bahasa, dll. Ia merupakan objek bagi sesebuah kelas *`calibre.ebooks.metadata.book.base.Metadata`* (halaman 180). Atribut yang berguna termasuklah, `title`, `authors` (senarai pengarang) dan `language` (kod bahasa).

Argumen `dictionaries`

Ia mewakili koleksi kamus yang digunakan untuk penyemakan ejaan buku semasa. Merupakan kaedah paling berguna iaitu `dictionaries.recognized(word)` yang akan kembalikan `True` jika perkataan dikenalpasti oleh kamus bagi bahasa buku semasa.

Argumen `data`

Ia merupakan `dict` Python ringkas. Kitika anda menjalankan *Ganti semua*, setiap padanan berjaya akan menyebabkan `replace()` dipanggil dengan `dict` yang sama sebagai data. Maka anda boleh guna ia untuk menyimpan data arbitrari diantara penyeruan `replace()` semasa operasi *Ganti semua*.

Argumen `functions`

Argumen `functions` memberikan anda capaian ke semua lain-lain fungsi ditakrif pengguna. Ia berguna untuk guna-semula kod. Anda boleh takrifkan fungsi utiliti dari satu tempat dan guna-semula ia dalam fungsi yang lain. Sebagai contoh, anggap anda menghasilkan satu fungsi bernama``My Function`` seperti di bawah:

```
def utility():
   # do something

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

Oleh itu, dalam fungsi lain, anda boleh boleh mencapai fungsi `utility()` seperti ini:

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

Anda juga boleh menggunakan objek fungsi untuk menyimpan data tekal, yang boleh diguna-semula oleh fungsi lain. Sebagai contoh, anda boleh mempunyai satu fungsi yang mana bila dijalankan dengan *Ganti Semua* dapat mengutip beberapa data dan lain-lain fungsi yang menggunakannya jika ia dijalankan selepas itu. Pertimbangkan dua fungsi berikut:

```
# Function One
persistent_data = {}

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,␣
↪**kwargs):
    ...
    persistent_data['something'] = 'some data'

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

Meyahpepijat fungsi anda

Anda boleh menyahpepijat fungsi yang telah dicipta menggunakan fungsi piawai `print()` daripada Python. Output print akan dipaparkan dalam tetingkap timbul selepas Find/replace telah selesai. Didapati terdapat contoh penggunaan `print()` untuk outputkan keseluruhan jadual kandungan di atas.

Pilih tertib fail bila dijalankan pada fail HTML berbilang

Bila anda menjalankan *Ganti semua* pada fail HTML berbilang, tertib fail diproses bergantung pada apakah fail anda telah buka untuk penyuntingan. Anda boleh paksa menggelintar untuk proses fail mengikut tertib yang muncul berdasarkan tetapan atribut `file_order` dalam fungsi anda, seperti berikut:

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

replace.file_order = 'spine'
```

`file_order` menerima dua nilai, `spine` dan `spine-reverse` yang menyebabkan gelintar memproses fail berbilang mengikut tertib muncul di dalam buku, sama ada maju atau mengundur.

Menambah masa tambahan ke dalam fungsi anda selepas padanan terakhir ditemui

Dalam senarai kandungan diauto-jana seperti ciontoh di atas, Lebih berguna anda menambah masa tambahan selepas padanan terakhir ditemui. Anda boleh membuatnya dengan menetapkan atribut `call_after_last_match` ke dalam fungsi anda, seperti di bawah:

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

replace.call_after_last_match = True
```

Menambah output dari fungsi ke teks betanda

Ketika menjalankan gelintar dan ganti pada teks bertanda, kadang kala berguna membuat penambahan di bahagian hujung teksd bertanda. Anda boleh membuatnya dengan menetapkan atribut `append_final_output_to_marked` pada fungsi anda (perhatian anda juga perlu tetapkan `call_after_last_match`), seperti berikut:

```
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
```

Memaksa dialog keputusan ketika membuat penggelintaran pada teks bertanda

Anda juga memaksa dialog keputusan (yang boleh memperlahankan aplikasi menggelintar/mengganti yang berulang pada kebanyakan blok teks) dengan menetapkan atribut `suppress_result_dialog` pada fungsi anda, seperti ini:

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

replace.suppress_result_dialog = True
```

Lagi banyak contoh

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

Snippet

The calibre E-book editor supports *snippets*. A snippet is a piece of text that is either re-used often or contains a lot of redundant text. The editor allows you to insert a snippet with only a few key strokes. For example, suppose you often find yourself inserting link tags when editing HTML files, then you can simply type `<a` in the editor and press `Control+J`. The editor will expand it to:

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

Bukan hanya itu sahaja, perkataan `filename` akan dipilih, dengan kursor berada diatasnya, anda dapat menaip dalam nama fail sebenar, menggunakan fitur penyunting *Auto-lengkap* (halaman 99). Dan selesai menaip nama fail, tekan `Control+J` sekali lagi dan kursor akan lompat ke kedudukan diantara tag `<a>` supaya anda boleh taipkan teks untuk pautan dengan mudah.

Sistem snippet dalam penyunting adalah sangat kompleks, terdapat beberapa snippet terbina-dalam dan anda boleh cipta sendiri untuk memenuhi gaya penyuntingan anda.

Penerangan snippet terbina-dalam berikut dapat mengilustrasikan kehebatan sistem snippet.

Note: Anda juga boleh guna snippet dalam medan masukan teks dalam panel :guilabel:`Gelintar dan Ganti', walaubagaimanapun, pemegang tempat atau placeholder (gunakan `Control+J` sebagai ganti) tidak akan berfungsi.

Snippet terbina-dalam

Snippet terbina-dalam dijelaskan seperti di bawah. Perhatian, anda boleh batalkannya dengan mencipta snippet anda sendiri dengan teks pemicu yang sama.

Menyisip teks pengisi [Lorem]

Snippet terbina-dalam pertama dan paling mudah digunakan untuk menyisip teks pengisi ke dalam dokumen. Teks pengisi diambil dari De finibus bonorum et malorum iaitu hasil kera oleh Cicero (diterjemah ke dalam bahasa Inggeris). Untuk gunakannya hanya taip `Lorem` dalam fail HTML dan tekan `Control+J`. Ia akan diganti dengan beberapa perenggan pengisi.

Takrifan snippet ini adalah sangat mudah, teks pemicu ditakrif sebagai `Lorem` dan templat ditakrif sebagai teks harfiah yang akan disisipkan. Anda boleh suai ia dengan mudah untuk guna bentuk teks pengisi kegemaran anda.

Menyisip tag HTML tertutup-kendiri [<>]

Sekarang cuba lihat contoh mudah konsep *placeholders*. Jika anda mahu menyisip tag tertutup-kendiri `<hr/>`. Hanya taip `<>`, kemudian tekan `Control+J`, penyunting akan kembangkan snippet menjadi:

```
<|/>
```

Di sini, simbol `|` mewakili kedudukan kursor tetikus semasa. Anda boleh taip `hr` kemudian tekan `Control+J` untuk gerak kursor dipenghujung tag. Snippet ini ditakrif sebagai:

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

Pemegang-tempat adalah tanda dollar ($) yang diikuti dengan nombor. Bila snippet dikembangkan dengan menekan `Control+J` kursor diletak pada pemegang-tempat pertama (pepegang-tempat dengan nombor paling rendah). Bila anda tekan `Control+J` sekali lagi kursor lompat ke pemegang-tempat berikutnya (pemegang-tempat dengan nombor lebih tinggi berikutnya).

Menyisip tag pautan HTML [<a]

Tag pautan HTML kesemuanya berkongsi struktur umumnya. Mereka mempunyai atribut `href` dan beberapa teks diantara tag pembuka dengan tag penutup. Snippet menjadikan penaipan mereka lebih efisyen yang mana terdapat beberapa fitur pemegang-tempat. Untuk guna snippet ini, hanya taip `<a` dan tekan `Control+J`. Penyunting akan kembangkan seperti ini:

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

Bukan hanya itu sahaja, perkataan `filename` akan dipilih, dengan kursor berada diatasnya, anda dapat menaip dalam nama fail sebenar, menggunakan fitur penyunting *Auto-lengkap* (halaman 99). Dan selesai menaip nama fail, tekan `Control+J` sekali lagi dan kursor akan lompat ke kedudukan diantara tag `<a>` supaya anda boleh taipkan teks untuk pautan dengan mudah. Selepas anda selesai menaip teks, tekan `Control+J` sekali lagi untuk lompat ke kedudukan selepas tag penutup. Snippet ini ditakrif sebagai:

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

Terdapat beberapa pasangan fitur baharu di sini. Pertama, pemegang-tempat `$1` menjadi lebih kompleks. Ia kini menyertakan beberapa *teks lalai* (perkataan `filename`). Jika pemegang-tempat mengandungi teks lalai, teks lalai adalah pengganti untuk pemegang-tempat jika snippet dikembangkan. Dan juga bila anda lompat ke pemegang-tempat dengan teks lalai menggunakan `Control+J`, teks lalai dipilih. Dengan cara ini, anda boleh guna teks lalai untuk bertindak sebagai peringatan pada anda untuk mengisi bahagian penting templat. Anda juga boleh nyatakan teks lalai untuk pemegang-tempat dengan menggunakan sintaks: `${<number>:default text}`.

Lain-lain fitur baharu ialah pemegang-tempat kedua yang mempunyai asterisk selepas ia (`${2*}`). Ini bermaksud mana-mana teks yang terpilih sebelum pengembangan templat digantikan untuk pemegang-tempat. Untuk melihat tindakan ini, pilih beberapa teks dalam penyunting, tekan `Control+J`, taip `<a` dan tekan `Control+J` sekali lagi, templat akan dikembangkan menjadi:

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

Menyisip tag imej HTML [<i]

Cara yang sama seperti menyisip pautan HTML, sepertimana di atas. Ia membolehkan anda masukkan tag `<img src="filename" alt="description" />` dengan pantas dan lompat diantara atribut `src` dengan `alt`:

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

Menyisip tag HTML arbitari [<<]

Ia membolehkan anda sisip tag HTML arbitari penuh (atau lilit teks terpilih terdahulu di dalam tag). Untuk gunakannya, hanya taip << dan tekan `Control+J`.Penyunting akan kembangkannya menjadi:

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

Taip nama tag, contohnya: `span` dan tekan `Control+J`, hasilnya:

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

Anda akan dapati tagi penutup telah diisi secara automatik dengan `span`. Ia dilakukan dengan satu lagi fitur pemegang-tempat, *mirroring*. Mirroring secara mudah jika anda nyatakan pemegang-tempat sampel lebih dari sekali dalam sesebuah templat, yang kedua dan semua kedudukan kemudian akan diisi secara automatik, bila anda tekan `Control+J`. Takrifan untuk snippet adalah:

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

Sepertimana yang anda lihat, pemegang-tempat pertama ($1) telah dinyatakan sebanyak dua kali, yang kedua dalam tag penutup, yang mana akan menyalin apa jua yang anda taip dalam tag pembukaan.

Menyisip tag HTML arbitari dengan atribut kelas [<c]

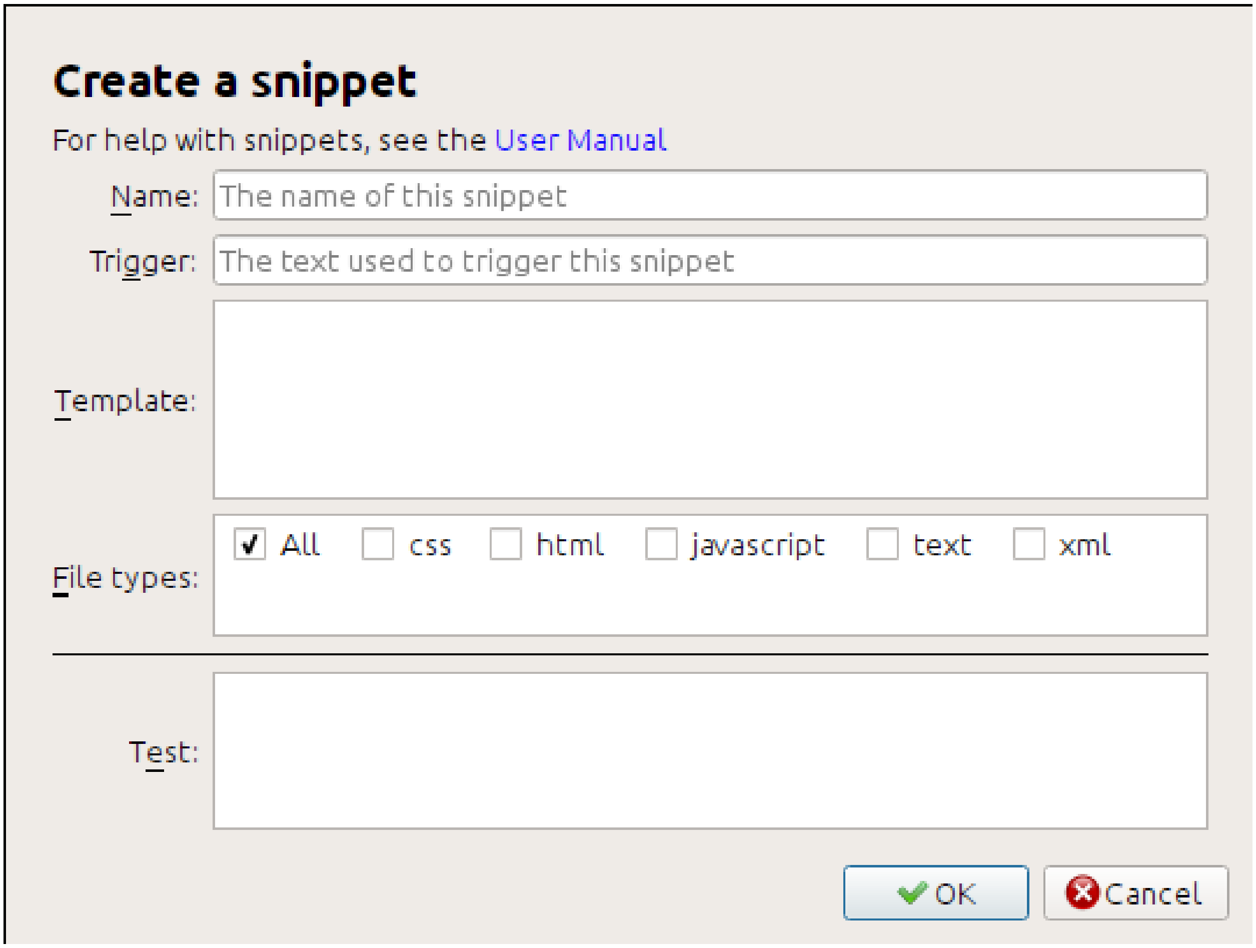
Ia hampir serupa dengan contoh penyisipan tag arbitari di atas, kecuali ia menganggap anda mahu nyatakan kelas untuk tag:

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

Ia membolehkan anda taip dahulu nama tag, kemudian tekan `Control+J`, taip nama kelas, tekan `Control+J` taip kandungan tag dan seterusnya tekan `Control+J` sekali lagi untuk keluar dari tag. Tag penutup akan diisi secara automatik.

Mencipta snippet anda sendiri

Snippet menyenangkan kerana anda boleh cipta sendiri untuk memenuhi gaya penyuntingan anda. Untuk mencipta snippets anda sendiri, pergi ke *Sunting→Keutamaan→Tetapan penyunting→Urus snippet* dalam penyunting. Satu dialog timbul akan bantu anda menghasilkan snippet anda sendiri. Hanya klik pada butang *Tambah snippet* dan anda akan lihat satu dialog seperti berikut:



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:

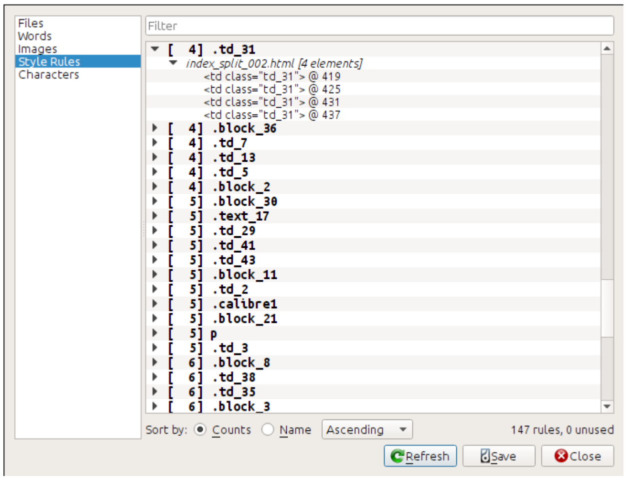
Mula-mula berikan nama snippet anda, yang dapat menjelaskan snippet tersebut. Kemudian nyatakan *trigger*. Trigger atau pemicu merupakan teks yang anda perlu taip dalam penyunting sebelum menekan `Control+J` untuk dapat kembangkan snippet.

Kemudian nyatakan templat snippet. Anda boleh mulakan seperti contoh di atas dan ubahsuai ia mengikut keperluan anda. Akhir sekali, nyatakan jenis fail yang anda mahu snippet aktifkan. Dengan cara ini anda boleh mempunyai snippet berbilang dengan teks pemicu yang sama tetapi dapat berfungsi dengan jenis fail yang berbeza.

Langkah seterusnya adalah menguji snippet yang baru anda cipta. Gunakan kotak *Uji* yang berada dibahagian bawah. Taipkan teks pemicu dan tekan `Control+J` untuk kembangkan snippet dan lompat diantara pemegang-tempat.

5.8.9 The Reports tool

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



```
[  4] .td_31
    index_split_002.html [4 elements]
        <td class="td_31"> @ 419
        <td class="td_31"> @ 425
        <td class="td_31"> @ 431
        <td class="td_31"> @ 437
[  4] .block_36
[  4] .td_7
[  4] .td_13
[  4] .td_5
[  4] .block_2
[  5] .block_30
[  5] .text_17
[  5] .td_29
[  5] .td_41
[  5] .td_43
[  5] .block_11
[  5] .td_2
[  5] .calibre1
[  5] .block_21
[  5] p
[  5] .td_3
[  6] .block_8
[  6] .td_38
[  6] .td_35
[  6] .block_3
```

Sort by: ● Counts ○ Name Ascending   147 rules, 0 unused

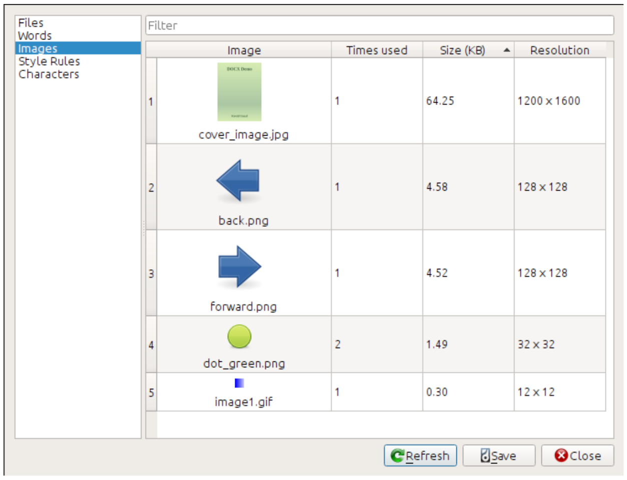


	Image	Times used	Size (KB)	Resolution
1	cover_image.jpg	1	64.25	1200 x 1600
2	back.png	1	4.58	128 x 128
3	forward.png	1	4.52	128 x 128
4	dot_green.png	2	1.49	32 x 32
5	image1.gif	1	0.30	12 x 12

5.9 Special features in the code editor

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

5.9.1 Syntax highlighting

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

- Teks di dalam tag tebal, condong dan pengepala menjadi tebal/condong
- 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`.
- HTML tidak sah disorot dengan garis bawah berwarna merah
- Ralat ejaan dalam teks di dalam tag dan atribut HTML seperti tajuk disorotkan. Penyemak ejaan adalah sedar bahasa, berdasarkan nilai atribut bahasa bagi tag semasa dan keseluruhan bahasa buku.
- CSS terbenam di dalam tag `<style>` disorotkan
- Aksara khas adalah sukar dikenalpasti seperti jarak, pelbagai jenis sempang, dan lain-lain akan disorotkan.
- Pautan ke fail lain dalam tag `<a>` , `<img>` dan `<link>` semuanya mempunyai nama fail tersorot. Jika nama fail yang ia tujukan tidak wujud, nama fail ditanda dengan garis bawah berwarna merah.

5.9.2 Bantuan peka konteks

Anda boleh klik kanan pada nama tag HTML atau nama sifat CSS untuk dapatkan bantuan berkenaan tag atau sifat tersebut.

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.

5.9.3 Auto-lengkap

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 Snippet

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 *Snippet* (halaman **??**).

CHAPTER 6

Pelayan Kandungan 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.

Kandungan

- *Accessing the Content server from other devices* (halaman 102)
  - *Accessing the server from devices on your home network* (halaman 102)
  - *Accessing the server from anywhere on the internet* (halaman 103)
- *The server interface* (halaman 103)
  - *The book list* (halaman 104)
  - *The book reader* (halaman 104)
- *Browser support* (halaman 104)
- *Enabling offline support* (halaman 104)
- *Managing user accounts from the command-line only* (halaman 105)
- *Integrating the calibre Content server into other servers* (halaman 105)
  - *Using a full virtual host* (halaman 105)
  - *Using a URL prefix* (halaman 106)
- *Creating a service for the calibre server on a modern Linux system* (halaman 107)

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.

Note: 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 Accessing the Content server from other devices

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

6.1.1 Accessing the server from devices on your home network

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

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

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

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

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

Trouble-shooting the home network connection

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

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

5. If you have setup a username and password, first try it without that to see if it is causing issues. Some e-ink devices have browsers that do not handle authentication. You can sometimes workaround this by including the username and password in the URL, for example: `http://username:password@192.168.1.2:8080`.

6. If you are stuck, you can always ask for help in the calibre user forums.

6.1.2 Accessing the server from anywhere on the internet

> **Warning:** 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 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 to setup an easy to remember address to use instead of the IP address you found in the first step.

Note: For maximum security, you should also enable HTTPS on the content server. You can either do so directly in the server by providing the path to the HTTPS certificate to use in the advanced configuration options for the server, or you can setup a reverse proxy as described below, to use an existing HTTPS setup.

6.2 The server interface

The server interface is a simplified version of the main calibre interface, optimised for use with touch screens. The home screen shows you books you are currently reading as well as allowing to choose a calibre library you want to browse. The server in calibre gives you access to all your libraries, not just a single one, as before.

6.2.1 The book list

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 The book reader

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

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

6.3 Browser support

The new calibre server makes lots of use of advanced HTML 5 and CSS 3 features. As such it requires an up-to-date browser to use. It has been tested on Android Chrome and iOS Safari as well as Chrome and Firefox on the desktop. It is known not to work with Internet Explorer and Microsoft Edge (hopefully Edge will start working when Microsoft gets around to implementing a few missing standards).

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 the server address.

Note: On iOS, Apple allows only a single browser engine, so Firefox, Chrome and Safari are all actually the same browser under the hood. The new server interface requires iOS 10.3.2 or newer. On Android, the server has been tested with Chrome version 58 and newer.

6.4 Enabling offline support

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

6.5 Managing user accounts from the command-line only

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

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

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

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

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

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

6.6 Integrating the calibre Content server into other servers

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

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

6.6.1 Using a full virtual host

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

```
calibre-server
```

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

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

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

Or, for Apache:

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

<VirtualHost *:80>
    ServerName myserver.example.com
    AllowEncodedSlashes On
```

(continues on next page)

(continued from previous page)

```
    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:

```
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.

Note: 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.

Note: If you have setup SSL for your main server, you should tell the calibre server to use basic authentication instead of digest authentication, as it is faster. To do so, pass the `--auth-mode=basic` option to `calibre-server`.

6.7 Creating a service for the calibre server on a modern Linux system

You can easily create a service to run calibre at boot on a modern (systemd) 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 directory"

[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 directory. Note that it is generally not a good idea to run the server as root. Also change the path to the calibre library directory 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
```

Note: 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.

Note: The calibre server also supports systemd socket activation, so you can use that, if needed, as well.

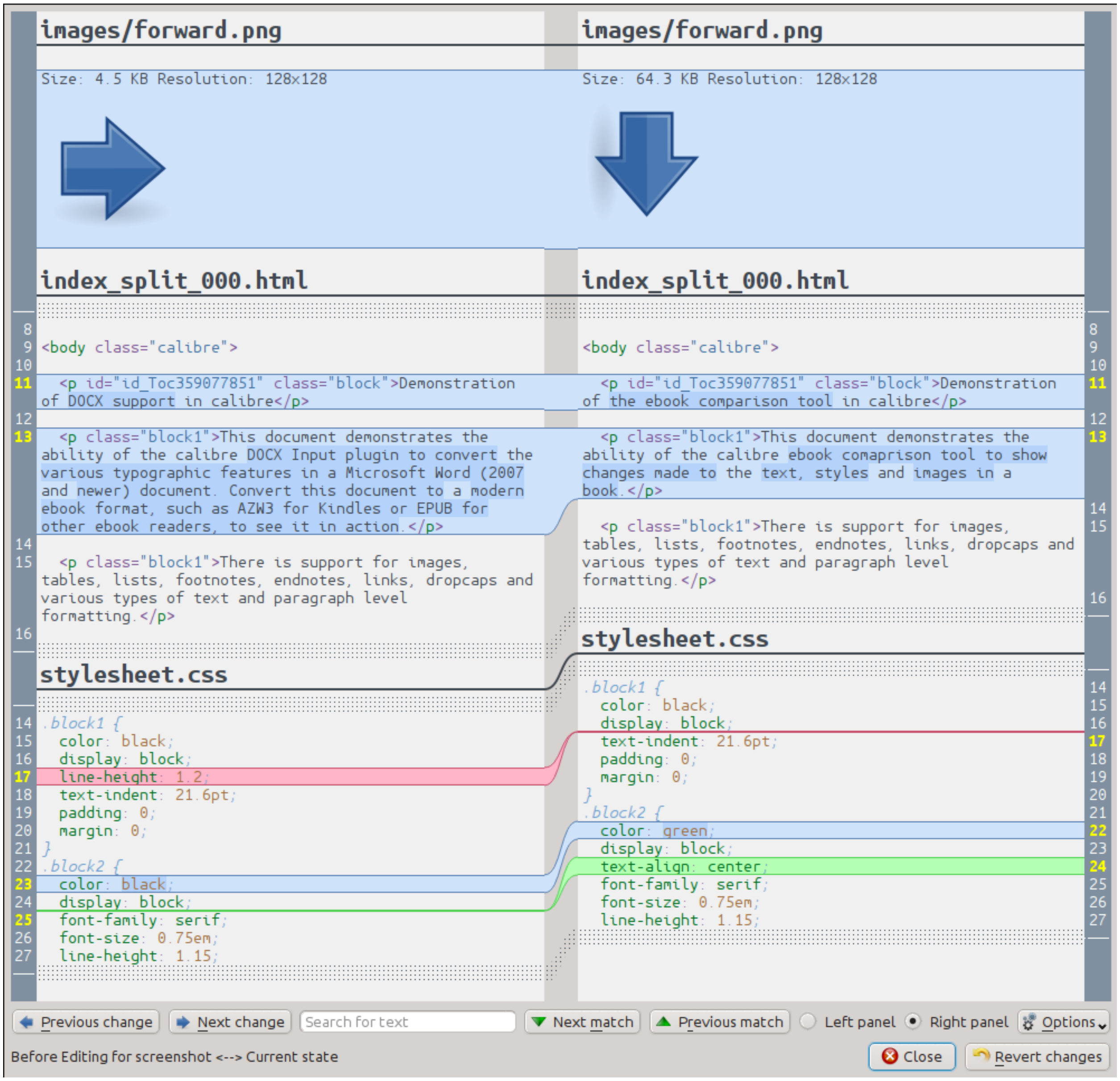
CHAPTER 7

Membandingkan e-buku

calibre menyertakan satu alat pembandingan e-buku bersepadu yang mana boleh digunakan untuk lihat apakah perubahan di dalam e-buku selepas penyuntingan atau pertukaran format. Ia boleh membandingkan buku dalam format EPUB dan AZW3.

Untuk menggunakannya, sama ada buka e-buku dalam alat untuk sunting dan kemudian klik *Fail→Banding dengan buku lain* atau guna panel perincian_buku. Jika anda membuat pertukaran format dari EPUB ke EPUB, fail EPUB asal akan disimpan sebagai ORIGINAL_EPUB. Hanya klik kanan pada masukan ORIGINAL_EPUB di dalam panel Perincian Buku dan pilih *Banding dengan format EPUB*.

Alat pembandingan yang dibuka akan kelihatan seperti dalam cekupan skrin di bawah. Ia menunjukkan perbezaan terhadap teks, gaya dan imej di dalam buku yang dipilih.



images/forward.png

Size: 4.5 KB Resolution: 128x128

index_split_000.html

```
<body class="calibre">

  <p id="id_Toc359077851" class="block">Demonstration
of DOCX support in calibre</p>

  <p class="block1">This document demonstrates the
ability of the calibre DOCX Input plugin to convert the
various typographic features in a Microsoft Word (2007
and newer) document. Convert this document to a modern
ebook format, such as AZW3 for Kindles or EPUB for
other ebook readers, to see it in action.</p>

  <p class="block1">There is support for images,
tables, lists, footnotes, endnotes, links, dropcaps and
various types of text and paragraph level
formatting.</p>
```

stylesheet.css

```
.block1 {
  color: black;
  display: block;
  line-height: 1.2;
  text-indent: 21.6pt;
  padding: 0;
  margin: 0;
}
.block2 {
  color: black;
  display: block;
  font-family: serif;
  font-size: 0.75em;
  line-height: 1.15;
```

images/forward.png

Size: 64.3 KB Resolution: 128x128

index_split_000.html

```
<body class="calibre">

  <p id="id_Toc359077851" class="block">Demonstration
of the ebook comparison tool in calibre</p>

  <p class="block1">This document demonstrates the
ability of the calibre ebook comaprison tool to show
changes made to the text, styles and images in a
book.</p>

  <p class="block1">There is support for images,
tables, lists, footnotes, endnotes, links, dropcaps and
various types of text and paragraph level
formatting.</p>
```

stylesheet.css

```
.block1 {
  color: black;
  display: block;
  text-indent: 21.6pt;
  padding: 0;
  margin: 0;
}
.block2 {
  color: green;
  display: block;
  text-align: center;
  font-family: serif;
  font-size: 0.75em;
  line-height: 1.15;
```

7.1 Memahami paparan perbandingan

Sepertimana yang ditunjukkan pada cekupan skrin di atas, paparan perbandingan menunjukkan perbezaan diantara dua buku secara sebelah-menyebelah. Hanya perbezaan, dengan beberapa baris konteks disekitarnya ditunjukkan. Ia memudahkan apa yang berubah di dalam dokumen bersaiz besar terutamanya buku.

Tambahan teks ditunjukkan dengan latar belakang hijau, teks dibuang dengan latar belakang merah dan teks berubah dengan latar belakang biru.

Bilangan baris bagi semua teks berubah ditunjukkan disisi, memudahkan pergi ke perubahan tersebut di dalam penyunting. Bila anda buka alat pembandingan di dalam penyunting, anda juga boleh dwi-klik pada baris pada panel kanan untuk pergi ke baris tersebut dalam penyunting secara automatik.

Satu teknik berguna bila membuat perbandingan buku adalah dengan memberitahu alat perbandingan untuk mencantikkan fail teks dan gaya sebelum mengira perbezaan. Ia menghasilkan perbezaan yang lebih bersih dan mudah diikuti. Untuk membuatnya, klik butang *Pilihan* di sebelah bawah-kanan dan pilih *Cantikkan fail sebelum membuat perbandingan*. Perhatian proses ini kadangkala menghasilkan kesan yang tidak diingini, seperti penanda menjadi tidak sah dan perlu diubah kembali supaya ia sah semula. Anda juga boleh menukar nombor baris konteks yang menunjukkan perbezaan melalui butang *Pilihan*.

Anda boleh gelintar mana-mana teks dalam perbezaan melalui palang gelintar di bahagian bawah. Anda perlu nyatakan panel yang hendak digelintarkan, *Kiri* atau *Kanan*.

7.2 Melancarkan alat perbandingan

Alat perbandingan sangat berguna jika anda mempunyai dua versi buku yang serupa dan anda mahu lihat perbezaan diantara mereka. Terdapat beberapa cara untuk melancarkan alat ini.

7.2.1 Membandingkan dua fail e-buku

Buka fail pertama dalam alat sunting. Sekarang klik *Fail→Banding dengan buku lain* dan pilih fail kedua (ia mesti dalam format yang sama). Paparan perbandingan akan dibuka dengan fail yang disunting sebelah kanan dan fail kedua sebelah kiri.

7.2.2 Membandingkan ORIGINAL_FMT dengan FMT

Bila anda membuat pertukaran format dalam calibre dari FMT ke format asalnya, fail asal akan disimpan sebagai ORIGINAL_FMT. Anda dapat lihat apa yang telah berubah selepas pertukaran format, dengan mengklik-kanan masukan ORIGINAL_FMT dalam panel perincian_buku pada tetingkap utama calibre dan memilih *Banding dengan FMT*. Paparan perbandingan akan dibuka dengan ORIGINAL_FMT disebelah kiri dan FMT disebelah kanan.

7.2.3 Membandingkan titik semak pada keadaan semasa buku ketika menyunting

Alat sunting mempunyai fitur yang sangat berguna, iaitu titik semak. Ia membolehkan anda simpan keadaan semasa buku dengan nama *checkpoint*, yang mana anda boleh kembali semula jika anda tidak suka perubahan yang dibuat selepas titik semak. Titik semak juga dicipta secara automatik bila anda membuat pelbagai tindakan berautomatik dalam penyunting. Anda boleh lihat senarai titik semak dengan pergi ke *Lihat→Titik Semak* dan kemudian guna butang *Banding* untuk membandingkan buku pada titik semak terpilih dengan keadaan semasa. Alat perbandingan akan menunjukkan titik semak disebelah kiri dan keadaan semasa disebelah kanan.

CHAPTER 8

Menyunting data meta e-buku

Kandungan

- *Menyunting data buku sebuah buku pada satu masa* (halaman 113)
  - *Memuat turun data meta* (halaman 114)
  - *Mengurus format buku* (halaman 114)
  - *Perihal kulit buku* (halaman 114)
- *Menyunting data buku lebih dari sebuah buku pada satu masa* (halaman 114)
  - *Gelintar dan ganti* (halaman 115)
  - *Memuat turun data meta secara pukal* (halaman 116)

E-buku didatangkan dalam semua bentuk dan saiz yang selalunya data metanya (perkara seperti tajuk/pengarang/siri/penerbit) tidak lengkap atau salah. Cara termudah adalah dengan menukar data meta dalam calibre dengan hanya mengklik dua kali pada masukan dan taip pada kedudukan yang betul. Untuk penyuntingan lanjutan, "power editing" gunakan alat data meta yang dijelaskan di bawah.

8.1 Menyunting data buku sebuah buku pada satu masa

Klik pada buku yang mahu disunting dan kemudian klik butang *Sunting data meta* atau tekan kekunci `E`. Satu dialog dibuka yang membolehkan anda sunting semua aspek berkenaan data meta. Ia mempunyai pelbagai fitur yang menjadikan penyuntingan lebih pantas dan lebih efisyen. Senarai tip atau petua yang kerap digunakan:

- Anda boleh klik butang diantara tajuk dengan pengarang untuk menyilih mereka secara automatik.
- Anda boleh klik butang disebelah isih pengarang untuk membolehkan calibre mengiisi secara automatik menggunakan nilai terisih yang tersimpan dalam setiap pengarang. Gunakan dialog *Urus pengarang* untuk lihat dan ubah nilai isih pengarang. Dialog ini boleh dibuka dengan mengklik dan menahan butang disebelah isih pengarang.

- Anda boleh klik butang disebelah tag untuk guna :guilabel:'Penyunting Tag' yang mana ia boleh digunakan untuk mengurus tag yang berkaitan dengan buku tersebut.
- Kotak "Ids" boleh digunakan untuk masukkan ISBN (dan lain-lain jenis id), ia akan berlatar belakang warna merah jika anda masukkan ISBN yang salah. Ia bertukar ke warna hijau jika ISBN yang sah dimasukkan.
- Kotak isih pengarang akan menjadi merah jika nilai isih pengarang berlainan dari apa yang calibre sepatutnya dicadangkan.

8.1.1 Memuat turun data meta

Fitur tebaik bagi dialog sunting data meta ialah keupayaannya mengisi secara automatik kebanyakan medan data dengan mendapatkan data meta dari pelbagai laman sesawang. Buat masa ini, calibre menggunakan isbndb.com, Google Book, Amazon dan Library Thing. Muat turun data meta digunakan untuk mengisi maklumat dalam Tajuk, pengarang, siri, tag, penarafan, keterangan dan ISBN untuk anda.

Untuk memuat turun data meta, isikan medan tajuk dan pengarang dan klik butang *Dapatkan data meta*. calibre akan sediakan kepada anda satu senarai buku yang hampir sepadan dengan tajuk dan pengarang tersebut. Jika anda isi medan ISBN dahulu, ia akan gunakannya dalam keutamaan dengan tajuk dan pengarang. Jika tiada padanan ditemui, cuba perincikan lagi penggelintaran anda dengan menyertakan beberapa kata kunci dalam tajuk dan nama akhir pengarang sahaja.

8.1.2 Mengurus format buku

Dalam calibre, satu masukan buku boleh mempunyai pelbagai jenis *format* berkaitan dengannya. Sebagai contoh, anda boleh dapatkan Complete Works of Shakespeare dalam format EPUB dan kemudian tukar format menjadi MOBI untuk dibaca dengan Kindle anda. calibre secara automatik mengurus pelbagai jenis format untuk anda. Dalam seksyen *Format yang tersedia* bagi dialog Sunting data meta, anda boleh ubah format-format ini. Anda boleh tambah juga format baharu, padam format sedia ada dan juga tanya calibre untuk tetapkan data meta serta kulit buku bagi masukan buku dari salah satu data buku dalam format-format tersebut.

8.1.3 Perihal kulit buku

Anda boleh minta calibre memuat turun kulit buku untuk anda, dengan syarat buku mempunyai ISBN. Selain itu, anda boleh tentukan satu fail dalam komputer anda untuk digunakan sebagai kulit buku. calibre juga boleh menjana kulit buku lalai dengan maklumat data meta asas untuk anda. Anda boleh seret dan lepas imej ke dalam kulit buku untuk mengubahnya dan juga boleh klik-kanan untuk salin/tampal imej kulit buku.

Selain itu, terdapat satu butang yang dapat mengerat sempadan kulit buku secara automatik, jika kulit buku anda mempunyai sempadan yang tidak menarik.

8.2 Menyunting data buku lebih dari sebuah buku pada satu masa

First select the books you want to edit by holding `Ctrl` or `Shift` and clicking on them. If you select more than one book, clicking the *Edit metadata* button will cause the *Bulk* metadata edit dialog to open. Using this dialog, you can quickly set the author/publisher/rating/tags/series etc of a bunch of books to the same value. This is particularly useful if you have just imported a number of books that have some metadata in common. This dialog is very powerful, for example, it has a *Search and replace* tab that you can use to perform bulk operations on metadata and even copy metadata from one column to another.

Dialog sunting data meta biasa ini juga mempunyai butang *Berikutnya* and *Terdahulu* yang mana anda boleh guna untuk menyunting data meta beberapa buah buku secara satu demi satu.

8.2.1 Gelintar dan ganti

Dialog *Sunting data meta pukal* membolehkan anda lakukan operasi gelintar dan ganti secara arbitari pada buku terpilih. Secara lalai ia guna gelintar dan ganti teks mudah, tetapi ia juga menyokong *ungkapan nalar*. Untuk mengetahui berkenaan ungkapan nalar, sila rujuk *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman 184).

As noted above, there are two search and replace modes: character match and regular expression. Character match will look in the *Search field* you choose for the characters you type in the *search for* box and replace those characters with what you type in the *replace with* box. Each occurrence of the search characters in the field will be replaced. For example, assume the field being searched contains *a bad cat*. If you search for *a* to be replaced with *HELLO*, then the result will be *HELLO bHELLOd cHELLOt*.

Jika medan yang anda gelintar pada medan 'berbilang' seperti tag, maka setiap tag diurus secara berasingan. Sebagai contoh, jika tag anda mengandungi perkataan 'Horror, Scary', ungkapan gelintar 'r,' tidak akan dipadankan keran ungkapan yang pertama dilaksanakan pada 'Horro' kemudian pada 'Scary'.

Jika anda mahu proses gelintar mengabaikan perbezaan huruf besar/huruf kecil, buang tanda pada kotak 'Sensitif kata'.

Anda boleh suruh calibre menukar huruf besar/huruf kecil pada hasil gelintar (maklumat selepas penggantian telah berlaku) dengan memilih salah satu fungsi dari kotak 'Laksana fungsi selepas ganti'. Operasi yang tersedia adalah:

- *Hurud kecil* -- ubah semua aksara dalam medan menjadi huruf kecil
- *Huruf besar* -- ubah semua aksara dalam medan menjadi huruf besar
- *Kata tajuk* -- hanya huruf besar pada huruf pertama dalam hasil gelintar.

Kotak *Ujian anda* membolehkan anda masukkan teks untuk memeriksa sama ada gelintar/ganti melakukan proses yang menepati kehendak anda. Dalam kebanyakan kata, kotak uji buku adalah mencukupi, tetapi berkemungkinan terdapat beberapa kata yang mana anda mahu periksa tidak ditunjukkan dalam kotak tersebut. Masukkan kata tersebut ke dalam *Ujian anda*.

Mod ungkapan nalar mempunyai beberapa perbezaan dari mod aksara. Yang pertama fungsi dilaksana pada bahagian rentetan yang sepadan dengan rentetan gelintar, tetapi bukan keseluruhan medan. Kedua adalah fungsi yang dilaksana pada rentetan penggantian, bukan pada keseluruhan medan.

Ketiga dan paling penting ialah rentetan ganti boleh dijadikan rujukan pada bahagian rentetan gelintar dengan menggunakan rujukan belakang atau backreference. Rujukan berlakang ialah `\\n` yang mana n merupakan integer yang dirujuk pada sekumpulan terkurung yang ke-n dalam ungkapan gelintar. Sebagai contoh, berdasarkan contoh yang sama di atas, *a bad cat*, a ungkapan nalar *a (...) (...)*, dan ungkapan ganti *a \2 \1*, hasilnya akan menjadi *a cat bad*. Sila rujuk *Semua berkenaan penggunaan ungkapan nalar di dalam calibre* (halaman 184) lagi berkenaan rujukan belakang.

Satu pola berguna: anggap anda mahu mengubah huruf keseluruhan medan. Cara termudah membuatnya dengan menggunakan mod aksara, buat masa ini kita gunakan mod ungkapan nalar. Ungkapan gelintar seharusnya *(.*)* ungkapan ganti seharusnya *\1*, dan fungsi huruf yang mahu diubah perlu dipilih.

Akhir sekali, dalam mod ungkapan nalar anda boleh salin nilai dari satu medan ke medan yang lain. Hanya jadikan medan sumber dan destinasi berlainan. Salinan boleh gantikan medan destinasi, pra-tambah ke medan (di hadapan), atau tambah ke medan (di hujung). Kotak tanda 'guna koma' memberitahu calibre untuk (atau tidak) menambah tanda koma diantara teks dengan medan destinasi dama mod pra-tambah dan mod tambah. Jika terdapat destinasi berbilang (iaitu, tags), maka anda tidak boleh nyahtanda kotak ini.

Proses gelintar dan ganti selepas selepas semua lain-lain perubahan data meta dalam lain-lain tab dilaksanakan. Ia boleh menyebabkan kekeliruan, kerana kotak uji akan tunjukkan maklumat sebelum perubahan lain, tetapi operasi akan dilaksanakan selepas perubahan yang lain. Jika anda masih ragu berkenaan apa yang telah berlaku, jangan campurkan gelintar/ganti dengan lain-lain perubahan.

8.2.2 Memuat turun data meta secara pukal

Jika anda mahu muat turun data meta untuk buku berbilang pada satu masa, klik-kanan pada butang *Sunting data meta* dan pilih *Muat turun data meta*. Anda boleh pilih untuk muat turun data meta, kulit buku, atau kedua-duanya sekaligus.

CHAPTER 9

Soalan Kerap Ditanya

Kandungan

- *E-book format conversion* (halaman 117)
- *Device integration* (halaman 121)
- *Library Management* (halaman 127)
- *Pelbagai* (halaman 131)

9.1 E-book format conversion

Kandungan

- *Apakah format yang calibre menyokong penukarannya?* (halaman 118)
- *Apakah format sumber terbaik yang boleh ditukarkan?* (halaman 118)
- *Saya telah tukarkan format ke fail PDF, tetapi hasilnya menimbulkan banyak masalah?* (halaman 118)
- *How do I convert my file containing non-English characters, or smart quotes?* (halaman 118)
- *What's the deal with Table of Contents in MOBI files?* (halaman 119)
- *The covers for my MOBI files have stopped showing up in Kindle for PC/Kindle for Android/iPad etc.* (halaman 119)
- *How do I convert a collection of HTML files in a specific order?* (halaman 120)
- *The EPUB I produced with calibre is not valid?* (halaman 120)
- *How do I use some of the advanced features of the conversion tools?* (halaman 121)

9.1.1 Apakah format yang calibre menyokong penukarannya?

calibre menyokong penukaran banyak format input ke lain-lain format output. Ia boleh tukarkan setiap format input dalam senarai berikut, ke setiap format output.

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

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

Note: PRC is a generic format, calibre supports PRC files with TextRead and MOBIBook headers. PDB is also a generic format. calibre supports eReader, Plucker (input only), PML and zTxt PDB files. DJVU support is only for converting DJVU files that contain embedded text. These are typically generated by OCR software. MOBI books can be of two types Mobi6 and KF8. calibre fully supports both. MOBI files often have .azw or .azw3 file extensions. DOCX files from Microsoft Word 2007 and newer are supported.

9.1.2 Apakah format sumber terbaik yang boleh ditukarkan?

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

9.1.3 Saya telah tukarkan format ke fail PDF, tetapi hasilnya menimbulkan banyak masalah?

PDF merupakan format yang tidak sesuai ditukarkan. Senarai di bawah menunjukkan pelbagai masalah yang anda hadapi ketika membuat pertukaran format ke PDF, sila rujuk: *Convert PDF documents* (halaman 62).

9.1.4 How do I convert my file containing non-English characters, or smart quotes?

There are two aspects to this problem:

1. Knowing the encoding of the source file: calibre tries to guess what character encoding your source files use, but often, this is impossible, so you need to tell it what encoding to use. This can be done in the GUI via the *Input character encoding* field in the *Look & feel→Text* section of the conversion dialog. The command-line tools have an `ebook-convert-txt-input --input-encoding` option.

2. When adding HTML files to calibre, you may need to tell calibre what encoding the files are in. To do this go to *Preferences→Advanced→Plugins→File Type plugins* and customize the HTML2Zip plugin, telling it what encoding your HTML files are in. Now when you add HTML files to calibre they will be correctly processed. HTML files from different sources often have different encodings, so you may have to change this setting repeatedly. A common encoding for many files from the web is `cp1252` and I would suggest you try that first. Note that when converting HTML files, leave the input encoding setting mentioned above blank. This is because the HTML2ZIP plugin automatically converts the HTML files to a standard encoding (utf-8).

9.1.5 What's the deal with Table of Contents in MOBI files?

The first thing to realize is that most e-books have two tables of contents. One is the traditional Table of Contents, like the ToC you find in paper books. This Table of Contents is part of the main document flow and can be styled however you like. This ToC is called the *content ToC*.

Then there is the *metadata ToC*. A metadata ToC is a ToC that is not part of the book text and is typically accessed by some special button on a reader. For example, in the calibre E-book viewer, you use the Show Table of Contents button to see this ToC. This ToC cannot be styled by the book creator. How it is represented is up to the viewer program.

In the MOBI format, the situation is a little confused. This is because the MOBI format, alone amongst mainstream e-book formats, *does not* have decent support for a metadata ToC. A MOBI book simulates the presence of a metadata ToC by putting an *extra* content ToC at the end of the book. When you click Goto Table of Contents on your Kindle, it is to this extra content ToC that the Kindle takes you.

Now it might well seem to you that the MOBI book has two identical ToCs. Remember that one is semantically a content ToC and the other is a metadata ToC, even though both might have exactly the same entries and look the same. One can be accessed directly from the Kindle's menus, the other cannot.

When converting to MOBI, calibre detects the *metadata ToC* in the input document and generates an end-of-file ToC in the output MOBI file. You can turn this off by an option in the MOBI Output settings. You can also tell calibre whether to put it and 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).

Note: 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 The covers for my MOBI files have stopped showing up in Kindle for PC/Kindle for Android/iPad etc.

This is caused by a bug in the Amazon software. You can work around it by going to *Preferences→Conversion→Output Options→MOBI output* and setting the *Enable sharing of book content* option. If you are reconverting a previously converted book, you will also have to enable the option in the conversion dialog for that individual book (as per book conversion settings are saved and take precedence).

Note that doing this will mean that the generated MOBI will show up under personal documents instead of Books on the Kindle Fire and Amazon whispersync will not work, but the covers will. It's your choice which functionality is more important to you. I encourage you to contact Amazon and ask them to fix this bug.

The bug in Amazon's software is that when you put a MOBI file on a Kindle, unless the file is marked as a Personal document, Amazon assumes you bought the book from it and tries to download the cover thumbnail for it from its servers. When the download fails, it refuses to fallback to the cover defined in the MOBI file. This is likely deliberate on Amazon's part to try to force authors to sell only through them. In other words, the Kindle only displays covers for books marked as Personal Documents or books bought directly from Amazon.

If you send a MOBI file to an e-ink Kindle with calibre using a USB connection, calibre works around this Amazon bug by uploading a cover thumbnail itself. However, that workaround is only possible when using a USB connection and sending with calibre. Note that if you send using email, Amazon will automatically mark the MOBI file as a Personal Document and the cover will work, but the book will show up in Personal Documents.

9.1.7 How do I convert a collection of HTML files in a specific order?

In order to convert a collection of HTML files in a specific order, you have to create a table of contents file. That is, another HTML file that contains links to all the other files in the desired order. Such a file looks like:

```html
<html>
   <body>
     <h1>Table of Contents</h1>
     <p style="text-indent:0pt">
        <a href="file1.html">First File</a><br/>
        <a href="file2.html">Second File</a><br/>
        .
        .
        .
     </p>
   </body>
</html>
```

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.

Note: 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* and customizing the HTML to ZIP plugin.

9.1.8 The EPUB I produced with calibre is not valid?

calibre does not guarantee that an EPUB produced by it is valid. The only guarantee it makes is that if you feed it valid XHTML 1.1 + CSS 2.1 it will output a valid EPUB. calibre tries hard to ensure that EPUBs it produces actually work as intended on a wide variety of devices, a goal that is incompatible with producing valid EPUBs, and one that is far more important to the vast majority of its users. If you need a tool that always produces valid EPUBs, calibre is not for you. This means, that if you want to send a calibre produced EPUB to an online store that uses an EPUB validity checker, you have to make sure that the EPUB is valid yourself, calibre will not do it for you -- in other words you must feed calibre valid XHTML + CSS as the input documents.

9.1.9 How do I use some of the advanced features of the conversion tools?

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

9.2 Device integration

Kandungan

- *What devices does calibre support?* (halaman 121)
- *How can I help get my device supported in calibre?* (halaman 121)
- *My device is not being detected by calibre?* (halaman 122)
- *My device is non-standard or unusual. What can I do to connect to it?* (halaman 122)
- *How do I use calibre with my iPad/iPhone/iPod touch?* (halaman 123)
- *How do I use calibre with my Android phone/tablet or Kindle Fire HD?* (halaman 123)
- *Can I access my calibre books using the web browser in my Kindle or other reading device?* (halaman 124)
- *I cannot send emails using calibre?* (halaman 125)
- *My device is getting mounted read-only in Linux, so calibre cannot connect to it?* (halaman 125)
- *Why does calibre not support collections on the Kindle or shelves on the Nook?* (halaman 126)
- *I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?* (halaman 126)
- *Covers for books I send to my e-ink Kindle show up momentarily and then are replaced by a generic cover?* (halaman 126)
- *I transferred some books to my Kindle using calibre and they did not show up?* (halaman 127)

9.2.1 What devices does calibre support?

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 How can I help get my device supported in calibre?

If your device appears as a USB disk to the operating system, adding support for it to calibre is very easy. We just need some information from you:

- Complete list of e-book formats that your device supports.
- Is there a special directory on the device in which all e-book files should be placed? Also does the device detect files placed in sub-directories?
- We also need information about your device that calibre will collect automatically. First, if your device supports SD cards, insert them. Then connect your device to the computer. In calibre go to *Preferences→Miscellaneous*

and click the "Debug device detection" button. This will create some debug output. Copy it to a file and repeat the process, this time with your device disconnected from your computer.

- Send both the above outputs to us with the other information and we will write a device driver for your device.

Once you send us the output for a particular operating system, support for the device in that operating system will appear in the next release of calibre. To send us the output, open a bug report and attach the output to it. See calibre bugs.

9.2.3 My device is not being detected by calibre?

Follow these steps to find the problem:

- 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 HD or other Android device, read the note under *Using a USB cable* (halaman 123).
- 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 5.7.2). The latest version can always be downloaded from the calibre website. 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.

9.2.4 My device is non-standard or unusual. What can I do to connect to it?

In addition to the *Connect to folder* function found under the *Connect/share* button, calibre provides a `User defined` device plugin that can be used to connect to any USB device that shows up as a disk drive in your operating system. Note: on Windows, the device must have a drive letter for calibre to use it. See the device plugin `Preferences -> Plugins -> Device plugins -> User defined` and `Preferences -> Miscellaneous -> Get information to setup the user defined device` for more information. Note that if you are using the user defined plugin for a device normally detected by a builtin calibre plugin, you must disable the builtin plugin first, so that your user defined plugin is used instead.

9.2.5 How do I use calibre with my iPad/iPhone/iPod touch?

The easiest way to transfer books wirelessly to your Apple device (iPad/iPhone/iPod) is to use the Calibre Companion iOS app. This app allows calibre to connect to your Apple device wirelessly, just as though you plugged in the device with a USB cable. You can browse files on the device in calibre and use the *Send to device* button to transfer files to your device wirelessly.

Another easy way to browse your calibre collection from your Apple device is by using the calibre Content server, which makes your collection available over the net. First perform the following steps in calibre

- Set the Preferred Output Format in calibre to EPUB (The output format can be set under *Preferences→Interface→Behavior*)
- Set the output profile to iPad (this will work for iPhone/iPods as well), under *Preferences→Conversion→Common options→Page setup*
- Convert the books you want to read on your iDevice to EPUB format by selecting them and clicking the *Convert* button.
- Turn on the Content server by clicking the *Connect/share* button and leave calibre running. You can also tell calibre to automatically start the Content server via *Preferences→Sharing→Sharing over the net*.

The Content server allows you to read books directly in Safari itself. In addition, there are many apps for your iDevice that can connect to the calibre Content server. Examples include: Marvin, Mapleread and iBooks itself.

Using the Content server

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 *Pelayan Kandungan calibre* (halaman **??**) 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.

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

There are two ways that you can connect your Android device to calibre. Using a USB cable -- or wirelessly, over the air. The first step to using an Android device is installing an e-book reading application on it. There are many free and paid e-book reading applications for Android: Some examples (in no particular order): FBReader, Moon+, Mantano, Aldiko, Kindle.

Using a USB cable

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.

Note: 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.

Note: 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.

Over the air

The easiest way to transfer books wirelessly to your Android device is to use the Calibre Companion Android app. This app allows calibre to connect to your Android device wirelessly, just as though you plugged in the device with a USB cable. You can browse files on the device in calibre and use the *Send to device* button to transfer files to your device wirelessly.

calibre also has a builtin web server, the *Content server*. You can browse your calibre collection on your Android device by using the calibre Content server, which makes your collection available over the net. 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 *Pelayan Kandungan calibre* (halaman **??**) 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.

Calibre Companion and many reading apps support browsing the calibre library directly. For example, in Aldiko, click My Catalogs, then + to add a catalog, then give the catalog a title such as "calibre" and provide the URL listed above. You can now browse the calibre library and download directly into the reading software.

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 *Pelayan Kandungan calibre* (halaman **??**) for details.

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

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

9.2.8 I cannot send emails using calibre?

Because of the large amount of spam in email, sending email can be tricky, as different mail servers use different strategies to block email. The most common problem is if you are sending email directly (without a mail relay) in calibre. Many servers (for example, Amazon) block email that does not come from a well known relay. The most robust way to setup email sending in calibre is to do the following:

- Create a free GMX account at GMX.
- 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*)
- 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.

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.

Note: Microsoft/Google/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.

Note: 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. If you have trouble with gmail you will need to allow "less secure" apps as described here.

Note: If you are concerned about giving calibre access to your email account, simply create a new free email account with GMX or Hotmail and use it only for calibre.

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

Linux kernels mount devices read-only when their filesystems have errors. You can repair the filesystem with:

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

Replace /dev/sdc with the path to the device node of your device. You can find the device node of your device, which will always be under /dev by examining the output of:

```
mount
```

9.2.10 Why does calibre not support collections on the Kindle or shelves on the Nook?

Neither the Kindle nor the Nook provide any way to manipulate collections over a USB connection. If you really care about using collections, I would urge you to sell your Kindle/Nook and get a Kobo. Only Kobo seems to understand that life is too short to be entering collections one by one on an e-ink screen :)

Note that in the case of the Kindle, there is a way to manipulate collections via USB, but it requires that the Kindle be rebooted *every time* it is disconnected from the computer, for the changes to the collections to be recognized. As such, it is unlikely that any calibre developers will ever feel motivated enough to support it. There is however, a calibre plugin that allows you to create collections on your Kindle from the calibre metadata. It is available from here.

Note: 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. 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.

Note: If the workaround is not working for you make sure you Kindle firmware is at least version 5.12.5, released in April 2020.

9.2.13 I transferred some books to my Kindle using calibre and they did not show up?

Books sent to the Kindle only show up on the Kindle after they have been *indexed* by the Kindle. This can take some time. If the book still does not show up after some time, then it is likely that the Kindle indexer crashed. Sometimes a particular book can cause the indexer to crash. Unfortunately, Amazon has not provided any way to deduce which book is causing a crash on the Kindle. Your only recourse is to either reset the Kindle, or delete all files from its memory using Windows Explorer (or whatever file manager you use) and then send the books to it again, one by one, until you discover the problem book. Once you have found the problem book, delete it off the Kindle and do a MOBI to MOBI or MOBI to AZW3 conversion in calibre and then send it back. This will most likely take care of the problem.

9.3 Library Management

Kandungan

- *Where are the book files stored?* (halaman 127)
- *How does calibre manage author names and sorting?* (halaman 128)
- *Why doesn't calibre let me store books in my own directory structure?* (halaman 129)
- *Why doesn't calibre have a column for foo?* (halaman 130)
- *Can I have a column showing the formats or the ISBN?* (halaman 130)
- *How do I move my calibre data from one computer to another?* (halaman 130)
- *The list of books in calibre is blank!* (halaman 131)
- *I am getting errors with my calibre library on a networked drive/NAS?* (halaman 131)

9.3.1 Where are the book files stored?

When you first run calibre, it will ask you for a folder in which to store your books. Whenever you add a book to calibre, it will copy the book into that folder. Books in the folder are nicely arranged into sub-folders by Author and Title. Note that the contents of this folder are automatically managed by calibre, **do not** add any files/folders manually to this folder, as they may be automatically deleted. If you want to add a file associated to a particular book, use the top right area of *Edit metadata* dialog to do so. Then, calibre will automatically put that file into the correct folder and move it around when the title/author changes.

Metadata about the books is stored in the file `metadata.db` at the top level of the library folder. This file is a sqlite database. When backing up your library make sure you copy the entire folder and all its sub-folders.

The library folder and all its contents make up what is called a calibre library. You can have multiple such libraries. To manage the libraries, click the calibre icon on the toolbar. You can create new libraries, remove/rename existing ones and switch between libraries easily.

You can copy or move books between different libraries (once you have more than one library setup) by right clicking on a book and selecting the *Copy to library* action.

9.3.2 How does calibre manage author names and sorting?

Author names are complex, especially across cultures, see this note 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 screen and selecting *Manage authors*. Using this dialog you can change the name of an author and also how that name is sorted. This will automatically change the name of the author in all the books of that author. When a book has multiple authors, separate their names using the & character.

Now coming to author name sorting:

- When a new author is added to calibre (this happens whenever a book by a new author is added), calibre automatically computes a sort string for both the book and the author.
- Authors in the Tag browser are sorted by the sort value for the **authors**. Remember that this is different from the Author sort field for a book.
- By default, this sort algorithm assumes that the author name is in `First name Last name` format and generates a `Last name, First name` sort value.
- You can change this algorithm by going to *Preferences→Advanced→Tweaks* and setting the *author_sort_copy_method* tweak.
- You can force calibre to recalculate the author sort values for every author by right clicking on any author and selecting *Manage authors*, then pushing the *Recalculate all author sort values* button. Do this after you have set the author_sort_copy_method tweak to what you want.
- You can force calibre to recalculate the author sort values for all books by using the bulk metadata edit dialog (select all books and click edit metadata, check the *Automatically set author sort* checkbox, then press OK.)
- When recalculating the author sort values for books, calibre uses the author sort values for each individual author. Therefore, ensure that the individual author sort values are correct before recalculating the books' author sort values.
- You can control whether the Tag browser display authors using their names or their sort values by setting the *categories_use_field_for_author_name* tweak in *Preferences→Advanced→Tweaks*

Note that you can set an individual author's sort value to whatever you want using *Manage authors*. This is useful when dealing with names that calibre will not get right, such as complex multi-part names like Miguel de Cervantes Saavedra or when dealing with Asian names like Sun Tzu.

With all this flexibility, it is possible to have calibre manage your author names however you like. For example, one common request is to have calibre display author names LN, FN. To do this, and if the note below does not apply to you, then:

- Set the `author_sort_copy_method` tweak to `copy` as described above.
- Restart calibre. Do not change any book metadata before doing the remaining steps.
- Change all author names to LN, FN using the Manage authors dialog.
- 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.

Note:

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

- Set the `author_sort_copy_method` tweak to `copy` as described above.

- Restart calibre. Do not change any book metadata before doing the remaining steps.
- Open the Manage authors dialog. Press the `copy all author sort values to author` button.
- Check through the authors to be sure you are happy. You can still press Cancel to abandon the changes. Once you press OK, there is no undo.
- Press OK, at which point calibre will change the authors in all your books. This can take a while.

9.3.3 Why doesn't calibre let me store books in my own directory 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 directory 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 wont ever feel the need to hunt through the files on your disk to find a book again. By managing books in its own directory 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 to learn how to create your own columns, or read this blog post.

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 *Bahasa templat calibre* (halaman 147).

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 How do I move my calibre data from one computer to another?

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.

Note: This import/export functionality is only available from calibre version 2.47 onwards. If you have an older version of calibre, or if you encounter problems with the import/export, you can just copy over your calibre library folder manually, as described in the next paragraph.

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

Now on the new computer, start calibre for the first time. It will run the *Welcome wizard* asking you for the location of the calibre library. Point it to the previously copied folder. If the computer you are transferring to already has a calibre installation, then the *Welcome wizard* wont run. In that case, right-click the calibre icon in the toolbar and point it to the newly copied directory. You will now have two calibre libraries on your computer and you can switch between them by clicking the calibre icon on the toolbar. Transferring your library in this manner preserver all your metadata, tags, custom columns, etc.

9.3.7 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 for details.

9.4 Pelbagai

Kandungan

- *I want calibre to download news from my favorite news website.* (halaman 132)
- *Why the name calibre?* (halaman 132)
- *Why does calibre show only some of my fonts on macOS?* (halaman 133)
- *calibre is not starting on Windows?* (halaman 133)
- *calibre freezes/crashes occasionally?* (halaman 134)

- *Using the viewer or doing any conversions results in a permission denied error on Windows* (halaman 134)
- *calibre is not starting/crashing on macOS?* (halaman 135)
- *I downloaded the installer, but it is not working?* (halaman 135)
- *My antivirus program claims calibre is a virus/trojan?* (halaman 135)
- *How do I backup calibre?* (halaman 136)
- *How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?* (halaman 136)
- *I am getting a "Permission Denied" error?* (halaman 136)
- *Can I have the comment metadata show up on my reader?* (halaman 137)
- *How do I get calibre to use my HTTP proxy?* (halaman 137)
- *I want some feature added to calibre. What can I do?* (halaman 137)
- *Why doesn't calibre have an automatic update?* (halaman 137)
- *How is calibre licensed?* (halaman 138)
- *How do I run calibre from my USB stick?* (halaman 138)
- *How do I run parts of calibre like news download and the Content server on my own Linux server?* (halaman 138)

9.4.1 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 *Menambah laman sesawang berita kegemaran anda* (halaman 27).

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

9.4.2 Why the name calibre?

Take your pick:

- Convertor And LIBRary for E-books
- A high *calibre* product
- A tribute to the SONY Librie which was the first e-ink based e-book reader
- My wife chose it ;-)

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.

9.4.3 Why does calibre show only some of my fonts on macOS?

calibre embeds fonts in e-book files it creates. E-book files support embedding only TrueType and OpenType (.ttf and .otf) fonts. Most fonts on macOS systems are in .dfont format, thus they cannot be embedded. calibre shows only TrueType and OpenType fonts found on your system. You can obtain many such fonts on the web. Simply download the .ttf/.otf files and add them to the Library/Fonts directory in your home directory.

9.4.4 calibre is not starting on Windows?

There can be several causes for this:

- If you are on Windows XP, or on a computer with a processor that does not support SSE2 (such as AMD processors from before 2003) try installing calibre version 1.48. calibre 2.0 and newer use Qt 5 which is known to be incompatible with Windows XP machines, and requires SSE2. Simply un-install calibre and then install version 1.48, doing so will not affect your books/settings.
- 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:
  - Uninstall calibre
  - Reboot your computer
  - Re-install calibre. But do not start calibre from the installation wizard.
  - Temporarily disable your antivirus program (disconnect from the Internet before doing so, to be safe)
  - Look inside the folder you chose for your calibre library. If you see a file named metadata.db, delete it.
  - Start calibre
  - From now on you should be able to start calibre normally.
- If you get an error about a Python function terminating unexpectedly after upgrading calibre, first uninstall calibre, then delete the folders (if they exists) `C:\Program Files\Calibre` and `C:\Program Files\Calibre2`. Now re-install and you should be fine.
- If you get an error in the *Welcome wizard* on an initial run of calibre, try choosing a folder like `C:\library` as the calibre library (calibre sometimes has trouble with library locations if the path contains non-English characters, or only numbers, etc.)
- Try running it as Administrator (Right click on the icon and select "Run as Administrator")

If it still wont 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.

9.4.5 calibre freezes/crashes occasionally?

There are several possible things I know of, that can cause this:

- You recently connected an external monitor or TV to your computer. In this case, whenever calibre opens a new window like the edit metadata window or the conversion dialog, it appears on the second monitor where you don't notice it and so you think calibre has frozen. Disconnect your second monitor and restart calibre.
- The following programs have been reported to cause crashes in calibre: If you are running any of these, close them before starting calibre, or uninstall them: *RoboForm*, *Logitech SetPoint Settings*, *Constant Guard Protection by Xfinity*, *Spybot*, *Killer Network Manager*, *Nahimic UI Interface*, *Acronis True Image*.
- You are using a Wacom branded USB mouse/tablet. There is an incompatibility between Wacom drivers and the graphics toolkit calibre uses. Try using a non-Wacom mouse.
- On some 64 bit versions of Windows there are security software/settings that prevent 64-bit calibre from working properly. If you are using the 64-bit version of calibre try switching to the 32-bit version.
- If the crash happens when you are trying to copy text from the calibre E-book viewer, it is most likely caused by some clipboard monitoring/managing application you have running. Turn it off and you should be fine.
- If the crashes happen specifically when you are using a file dialog, like clicking on the *Add books* button or the *Save to Disk button*, then you have some software that has installed broken Shell extensions on your computer. Known culprits include: *SpiderOak*, *odrive sync* and *Dell Backup and Recovery* and *NetDrive*. If you have one of these, uninstall them and you will be fine. You can also use the NirSoft Shell Extension Viewer to see what shell extensions are installed on your system and disable them individually, if you don't want to uninstall the full program. Remember to use "Restart Explorer" or reboot your computer after disabling the shell extensions.

If none of the above apply to you, then there is some other program on your computer that is interfering with calibre. First reboot your computer in safe mode, to have as few running programs as possible, and see if the crashes still happen. If they do not, then you know it is some program causing the problem. The most likely such culprit is a program that modifies other programs' behavior, such as an antivirus, a device driver, something like RoboForm (an automatic form filling app) or an assistive technology like Voice Control or a Screen Reader.

The only way to find the culprit is to eliminate the programs one by one and see which one is causing the issue. Basically, stop a program, run calibre, check for crashes. If they still happen, stop another program and repeat.

9.4.6 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.

9.4.7 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. If that doesn't solve it, look for a corrupted font file on your system, in ~/Library/Fonts or the like. An easy way to check for corrupted fonts in macOS is to start the "Font Book" application, select all fonts and then in the File menu, choose "Validate fonts".

9.4.8 I downloaded the installer, but it is not working?

Downloading from the Internet can sometimes result in a corrupted download. If the calibre installer you downloaded is not opening, try downloading it again. If re-downloading it does not work, download it from an alternate location. 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 and you have Microsoft PowerToys running, quit it.
- Try rebooting your computer and running a registry cleaner like Wise registry cleaner.
- 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 Internet Explorer, try using Firefox or Chrome instead.
- If you get an error about a missing DLL on Windows, then most likely, the permissions on your temporary folder are incorrect. Go to the folder `C:\Users\USERNAME\AppData\Local` in Windows explorer and then right click on the `Temp` folder and select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

If you still cannot get the installer to work and you are on Windows, you can use the calibre portable install, which does not need an installer (it is just a zip file).

9.4.9 My antivirus program claims calibre is a virus/trojan?

The first thing to check is that you are downloading calibre from the official website: https://calibre-ebook.com/download. Make sure you are clicking the download links on the left, not the advertisements on the right. calibre is a very popular program and unscrupulous people try to setup websites offering it for download to fool the unwary.

If you have the official download and your antivirus program is still claiming calibre is a virus, then, your antivirus program is wrong. Antivirus programs use heuristics, patterns of code that "look suspicious" to detect viruses. It's rather like racial profiling. calibre is a completely open source product. You can actually browse the source code yourself (or hire someone to do it for you) to verify that it is not a virus. Please report the false identification to whatever company you buy your antivirus software from. If the antivirus program is preventing you from downloading/installing calibre, disable it temporarily, install calibre and then re-enable it.

9.4.10 How do I backup calibre?

The most important thing to backup is the calibre library folder, that contains all your books and metadata. This is the folder you chose for your calibre library when you ran calibre for the first time. You can get the path to the library folder by clicking the calibre icon on the main toolbar. You must backup this complete folder with all its files and sub-folders.

You can switch calibre to using a backed up library folder by simply clicking the calibre icon on the toolbar and choosing your backup library folder. A backed up library folder backs up your custom columns and saved searches as well as all your books and metadata.

If you want to backup the calibre configuration/plugins, you have to backup the config directory. You can find this config directory via *Preferences→Miscellaneous*. Note that restoring configuration directories is not officially supported, but should work in most cases. Just copy the contents of the backup directory into the current configuration directory to restore.

9.4.11 How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?

Most purchased EPUB books have *DRM* (halaman **??**). 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.

9.4.12 I am getting a "Permission Denied" error?

A permission denied error can occur because of many possible reasons, none of them having anything to do with calibre.

- You can get permission denied errors if you are using an SD card with write protect enabled.
- On macOS if you get permission errors when connecting a device to calibre, you can fix that by looking under *System Preferences* > *Security and Privacy* > *Privacy* > *Files and Folders*.
- If you, or some program you used changed the file permissions of the files in question to read only.
- If there is a filesystem error on the device which caused your operating system to mount the filesystem in read only mode or mark a particular file as read only pending recovery.
- If the files have their owner set to a user other than you.
- If your file is open in another program.
- If the file resides on a device, you may have reached the limit of a maximum of 256 files in the root of the device. In this case you need to reformat the device/sd card referred to in the error message with a FAT32 filesystem, or delete some files from the SD card/device memory.

You will need to fix the underlying cause of the permissions error before resuming to use calibre. Read the error message carefully, see what file it points to and fix the permissions on that file or its containing folders.

9.4.13 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.

9.4.14 How do I get calibre to use my HTTP proxy?

By default, calibre uses whatever proxy settings are set in your OS. Sometimes these are incorrect, for example, on Windows if you don't use Internet Explorer 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.

9.4.15 I want some feature added to calibre. What can I do?

You have two choices:

1. Create a patch by hacking on calibre and send it to me for review and inclusion. See Development.
2. Open a bug requesting the feature . 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 *Menulis pemalam anda sendiri untuk menambahbaik kefungsian calibre* (halaman 194).

9.4.16 Why doesn't calibre have an automatic update?

For many reasons:

- *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. Implementing automatic updates would greatly increase that and end up costing thousands of dollars a month, which someone has to pay.
- If I implement a dialog that downloads the update and launches it, instead of going to the website as it does now, that would save the most ardent calibre updater, *at most five clicks a week*. There are far higher priority things to do in calibre development.
- If you really, really hate downloading calibre every week but still want to be up to the latest, I encourage you to run from source, which makes updating trivial. Instructions are *available here* (halaman 305).
- There are third party automatic updaters for calibre made by calibre users in the calibre forum.

9.4.17 How is calibre licensed?

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. 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.

9.4.18 How do I run calibre from my USB stick?

A portable version of calibre is available here.

9.4.19 How do I run parts of calibre like news download and the Content server on my own Linux server?

First, you must install calibre onto your Linux server. If your server is using a modern Linux distribution, you should have no problems installing calibre onto it.

Note: calibre needs GLIBC >= 2.18 and libstdc++ >= 6.0.21. 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.

You can run the calibre server via the command:

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

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

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

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

You can email downloaded news with the command:

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

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

Finally, you can add downloaded news to the calibre library with:

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

Remember to read the *Antaramuka Baris Perintah* (halaman 267) section of the calibre User Manual to learn more about these, and other commands.

CHAPTER 10

Tutorial

Di sini anda boleh mencari tutorial memulakan calibre dengan menggunakan fitur lanjutan, seperti XPath dan templat.

10.1 Mengurus sub-kumpulan buku, sebagai contoh "genre"

Sesetengah orang mahu mengurus buku dalam pustaka mereka menjadi subkumpulan, serupa dengan subfolder. Sebab paling umum adalah untuk mewujudkan hieraki genre, tetapi ada juga sebab-sebab lain. Seorang pengguna mahu mengurus buku teks mengikut subjek dan kod kursus. Yang lain pula mahu kekalkan jejak hadiah mengikut subjek dan penerima. Tutorial ini akan menggunakan genre sebagai contoh untuk baki pos ini.

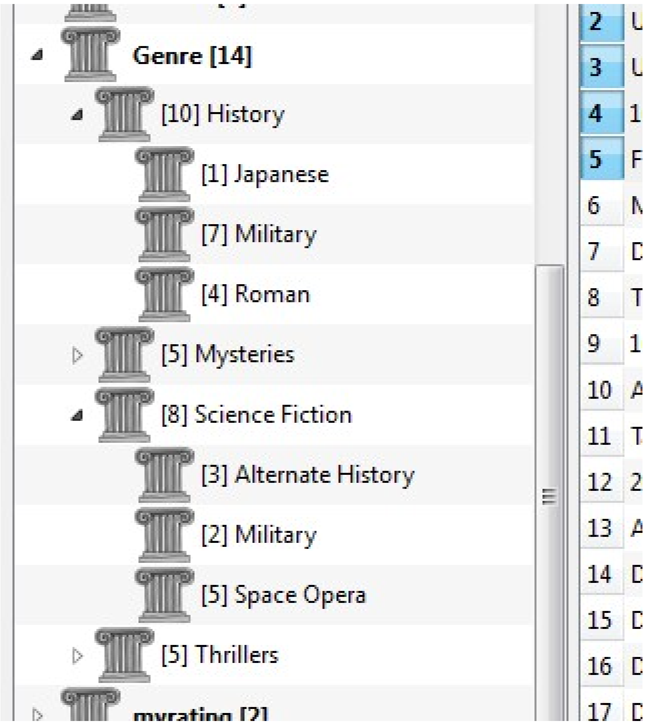
Sebelum itu, ambil perhatian kita tidak membincangkan berkenaan folder di dalam cakera keras. Subkumpulan bukanlah folder fail. Buku tidak akan disalin di tempat lain. Struktur fail pustaka calibre tidak juga terlibat. Sebaliknya, kita akan persembahkan satu cara mengurus dan paparkan subkumpulan buku di dalam pustak calibre itu sendiri.

- *Persediaan* (halaman 141)
- *Menggelintar* (halaman 143)
- *Sekatan* (halaman 144)
- *Fungsi templat yang berguna* (halaman 144)

Keperluan yang umumnya disediakan untuk sub-kumpulan seperti genre adalah:

- Sub-kumpulan (seperti genre) mesti mengandungi (menuju ke) buku, bukan kategori buku. Inilah yang membezakan sub-kumpulan dari kategori pengguna calibre.
- Sebuah buku boleh berada dalam sub-kumpulan berbilang (genre). Ini juga membezakan sub-kumpulan dari folder fail fizikal.
- Sub-kumpulan (genre) mesti membentuk satu hieraki; sub-kumpulan boleh mengandungi beberapa sub-kumpulan.

Tag memberikan dua pengelompokan pertama. Jika anda tag sebuah buku dengan genre maka anda boleh gunakan pelayar Tag (atau gelintar) untuk mencari buku dengan genre tersebut iaitu yang pertama. Kebanyakan buku juga mempunyai tag(s) yang serupa iaitu pengelompokan yang kedua. Masalahnya tag tidak dapat memenuhi keperluan yang ketiga. Iaitu ia tidak menyediakan hieraki.



Fitur hieraki calibre memberikan pengelompokan yang ketiga, keupayaan melihat genre dalam bentuk 'pepohon' dan keupayaan untuk menggelintar buku dengan mudah dalam genre atau sub-genre. Sebagai contoh, anggap struktur genre anda seperti berikut:

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

Dengan menggunakan fitur hieraki, anda boleh lihat genre ini di dalam pelayar Tag dalam bentuk pepohon, seperti yang dipapar dalam imej skrin. Dalam contoh ini aras paling terluar (Genre) ialah lajur suai yang mengandungi genre. Genre yang mengandungi sub-genru muncul dengan simbol segitga kecil sebelahnya. Bila mengklik segitiga tersebut, akan dibuka item dan dipaparkan sub-genre yang dikandunginya, seperti yang anda lihat pada History dan Science Fiction.

Dengan mengklik pada genre dapat menggelintar semua buku dalam genre tersebut atau anak bagi genre tersebut. Sebagai contoh, mengklik Science Fiction memaparkan semua ketiga-tiga genre anak iaitu Alternate History, Military, dan Space Opera. Jika mengklik pada Alternate History akan memaparkan buku dalam genre tersebut, tetapi mengabaikan kedua-dua subgenre Military dan Space Opera. Melainkan, ada buku yang mempunyai genre berbilang. Jika buku tersebut mempunyai kedua-dua genre Space Opera dan Military, maka anda akan dapat melihat buku tersebut jika mengklik pada mana-mana genre. Perihal penggelintaran lebih lanjut di bawah.

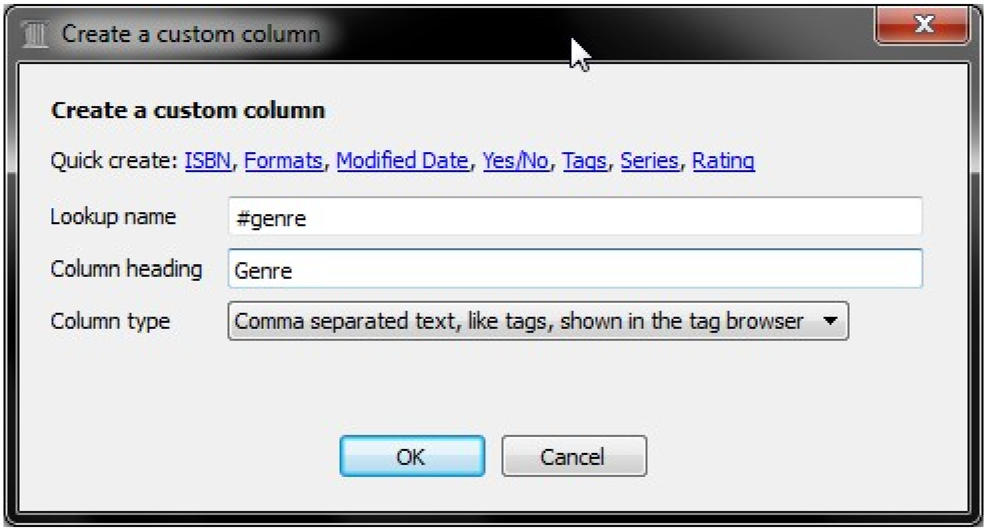
Lain-lain perkara yang anda dapati dari imej ialah genre Military muncul sebanyak dua kali, sekali di bawah History dan lagi sekali di bawah Science Fiction. Ini kerana, genre berada dalam bentuk hieraki, walaupun kedua-duanya di bawah genre berlainan. Satu buku boleh berada dalam satu tempat, tempat yang lain, atau (bagi kes ini) kedua-duanya. Sebagai contoh, buku Winston Churchill iaitu "The Second World War" boleh berada dalam genre "History.Military". Manakala, buku David Weber iaitu Honor Harrington boleh ditempatkan dalam genre "Science Fiction.Military", dan juga boleh berada di dalam genre "Science Fiction.Space Opera."

Sebaik sahaja satu genre wujud, sekurang-kurangnya satu buku berada dalam genre tersebut, anda boleh laksanakannya dengan mudah kepada lain-lain buku dengan menyeret buku dalam paparan pustaka ke dalam genre yang anda mahu buku tersebut berada. Anda juga boleh laksanakan genre dalam penyunting data meta; maklumat lebih lanjut di bawah.

10.1.1 Persediaan

Mulai sekarang, persoalan anda mungkin "Bagaimanakah ia boleh dilakukan?" Terdapat tiga langkah iaitu: 1) cipta lajur suai. 2) beritahu calibre lajur baharu dianggap sebagai hieraki, dan 3) tambah genre yang dikehendaki.

Anda boleh mencipta lajur suai dengan cara berikut, melalui Keutamaan -> Tambah lajur anda sendiri. Contoh ini menggunakan "#genre" sebagai nama carian dan "Genre" sebagai pengepala lajur. Jenis lajur ialah "Teks dipisan-tanda-koma, seperti tag, ditunjukkan dalam pelayar Tag."

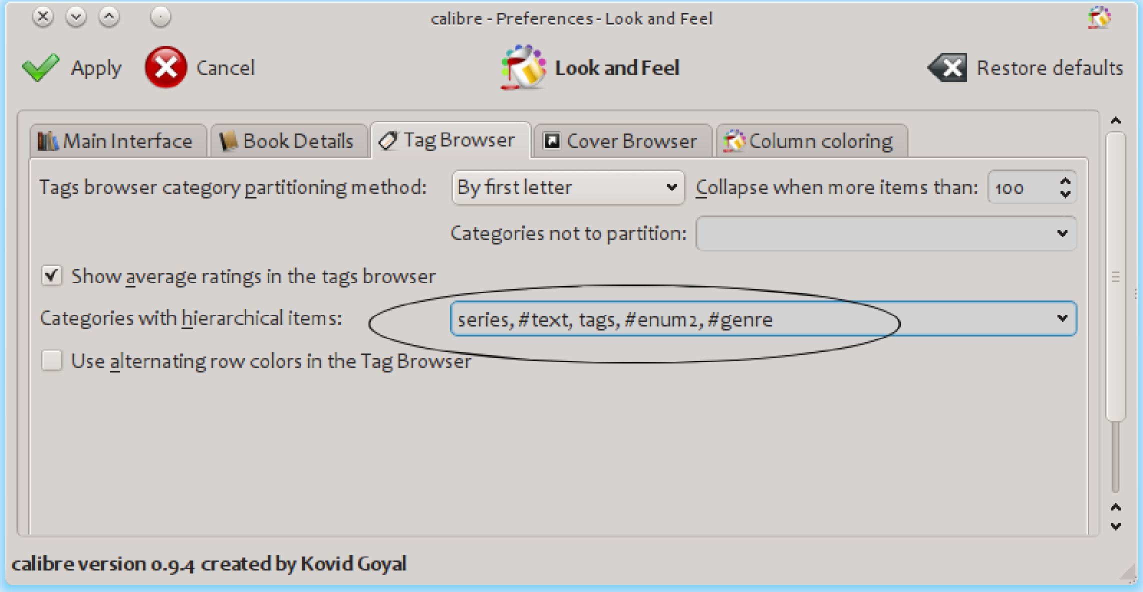

Create a custom column
Create a custom column
Quick create: ISBN, Formats, Modified Date, Yes/No, Tags, Series, Rating
OK
Cancel

Lookup name: #genre

Column heading: Genre

Column type: Comma separated text, like tags, shown in the tag browser

Kemudian selepas memulakan semula calibre, anda mesti beritahu calibre bahawa lajur tersebut dianggap sebagai hieraki. Pergi ke Keutamaan -> Penampilan -> Pelayar Tag dan masukkan nama carian "#genre" ke dalam kotak "Kategori dengan item berhieraki". Tekan *Laksana*, dan tetapan selesai.



calibre - Preferences - Look and Feel

Tags browser category partitioning method: By first letter

Collapse when more items than: 100

Categories not to partition:

☑ Show average ratings in the tags browser

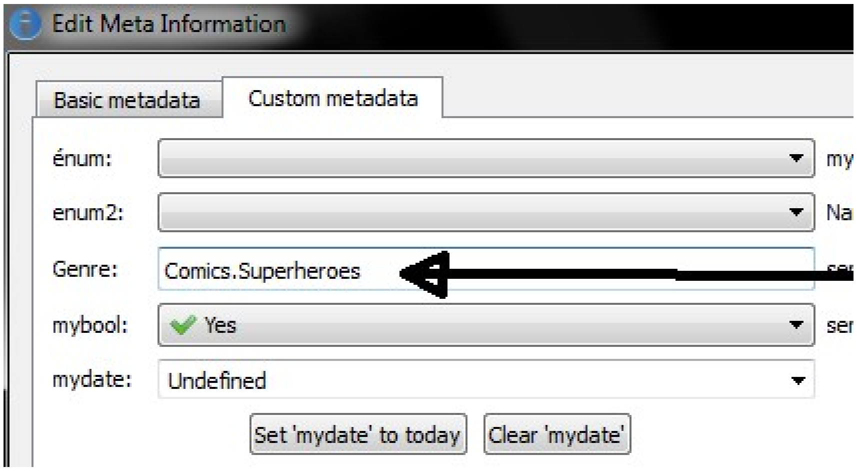
Categories with hierarchical items: series, #text, tags, #enum2, #genre

☐ Use alternating row colors in the Tag Browser

calibre version 0.9.4 created by Kovid Goyal

Pada ketika ini tiada lagi genre di dalam lajur. Kita masih tertinggal langkah terakhir: bagaimana hendak terapkan genre ke dalam sebuah buku. Satu genre tidak akan wujud sehinggalah ia muncul sekurang-kurangnya pada sebuah buku. Untuk mengetahui bagaimana hendak terapkan satu genre buat kali pertama, kita mesti lihat dahulu secara terperinci apakah genre yang ada di dalam data meta sebuah buku.

Hieraki sesuatu 'perkara' dibina dengan mewujudkan item yang mengandungi frasa yang dipisah dengan tanda titik. Berdasarkan contoh sebelum ini, item-item tersebut adalah "History.Military", "Mysteries.Vampire", "Science Fiction.Space Opera", dan lain-lain. Seterusnya untuk mencipta genre baharu, anda boleh pilih buku yang mempunyai genre tersebut, sunting data metanya, dan masukkan genre baharu ke dalam lajur yang anda telah cipta. Berdasarkan contoh terdahulu, jika anda mahu umpuk genre baharu "Comic" dengan sub-genre "Superheroes" pada sebuah buku, anda patut 'sunting data meta' buku (komik) tersebut, pilih tab data meta, dan kemudian masukkan "Comics.Superheroes" yang ditunjukkan seperti berikut (abaikan lain-lain lajur suai):



Edit Meta Information

énum:

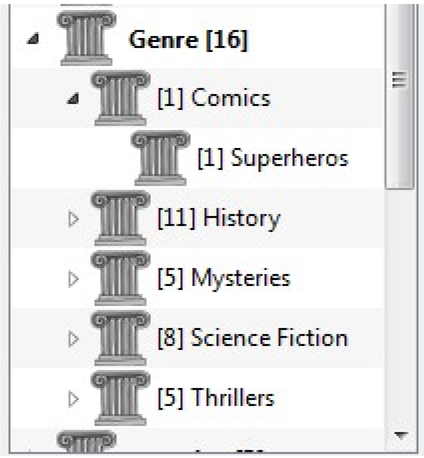
enum2:

Genre: Comics.Superheroes

mybool: Yes

mydate: Undefined

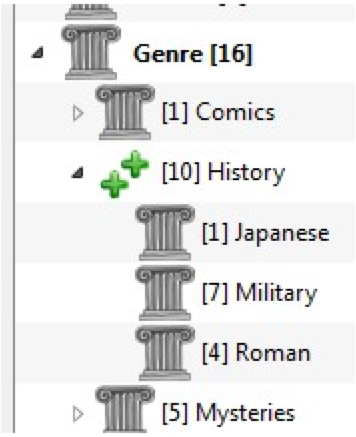
Selepas melakukan perkara di atas, anda lihat seperti dalam pelayar Tag:



Seterusnya dari sini, untuk laksanakan genre baharu pada sebuah buku (anggaplah buku komik), anda sama ada seret buku ke dalam genre, atau tambah ia ke dalam buku menggunakan sunting data meta sepertimana yang dilakukan seperti di atas.

Note: Paparan berhieraki hanya berfungsi jika pelayar Tag ditetapkan untuk mengisih item mengikut nama. Ini adalah lalai dan boleh diperiksa dengan mengklik butang *Konfigur*ădi bahagian bawah pelayar Tag.

10.1.2 Menggelintar

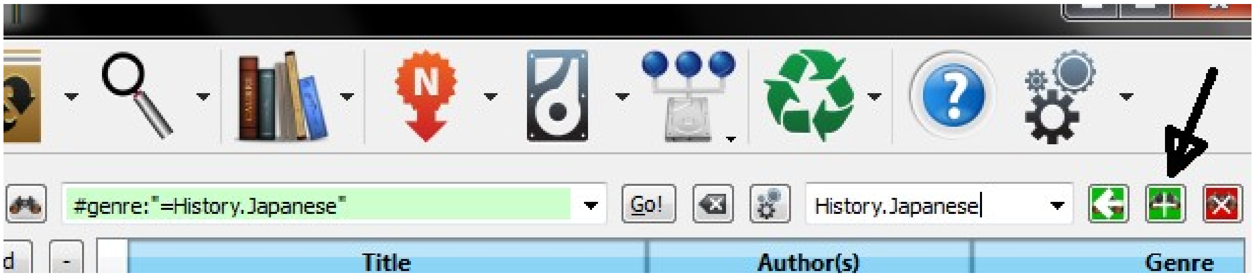


Cara termudah menggelintar genre ialah menerusi pelayar Tag, dengan mengklik genre yang anda mahu cari. Klik pada genre dengan anak yang akan tunjukkan buku dengan genre tersebut dan semua genre anaknya. Walaubagaimanapun, ia menimbulkan satu persoalan. Walaupun genre mempunyai anak tidak bermaksud ia bukanlah genrenya sendiri. Sebagai contoh, sebuah buku boleh mempunyai genre "History" tetapi bukannya "History.Military". Bagaimana caranya anda menggelintar buku dengan genre "History" sahaja?

Mekanisma gelintar pelayar Tag tahu jika sesebuah item mempunyai anak. Jika ia ada, dengan mengklik pada item berkitar 5 gelintar selain dari kebiasaannya tiga. Yang pertama ialah tanda tambah hijau biasa, yang menunjukkan pada anda buku dengan genre itu sahaja (contohnya History). Kedua ialah tanda tambah berganda dua (ditunjukkan seperti di atas), yang menunjukkan pada anda buku dengan genre tersebut dan juga semua sub-genre (contohnya History dan History.Military). Yang ketiga ialah tanda tolak merah normal, yang menunjukkan pada anda buku tanpa genre secara tepat. Keempat pula ialah tanda tolak berganda dua, yang menunjukkan pada anda buku tanpa genre tersebut atau sub-genrenya. Akhirnya, kembali ke permulaan, tanpa tanda, bermaksud tiada gelintar.

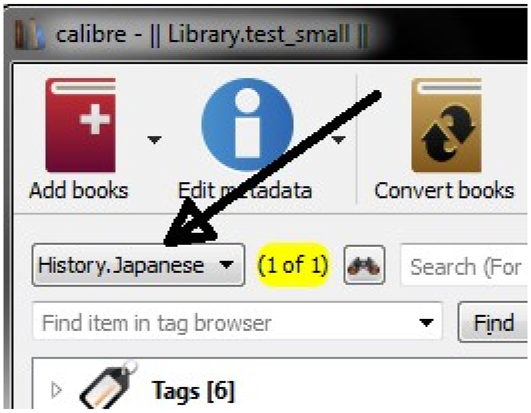
10.1.3 Sekatan

If you search for a genre then create a saved search for it, you can use the 'restrict to' box to create a Virtual library of books with that genre. This is useful if you want to do other searches within the genre or to manage/update metadata for books in the genre. Continuing our example, you can create a Saved search named 'History.Japanese' by first clicking on the genre Japanese in the Tag browser to get a search into the search field, entering History.Japanese into the saved search field, then pushing the "Save search" button (the green box with the white plus, on the right-hand side).


#genre:"=History.Japanese"

History.Japanese

Selepas mencipta gelintar tersimpan, anda boleh gunakannya sebagai sekatan.


calibre - || Library.test_small ||

History.Japanese

10.1.4 Fungsi templat yang berguna

Anda mungkin mahu guna maklumat genre di dalam templat, seperti simpan ke dalam cakera atau hantar ke dalam peranti. Persoalannya, "Bagaimana saya boleh dapatkan nama genre atau nama terluar?" Satu fungsi templat calibre, subitem, menjadikan kerja ini lebih mudah.

Sebagai contoh, anggap anda mahu tambah aras genre paling luar ke templat save-to-disk untuk membuat folder genre, seperti "History/The Gathering Storm - Churchill, Winston". Langkahnya, anda mesti ekstrak aras pertama hieraki dan tambah ia dihadapan bersama-sama dengan tanda slash atau miring untuk menunjukkan ia adalah folder. Templat di bawah menunjukkan bagaimana ia telah siap:

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

Sila ruju *Bahasa templat* (halaman 147) untuk maklumat lanjut berkenaan templat dan fungsi `subitems()`.

10.2 Tutorial XPath

Dalam tutorial ini, anda diperkenalkan dengan bahasa XPath, iaitu bahasa kueri atau pertanyaan yang boleh digunakan untuk memilih bahagian arbitrari HTML dokumen di dalam calibre. XPath merupakan piawaian yang banyak digunakan, dan jika anda mencari di dalam google akan dapati sangat banyak maklumat mengenai. Walaubagaimanapun, tutorial ini lebih fokuskan pada penggunaan XPath untuk tugasan berkaitan e-buku seperti mencari pengepala bab di dalam dokumen HTML tidak berstruktur.

Kandungan

- *Memilih berdasarkan nama tag* (halaman 145)
- *Memilih berdasarkan atribut* (halaman 146)
- *Memilih berdasarkan kandungan tag* (halaman 146)
- *E-buku sampel* (halaman 146)
- *Fungsi terbina-dalam XPath* (halaman 147)

10.2.1 Memilih berdasarkan nama tag

Bentuk paling ringkas pemilihan ialah dengan memilih tag mengikut nama. Contohnya, anda mahu memilih semua tag `<h2>` di dalam dokumen. Pertanyaan XPath untuk ini hanyalah simply:

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

Awalan // bermaksud *gelintar pada mana-mana aras dokumen*. Sekarang anda mahu gelintar tag `<span>` yang berada di dalam tag `<a>` tags. Ia boleh dibuat dengan:

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

Jika anda mahu gelintar tag pada aras tertentu di dalam dokumen, ubah prefix:

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

Ini hanya padankan `<p>A very short ebook to demonstrate the use of XPath.</p>` di dalam *E-buku sampel* (halaman 146) tetapi bukan pada mana-mana tag `<p>` yang lain. Awalan `h:` dalam contoh di atas diperlukan untuk memadani tag XHTML. Ini kerana secara dalaman, calibre mewakili semua kandungan sebagai XHTML. Dalam tag XHTML yang mempunyai *namespace*, dan `h:` merupakan awalan ruang nama untuk tag HTML.

Sekarang anda mahu memilih kedua-dua tag `<h1>` dan `<h2>`. Untuk membuatnya,m kita perlukan konstruk XPath yang dikenali sebagai *predikat*. *Predikat* :dfn: ialah ujian yang digunakan untuk memilih tag. Ujian boleh jadi berguna sepanjang tutorial ini dijalanakn, anda akan lihat lagu beberapa contoh yang berguna. Predikat dicipta dengan menutup ungkapan ujian dengan simbol kurungan:

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

Terdapat beberapa fitur baharu di dalam ungkapan XPath ini. Yang pertama ialah penggunaan kad liar `*`. Ia bermaksud *padan mana-mana tag*. Sekarang lihat pada ungkapan uji *name()='h1' atau name()='h2'*`. *name()* ialah contoh *fungsi terbina-dalam*. Ia hanyalah menilai nama tag. Oleh itu dengan menggunakannya, kita boleh memilih tag yang namanya sama ada *h1* atau *h2*. Perhatian, fungsi *name()* mengabaikan ruang nama supaya tidak memerlukan awalan `h:`. XPath mempunyai beberapa fungsi terbina-dalam yang berguna. Ada beberapa lagi fungsi akan diperkenalkan di dalam tutorial ini.

10.2.2 Memilih berdasarkan atribut

Untuk memilih tag berdasarkan atribut mereka, penggunaan predikat diperlukan:

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

Di sini, operator @ merujuk pada atribut tag. Anda boleh guna beberapa *fungsi terbina-dalam XPath* (halaman 147) untuk membuat padanan lebih kompleks pada nilai atribut.

10.2.3 Memilih berdasarkan kandungan tag

Dengan menggunakan XPath, anda boleh pilih tag berdasarkan pada teks yang dikandunginnya. Cara terbaik membuatnya ialah denga gunakan kehebatan *ungkapan nalar* melalui fungsi terbina-dalam *re:test()*:

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

Di sini operator . merujuk pada kandungan tag, sepertimana operator @ merujuk pada atributnya.

10.2.4 E-buku sampel

```
<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 Fungsi terbina-dalam XPath

name() Nama bagi tag semasa

contains() `contains(s1, s2)` kembali *true* jika s1 mengandungi s2.

re:test() `re:test(src, pattern, flags)` returns *true* if the string *src* matches the regular expression *pattern*. A particularly useful flag is `i`, it makes matching case insensitive. A good primer on the syntax for regular expressions can be found at regexp syntax

10.3 Bahasa templat calibre

Bahasa templat calibre digunakan di merata-rata tempat. Ia digunakan untuk mengawal struktur folder dan nama fail bila menyimpan dari pustaka calibre ke cakera atau pembaca e-buku. Ia juga digunakan untuk takrif lajur "maya" yang mengandungi data dari lajur lain dan selanjutnya.

The basic template language is simple but has powerful advanced features. A template consists of text and names in curly brackets that are then replaced by the corresponding metadata from the book being processed. For example, the default template used for saving books to device in calibre is:

```
{author_sort}/{title}/{title} - {authors}
```

Bagi buku "The Foundation" oleh "Isaac Asimov" ia akan jadi:

```
Asimov, Isaac/The Foundation/The Foundation - Isaac Asimov
```

Miring atau slash adalah teks, yang diletak ke dalam templat dilokasi yang sepatutnya. Sebagai contoh, jika templat anda ialah:

```
{author_sort} Some Important Text {title}/{title} - {authors}
```

Bagi buku "The Foundation" oleh "Isaac Asimov" ia akan jadi:

```
Asimov, Isaac Some Important Text The Foundation/The Foundation - Isaac Asimov
```

You can use all the metadata fields available in calibre in a template, including any custom columns you have created, by using its 'lookup name'. To find the lookup name for a column (field) hover your mouse over the column header. Names for custom columns (columns you have created yourself) always have a # as the first character. For series type custom columns there is always an additional field named `#seriesname_index` that is the series index for that series. So if you have a custom series field named `#myseries`, there will also be a field named `#myseries_index`.

Selain dari medan berasaskan lajur, anda juga boleh guna:

```
{formats} - A list of formats available in the calibre library for a book
{identifiers:select(isbn)} - The ISBN of the book
```

If a book does not have a particular piece of metadata, the field in the template is replaced by the empty string for that book. Consider, for example:

```
{author_sort}/{series}/{title} {series_index}
```

Jika buku mempunyai siri, templat akan hasilkan:

```
Asimov, Isaac/Foundation/Second Foundation 3
```

dan jika buku tidak mempunyai siri:

```
Asimov, Isaac/Second Foundation
```

(calibre akan buang slash berbilang secara automatik dan jarak hadapan atau belakang).

10.3.1 Pemformatan lanjutan

Anda boleh lakukan lebih dari penggantian mudah dengan templat. Anda juga boleh sertakan teks secara bersyarat dan kawal bagaimana data terganti diformatkan.

First, conditionally including text. There are cases where you might want to have text 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 with a hyphen between them. calibre handles this case using a special field syntax.

For example, assume you want to use the template:

```
{series} - {series_index} - {title}
```

If the book has no series, the answer will be `- - title`. Most people would rather the result be simply `title`, without the hyphens. To do this, use the extended syntax `{field:|prefix_text|suffix_text}`. When you use this syntax, if field has the value SERIES then the result will be `prefix_textSERIESsuffix_text`. If field has no value, then the result will be the empty string (nothing); the prefix and suffix are ignored. The prefix and suffix can contain blanks. **Do not use subtemplates (`{ ... }`) or functions (see below) as the prefix or the suffix.**

Using this syntax, we can solve the above series problem with the template:

```
{series}{series_index:| - | - }{title}
```

The hyphens will be included only if the book has a series index, which it will have only if it has a series.

Notes: you must include the : character if you want to use a prefix or a suffix. You must either use no | characters or both of them; using one, as in `{field:| - }`, is not allowed. It is OK not to provide any text for one side or the other, such as in `{series:|| - }`. Using `{title:||}` is the same as using `{title}`.

Second: formatting. Suppose you wanted to ensure that the series_index is always formatted as three digits with leading zeros. This would do the trick:

```
{series_index:0>3s} - Three digits with leading zeros
```

If instead of leading zeros you want leading spaces, use:

```
{series_index:>3s} - Three digits with leading spaces
```

For trailing zeros, use:

```
{series_index:0<3s} - Three digits with trailing zeros
```

If you use series indices with sub values (e.g., 1.1), you might want to ensure that the decimal points 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. 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
```

Jika anda hanya mahukan dua abjad pertama data, gunakan:

```
{author_sort:.2} - Only the first two letter of the author sort name
```

The calibre template language comes from Python and for more details on the syntax of these advanced formatting operations, look at the Python documentation.

10.3.2 Menggunakan templat dalam lajur suai

Sometimes you want to display metadata in the book list that calibre does not normally display, or to display data in a way different from how calibre normally does. For example, you might want to display the ISBN, a field that calibre does not display. You can use custom columns for this by creating a column with the type 'column built from other columns' (hereafter called composite columns), and entering a template. Result: calibre will display a column showing the result of evaluating that template. To display the ISBN, create the column and enter `{identifiers:select(isbn)}` into 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.

You cannot edit the data displayed in a composite column. If you edit a composite column, for example by double-clicking it, you will open the template for editing, not the underlying data. Editing the template on the GUI is a quick way of testing and changing composite columns.

10.3.3 Using functions in templates - Single Function Mode

Suppose you want to display the value of a field in upper case, when that field is normally in title case. You can do this (and many more things) using the functions available for templates. For example, to display the title in upper case, use `{title:uppercase()}`. To display it in title case, use `{title:titlecase()}`.

Functions appear in the format part, going after the `:` and before the first `|` or the closing `}`. If you have both a format and a function reference, the function comes after another `:`. Functions must always end with `()`. Some functions take extra values (arguments), and these go inside the `()`.

Functions are always applied before format specifications. See further down for an example of using both a format and a function, where this order is demonstrated.

The syntax for using functions is `{field:function(arguments)}`, or `{field:function(arguments)|prefix|suffix}`. Arguments are separated by commas. Commas inside arguments must be preceded by a backslash (`\`). The last (or only) argument cannot contain a closing parenthesis (`)`). Functions return the value of the field used in the template, suitably modified.

Important: If you have programming experience, please note that the syntax in this mode (single function) is not what you might expect. Strings are not quoted. Spaces are significant. All arguments must be constants; there is no sub-evaluation. **Do not use subtemplates (`{ ... }`) as function arguments.** Instead, use *Template Program Mode* (halaman 152) and *General Program Mode* (halaman 158).

Many functions use regular expressions. In all cases, regular expression matching is case-insensitive.

The functions available are listed below. Note that the definitive documentation for functions is available in the section *Function reference* (halaman 162):

- `lowercase()` -- kembalikan nilai medan dengan huruf kecil.
- `uppercase()` -- kembalikan nilai medan dengan huruf besar.
- `titlecase()` -- kembalikan nilai medan dengan format kata tajuk.
- `capitalize()` -- kembalikan nilai dengan huruf pertama berhuruf besar dan lain-lain seterusnya berhuruf kecil.

- `contains(pattern, text if match, text if not match)` -- checks if field contains matches for the regular expression *pattern*. Returns *text if match* if matches are found, otherwise it returns *text if no match*.
- `count(separator)` -- interprets the value as a list of items separated by *separator*, returning 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(&)}*
- `format_number(template)` -- interprets the field as a number and format that number using a Python formatting template such as "{0:5.2f}" or "{0:,d}" or "${0:5,.2f}". The field_name part of the template must be a 0 (zero) (the "{0:" in the above examples). You can leave off the leading "{0:" and trailing "}" if the template contains only a format. See the template language and Python documentation 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 the field is not empty, return the value of the field. Otherwise return *text*.
- `in_list(separator, pattern, found_val, ..., not_found_val)` -- interpret the field as a list of items separated by *separator*, evaluating the *pattern* against each value in the list. If the *pattern* matches a value, 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.
- `language_codes(lang_strings)` -- return the language codes for the strings passed in *lang_strings*. The strings must be in the language of the current locale. *Lang_strings* is a comma-separated list.
- `language_strings(lang_codes, localize)` -- return the strings for the language codes passed in *lang_codes*. 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 field as a list of items separated by *separator*, returning the *index `th item. The first item is number zero. The last item can be returned using `list_item(-1,separator)*. If the item is not in the list, then the empty value is returned. The separator has the same meaning as in the *count* function.
- `lookup(pattern, field, pattern, field, ..., else_field)` -- like switch, except the arguments are field (metadata) names, not text. The value of the appropriate field will be fetched and used. Note that because composite columns are fields, you can use this function in one composite field to use the value of some other composite field. This is extremely useful when constructing variable save paths (more later).
- `rating_to_stars(use_half_stars)` -- Returns the rating as string of star characters. The source value must be a number between 0 and 5. Set use_half_stars to 1 if you want half star characters for custom ratings columns that are not integers, for example 2.5.
- `re(pattern, replacement)` -- return the field after applying the regular expression. All instances of *pattern* are replaced with *replacement*. As in all of calibre, these are Python-compatible regular expressions.
- `select(key)` -- interpret the field as a comma-separated list of items, with the items being of the form "id:value". Find the pair with the id equal to key, and return the corresponding value. This function is particularly useful for extracting a value such as an ISBN from the set of identifiers for a book.
- `shorten(left chars, middle text, right chars)` -- Return a shortened version of the field, consisting of *left chars* characters from the beginning of the field, followed by *middle text*, followed by *right chars* characters from the end of the string. *Left chars* and *right chars* must be integers. For example, assume the title of the book is *Ancient English Laws in the Times of Ivanhoe*, and you want it to fit in a space of at most 15 characters. If you use `{title:shorten(9,-,5)}`, the result will be *Ancient E-nhoe*. If the field's

length is less than `left chars` + `right chars` + the length of `middle text`, then the field will be used intact. For example, the title *The Dome* would not be changed.

- `str_in_list(separator, string, found_val, ..., not_found_val)` -- interpret the field as a list of items separated by *separator*, comparing the *string* against each value in the list. If the *string* matches a value (ignoring case), return *found_val*, otherwise return *not_found_val*. If the string contains separators, then it is also treated as a list and each value is checked. The *string* and *found_value* can be repeated as many times as desired, permitting returning different values depending on the search. The strings are checked in order. The first match is returned.
- `subitems(start_index, end_index)` -- This function is used to break apart lists of tag-like hierarchical items such as genres. It interprets the field as a comma-separated list of tag-like items, where each item is a period-separated list. Returns a new list made by first finding all the period-separated tag-like items, then for each such item extracting the components from *start_index* to *end_index*, then combining the results back together. The first component in a period-separated list has an index of zero. If an index is negative, then it counts from the end of the list. As a special case, an end_index of zero is assumed to be the length of the list. Examples:

```
Assuming a #genre column containing "A.B.C":
    {#genre:subitems(0,1)} returns "A"
    {#genre:subitems(0,2)} returns "A.B"
    {#genre:subitems(1,0)} returns "B.C"
Assuming a #genre column containing "A.B.C, D.E":
    {#genre:subitems(0,1)} returns "A, D"
    {#genre:subitems(0,2)} returns "A.B, D.E"
```

- `sublist(start_index, end_index, separator)` -- interpret the field as a list of items separated by *separator*, returning a new list made from the items from *start_index* to *end_index*. The first item is number zero. If an index is negative, then it counts from the end of the list. As a special case, an end_index of zero is assumed to be the length of the list. Examples assuming that the tags column (which is comma-separated) contains "A, B ,C":

```
{tags:sublist(0,1,\,)} returns "A"
{tags:sublist(-1,0,\,)} returns "C"
{tags:sublist(0,-1,\,)} returns "A, B"
```

- `swap_around_articles(separator)` -- returns the val with articles moved to the end. The value can be a list, in which case each member of the list is processed. If the value is a list then you must provide the list value separator. If no separator is provided then the value is treated as being a single value, not a list.
- `swap_around_comma()` -- given a field with 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, the function returns val unchanged.
- `switch(pattern, value, pattern, value, ..., else_value)` -- for each `pattern, value` pair, checks if the field matches the regular expression `pattern` and if so, returns that `value`. If no `pattern` matches, then `else_value` is returned. You can have as many `pattern, value` pairs as you want.
- `test(text if not empty, text if empty)` -- return *text if not empty* if the field is not empty, otherwise return *text if empty*.
- `transliterate()` -- Returns 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 a the function returns `Fiodor Mikhailovich Dostoievskii`.'

Now, what about using functions and formatting in the same field. Suppose you have an integer custom column called `#myint` that you want to see with leading zeros, as in `003`. To do this, you would use a format of `0>3s`. However, by default, if a number (integer or float) equals zero then the field produces the empty value, so zero values will produce nothing, not 000. If you really 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 field reference 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 field:

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

10.3.4 More complex functions in templates - Template Program Mode

Template Program Mode differs from Single Function Mode in that it permits writing template expressions that refer to other metadata fields, use nested functions, modify values, and do arithmetic. It is a reasonably complete programming language.

You can use the functions documented above in Template Program Mode. See below for details.

Beginning with an example, assume you want your 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 basic template language because you cannot make reference to another metadata field within a template expression. In Template Program Mode, you can. The following expression works:

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

The example shows several things:

- Template Program Mode is used if the expression begins with :' and ends with '. Anything else is assumed to be in single function mode.
- the variable $ stands for the field 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, which is a significant difference from single-function mode.
- white space is ignored and can be used anywhere within the expression.
- constant strings are enclosed in matching quotes, either ' or ".

The syntax of the language is shown by the following grammar. For a discussion of 'compare','if_expression', and 'template_call' see *General Program Mode* (halaman 158)::

```
program         ::= expression_list
expression_list ::= expression [ ';' expression ]*
expression      ::= identifier | constant | function | assignment | compare | if_
↪expression
function        ::= identifier '(' expression [ ',' expression ]* ')'
compare         ::= expression compare_op expression
compare_op      ::= '==' | '!=' | '>=' | '>' | '<=' | '<' | '==#' | '!=#' | '>=#' | '>
↪#' | '<=#' | '<#'
if_expression   ::= 'if' expression 'then' expression_list [elif_expression] ['else'␣
↪expression_list] 'fi'
elif_expression ::= 'elif' expression 'then' expression_list elif_expression | ''
assignment      ::= identifier '=' expression
constant        ::= " string " | ' string ' | number
identifier      ::= sequence of letters or ``_`` characters
```

Comments are lines with a '#' character at the beginning of the line.

An expression without errors always has a value. The value of an expression_list is the value of the last expression in the list. As such, the value of the program (expression_list):

```
1; 2; 'foobar'; 3
```

is 3.

Another example of a complex but rather silly program might help make things clearer:

```
{series_index:'
    substr(
        strcat($, '->',
            cmp(divide($, 2), 1,
                assign(c, 1); substr('lt123', c, 0),
                'eq', 'gt')),
        0, 6)
   '| prefix | suffix}
```

This program does the following:

- specify that the field being looked at is series_index. The variable $ is set to its value.
- calls the substr function, which takes 3 parameters (str, start, end). It returns a string formed by extracting the start through end characters from string, zero-based (the first character is character zero). In this case the string will be computed by the strcat function, the start is 0, and the end is 6. In this case it will return the first 6 characters of the string returned by strcat, which must be evaluated before substr can return.
- calls the strcat (string concatenation) function. Strcat accepts 1 or more arguments, and returns a string formed by concatenating all the values. In this case there are three arguments. The first parameter is the value in $, which here is the value of series_index. The second paremeter is the constant string '->'. The third parameter is the value returned by the cmp function, which must be fully evaluated before strcat can return.
- The cmp function takes 5 arguments (x, y, lt, eq, gt). It compares x and y and returns the third argument lt if x < y, the fourth argument eq if x == y, and the fifth argument gt if x > y. As with all functions, all of the parameters can be statements. In this case the first parameter (the value for x) is the result of dividing the series_index by 2. The second parameter y is the constant 1. The third parameter lt is a statement (more later). The fourth parameter eq is the constant string 'eq'. The fifth parameter is the constant string 'gt'.
- The third parameter (the one for lt) is a statement, or a sequence of expressions. Remember that a statement (a sequence of semicolon-separated expressions) is also an expression, returning the value of the last expression in the list. In this case, the program first assigns the value 1 to a local variable c, then returns a substring made by extracting the c'th character to the end. Since c always contains the constant 1, the substring will return the second through end'th characters, or 't123'.
- Once the statement providing the value to the third parameter is executed, cmp can return a value. At that point, strcat` can return a value, then ``substr can return a value. The program then terminates.

For various values of series_index, the program returns:

- series_index == undefined, result = prefix ->t123 suffix
- series_index == 0.5, result = prefix 0.50-> suffix
- series_index == 1, result = prefix 1->t12 suffix
- series_index == 2, result = prefix 2->eq suffix
- series_index == 3, result = prefix 3->gt suffix

All the functions listed under single-function mode can be used in program mode. To do so, you must supply the value that the function is to act upon as the first parameter in addition to the parameters documented above. For example, in program mode the parameters of the *test* function are `test(x, text_if_not_empty, text_if_empty)`. The *x* parameter, which is the value to be tested, will almost always be a variable or a function call, often *field()*.

The following functions are available in addition to those described in single-function mode. Remember from the example above that the single-function mode functions require an additional first parameter specifying the field to operate on. With the exception of the `id` parameter of assign, all parameters can be statements (sequences of expressions). Note that the definitive documentation for functions is available in the section *Function reference* (halaman 162):

- `and(value, value, ...)` -- returns the string "1" if all values are not empty, otherwise returns the empty string. This function works well with test or first_non_empty. You can have as many values as you want.
- `add(x, y)` -- returns x + y. Throws an exception if either x or y are not numbers.
- `assign(id, val)` -- assigns val to id, then returns val. id must be an identifier, not an expression
- `approximate_formats()` -- return a comma-separated list of formats that at one point were associated with the book. There is no guarantee that the list is correct, although it probably is. This function can be called in Template Program Mode using the template `{:'approximate_formats()'}`. Note that format names are always uppercase, as in EPUB.
- `author_links(val_separator, pair_separator)` -- returns a string containing a list of authors and that author's link values in the form `author1 val_separator author1link pair_separator author2 val_separator author2link` 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 (different from the author_sort in books). The returned list has the form author sort 1 `val_separator` author sort 2 etc. 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 separator string
- `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 the value of the yes/no field named by the lookup key `field_name` for a value specified by the parameters, returning "yes" if a match is found, otherwise returning an 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 undefined (neither True nor False) or True. More than one of `is_undefined`, `is_false`, or `is_true` can be set to 1. This function is usually used by the `test()` or `is_empty()` functions.
- `ceiling(x)` -- returns the smallest integer greater than or equal to x. Throws an exception if x is not a number.
- `cmp(x, y, lt, eq, gt)` -- compares x and y after converting both to numbers. Returns `lt` if x < y. Returns `eq` if x == y. Otherwise returns `gt`.
- `connected_device_name(storage_location)` -- if a device is connected then return the device name, otherwise return the empty string. Each storage location on a device can have a different name. The 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. This function can be called in Template Program Mode using the template `{:'current_library_name()'}`.

- `current_library_path()` -- return the path to the current calibre library. This function can be called in Template Program Mode using the template `{:'current_library_path()'}`.
- `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.
- `eval(string)` -- evaluates the string as a program, passing the local variables (those `assign` ed to). This permits using the template processor to construct complex results from local variables. Because the *{* and *}* characters are special, 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.
- `field(name)` -- returns the metadata field named by `name`.
- `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 *General Program Mode* (halaman 158) (see below) to take advantage of GPM template compilation. 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, cmp1, result1, cmp2, r2, ..., else_result)` -- compares `val < cmpN` in sequence, returning resultN for the first comparison that succeeds. Returns else_result if no comparison succeeds. Example:

  ```
  first_matching_cmp(10,5,"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 value 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 field, 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").
  ```

(continues on next page)

(continued from previous page)

```
yy   : the year as two digit number (00 to 99).
yyyy : the year as four digit number.
h    : the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
hh   : the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
m    : the minutes without a leading 0 (0 to 59)
mm   : the minutes with a leading 0 (00 to 59)
s    : the seconds without a leading 0 (0 to 59)
ss   : the seconds with a leading 0 (00 to 59)
ap   : use a 12-hour clock instead of a 24-hour clock, with 'ap' replaced by the␣
↪localized string for am or pm.
AP   : use a 12-hour clock instead of a 24-hour clock, with 'AP' replaced by the␣
↪localized string for AM or PM.
iso  : the date with time and timezone. Must be the only format present.
```

You might get unexpected results if the date you are formatting contains localized month names, which can happen if you changed the format tweaks to contain `MMMM`. In this case, instead of using something like `{pubdate:format_date(yyyy)}`, write the template using Template Program Mode as in `{:'format_date(raw_field('pubdate'),'yyyy')'}`.

- `formats_modtimes(format_string)` -- return a comma-separated list of colon-separated items representing modification times for the formats of a book. The 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 mod 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 representing 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 items representing 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 return the empty string
- `not(value)` -- returns the string "1" if the value is empty, otherwise returns the empty string. This function works well with test or first_non_empty.
- `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_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_1, for_group_2, ...)` -- Like list_re except replacements are not optional. It uses re_group(item, search_re, template ...) when doing the replacements.
- `list_sort(list, direction, separator)` -- return list sorted using a case-insensitive sort. If *direction* is zero, the list is sorted ascending, otherwise descending. The list items are separated by separator, as are the items in the returned list.
- `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.
- `mod(x)` -- returns the remainder of x / y, where x, y, and the result are integers. Throws an exception if either x or y is not a number.
- `multiply(x, y)` -- returns x * y. Throws an exception if either x or y are not numbers.
- `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. This function works well with test or first_non_empty. You can have as many values as you want.
- `print(a, b, ...)` -- prints the arguments to standard output. Unless you start calibre from the command line (`calibre-debug -g`), the output will go to a black hole.
- `raw_field(name)` -- returns the metadata field named by name without applying any formatting.
- `raw_list(name, separator)` -- returns the metadata list named by name without applying any formatting or sorting and with items separated by separator.
- `re_group(val, pattern, template_for_group_1, for_group_2, ...)` -- return a string made by applying the regular expression pattern to the val and replacing each matched instance with the string computed by replacing each matched group by the value returned by the corresponding template. The original matched value for the group is available as $. In Template Program Mode, like for the template and the eval functions, you use [[for { and]] for }. The following example in Template Program Mode looks for series with more than one word and uppercases the first word:

```
{series:'re_group($, "(\S* )(.*)", "[[$: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. *Prefix, string* pairs are added to the end of the value as long as the resulting string length is less than *max*. String1 is returned 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 comparison x and y as strings. Returns `lt` if x < y. Returns `eq` if x == y. Otherwise returns `gt`.
- `strlen(a)` -- Returns the length of the string passed as the argument.
- `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.

- `today()` -- return a date string for today. This value is designed for use in format_date or days_between, but can be manipulated like any other string. The date is in ISO 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. Because the *{* and *}* characters are special, you must use *[[* for the *{* character and *]]* for the '}' character; they are converted automatically. For example, ```template('[[title_sort]]') will evaluate the template ``{title_sort}``` and return its value. Note also that prefixes and suffixes (the *|prefix|suffix* syntax) cannot be used in the argument to this function when using Template Program Mode.

10.3.5 Using General Program Mode

For more complicated template programs it is often easier to avoid template syntax (all the *{* and *}* characters), instead writing a more classic-looking program. You can do this by beginning the template with *program:*. The template program is compiled and executed. No template processing (e.g., formatting, prefixes, suffixes) is done. The special variable *$* is not set.

One advantage of *program:* mode is that braces are no longer special. For example, it is not necessary to use *[[* and *]]* when using the *template()* function. Another advantage is readability.

Both General and Template Program Modes support `if` expressions with the following syntax:

```
if <<expression>> then
    <<expression_list>>
[elif <<expression>> then <<expression_list>>]*
[else <<expression_list>> ]
fi
```

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. <<expression>> is one template language expression; semicolons are not allowed. <<expression_list>> is a semicolon-separated sequence of template language expressions, including nested ifs. Examples:

- `program: if field('series') then 'yes' else 'no' fi`
- `program: if field('series') then a = 'yes'; b = 'no' else a = 'no'; b='yes' fi; strcat(a, '-', b)`
- Nested `if` example:

```
program:
    if field('series')
    then
        if check_yes_no(field('#mybool'), '', '', '1')
        then
            'yes'
        else
            'no'
        fi
    else
        'no series'
    fi
```

An `if` produces a value like any other language expression. This means that all the following are valid:

- `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`
- `program: a = field(if field('series') then 'series' else 'title' fi); a`

Template Program Mode and General Program Mode support classic relational (comparison) operators: `==`, `!=`, `<`, `<=`, `>`, `>=`. The operators return '1' if they evaluate to True, otherwise ''. They do case-insensitive string comparison using lexical order. Examples:

- `program: field('series') == 'foo'` returns '1' if the book's series is 'foo'.
- `program: if field('series') != 'foo' then 'bar' else 'mumble' fi` returns 'bar' if the book's series is not 'foo', else 'mumble'.
- `program: if or(field('series') == 'foo', field('series') == '1632') 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 it is doing a lexical comparison. If you want numeric comparison instead of lexical comparison, use the operators `==#`, `!=#`, `<#`, `<=#`, `>#`, `>=#`. In this case the left and right values are set to zero if they are undefined or the empty string. If they are not numbers then an error is raised.

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

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

In the stored template you retrieve the arguments passed in the call using the `arguments` function. It both declares and initializes local variables, effectively parameters. The variables are positional; they get the value of the value 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` will have the value `myseries` and the argument `alternate` will have the 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 `#genre`.

An easy way to test stored templates is using the `Template tester` dialog. 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.

10.3.6 Notes on the difference between modes

The three program modes, Single Function Mode (SFM), Template Program Mode (TPM), and General Program Mode (GPM), work differently. SFM is intended to be 'simple' so it hides a lot of programming language bits. For example, the value of the column is always passed as an 'invisible' first argument to a function included in the template. SFM also doesn't support the difference between variables and strings; all values are strings.

Example: 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))``
```

10.3.7 User-defined Python template functions

You can add your own Python functions to the template processor. Such functions can be used in any of the three template programming modes. The functions are added by going to *Preferences* → *Advanced* → *Template functions*. Instructions are shown in that dialog.

10.3.8 Special notes for save/send templates

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, 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.

10.3.9 Templates and plugboards

Plugboards are used for changing the metadata written into books during send-to-device and save-to-disk operations. A plugboard permits you to specify a template to provide the data to write into the book's metadata. You can use plugboards to modify the following fields: authors, author_sort, language, publisher, tags, title, title_sort. This feature helps people who want to use different metadata in books on devices to solve sorting or display issues.

When you create a plugboard, you specify the format and device for which the plugboard is to be used. A special device is provided, save_to_disk, that is used when saving formats (as opposed to sending them to a device). Once you have chosen the format and device, you choose the metadata fields to change, providing templates to supply the new values. These templates are *connected* to their destination fields, hence the name *plugboards*. You can, of course, use composite columns in these templates.

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:

- satu papan palam dengan padanan tepat bagi format dan peranti seperti, `EPUB` dan `ANDROID`
- satu papan palam dengan padanan tepat pada format dan pilihan `any device` khas, seperti, `EPUB` dan `any device`
- satu papan palam dengan pilihan `any format` khas dan padanan tepat pada peranti seperti, `any format` dan `ANDROID`
- satu papan palam dengan `any format` dan `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.

10.3.10 Tips

Anda boleh dapati petua berikut adalah berguna!

- 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 could use Stored Templates.
- Dalam papan palam, anda boleh tetapkan medan menjadi kosong (atau apa jua yang menyamai dengan kosong) dengan menggunakan templat khas `{}`. Templat ini akan sentiasa menilai pada rentetan kosong.
- The technique described above to show numbers even if they have a zero value works with the standard field series_index.

10.3.11 Function reference

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* (halaman 165)
  - *add(x, y)* (halaman 165)
  - *ceiling(x)* (halaman 165)
  - *divide(x, y)* (halaman 165)
  - *floor(x)* (halaman 165)
  - *fractional_part(x)* (halaman 166)
  - *mod(x)* (halaman 166)
  - *multiply(x, y)* (halaman 166)
  - *round(x)* (halaman 166)
  - *subtract(x, y)* (halaman 166)
- *Boolean* (halaman 166)
  - *and(value, value, ...)* (halaman 166)
  - *not(value)* (halaman 166)
  - *or(value, value, ...)* (halaman 167)
- *Date functions* (halaman 167)
  - *days_between(date1, date2)* (halaman 167)

  - *today()* (halaman 167)
- *Formatting values* (halaman 167)
  - *finish_formatting(val, fmt, prefix, suffix)* (halaman 167)
  - *format_date(val, format_string)* (halaman 167)
  - *format_number(v, template)* (halaman 168)
  - *human_readable(v)* (halaman 168)
  - *rating_to_stars(value, use_half_stars)* (halaman 168)
- *Get values from metadata* (halaman 168)
  - *annotation_count()* (halaman 168)
  - *approximate_formats()* (halaman 168)
  - *author_links(val_separator, pair_separator)* (halaman 168)
  - *author_sorts(val_separator)* (halaman 169)
  - *booksize()* (halaman 169)
  - *connected_device_name(storage_location)* (halaman 169)
  - *current_library_name()* (halaman 169)
  - *current_library_path()* (halaman 169)
  - *field(name)* (halaman 169)
  - *formats_modtimes(date_format)* (halaman 170)
  - *formats_paths()* (halaman 170)
  - *formats_sizes()* (halaman 170)
  - *has_cover()* (halaman 170)
  - *language_codes(lang_strings)* (halaman 170)
  - *language_strings(lang_codes, localize)* (halaman 170)
  - *ondevice()* (halaman 171)
  - *raw_field(name)* (halaman 171)
  - *raw_list(name, separator)* (halaman 171)
  - *series_sort()* (halaman 171)
  - *user_categories()* (halaman 171)
  - *virtual_libraries()* (halaman 171)
- *If-then-else* (halaman 172)
  - *check_yes_no(field_name, is_undefined, is_false, is_true)* (halaman 172)
  - *contains(val, pattern, text if match, text if not match)* (halaman 172)
  - *ifempty(val, text if empty)* (halaman 172)
  - *test(val, text if not empty, text if empty)* (halaman 172)
- *Iterating over values* (halaman 172)

  - *first_non_empty(value, value, ...)* (halaman 172)
  - *lookup(val, pattern, field, pattern, field, ..., else_field)* (halaman 172)
  - *switch(val, pattern, value, pattern, value, ..., else_value)* (halaman 173)
- *List lookup* (halaman 173)
  - *_identifier_in_list(val, id, found_val, not_found_val)* (halaman 173)
  - *in_list(val, separator, pattern, found_val, ..., not_found_val)* (halaman 173)
  - *list_item(val, index, separator)* (halaman 173)
  - *select(val, key)* (halaman 173)
  - *str_in_list(val, separator, string, found_val, ..., not_found_val)* (halaman 174)
- *List manipulation* (halaman 174)
  - *count(val, separator)* (halaman 174)
  - *list_difference(list1, list2, separator)* (halaman 174)
  - *list_equals(list1, sep1, list2, sep2, yes_val, no_val)* (halaman 174)
  - *list_intersection(list1, list2, separator)* (halaman 174)
  - *list_re(src_list, separator, include_re, opt_replace)* (halaman 175)
  - *list_re_group(src_list, separator, include_re, search_re, group_1_template, ...)* (halaman 175)
  - *list_sort(list, direction, separator)* (halaman 175)
  - *list_union(list1, list2, separator)* (halaman 175)
  - *subitems(val, start_index, end_index)* (halaman 175)
  - *sublist(val, start_index, end_index, separator)* (halaman 176)
- *Other* (halaman 176)
  - *assign(id, val)* (halaman 176)
  - *print(a, b, ...)* (halaman 176)
- *Recursion* (halaman 176)
  - *eval(template)* (halaman 176)
  - *template(x)* (halaman 176)
- *Relational* (halaman 177)
  - *cmp(x, y, lt, eq, gt)* (halaman 177)
  - *first_matching_cmp(val, cmp1, result1, cmp2, r2, ..., else_result)* (halaman 177)
  - *strcmp(x, y, lt, eq, gt)* (halaman 177)
- *String case changes* (halaman 177)
  - *capitalize(val)* (halaman 177)
  - *lowercase(val)* (halaman 177)
  - *titlecase(val)* (halaman 177)
  - *uppercase(val)* (halaman 178)

- *String manipulation* (halaman 178)
  - *re(val, pattern, replacement)* (halaman 178)
  - *re_group(val, pattern, template_for_group_1, for_group_2, ...)* (halaman 178)
  - *shorten(val, left chars, middle text, right chars)* (halaman 178)
  - *strcat(a, b, ...)* (halaman 178)
  - *strcat_max(max, string1, prefix2, string2, ...)* (halaman 179)
  - *strlen(a)* (halaman 179)
  - *substr(str, start, end)* (halaman 179)
  - *swap_around_articles(val, separator)* (halaman 179)
  - *swap_around_comma(val)* (halaman 179)
  - *transliterate(a)* (halaman 179)
- *API of the Metadata objects* (halaman 180)

Arithmetic

add(x, y)

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

add(x, y) -- kembalikan x + y. Buang pengecualian jika sama ada x atau y bukan nombor.

ceiling(x)

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

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

divide(x, y)

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

divide(x, y) -- kembalikan x / y. Buang pengecualian jika sama ada x atau y bukan nombor.

floor(x)

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

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

fractional_part(x)

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

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.

mod(x)

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

mod(x) -- returns the remainder of x / y, where x, y, and the result are integers. Throws an exception if either x or y is not a number.

multiply(x, y)

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

multiply(x, y) -- kembalikan x * y. Buang pengecualian jika sama ada x atau y bukan nombor.

round(x)

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

round(x) -- returns the nearest integer to x. Throws an exception if x is not a number.

subtract(x, y)

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

subtract(x, y) -- kembalikan x - y. Buang pengecualian jika sama ada x atau y bukan nombor.

Boolean

and(value, value, ...)

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

and(value, value, ...) -- kembalikan rentetan "1" jika semua nilai tidak kosong, jika tidak kembalikan rentetan kosong. Fungsi ini berfungsi baik dengan test atau first_non_empty. Anda boleh adakan sebanyak mana nilai yang anda mahu.

not(value)

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

not(value) -- kembalikan rentetan "1" jika semua nilai tidak kosong, jika tidak kembalikan rentetan kosong. Fungsi ini berfungsi baik dengan test atau first_non_empty.

or(value, value, ...)

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

or(value, value, ...) -- kembalikan rentetan "1" jika semua nilai tidak kosong, jika tidak kembalikan rentetan kosong. Fungsi ini berfungsi baik dengan test atau first_non_empty. Anda boleh adakan sebanyak mana nilai yang anda mahu.

Date functions

days_between(date1, date2)

class calibre.utils.formatter_functions.**BuiltinDaysBetween**

days_between(date1, date2) -- kembalikan bilangan hari diantara date1 dan date2. Bilangan adalah positif jika date1 lebih besar dari date2, jika tidak ia bernilai negatif. Jika date1 atau date2 tidak mempunyai tarikh, maka fungsi kembalikan rentetan kosong.

today()

class calibre.utils.formatter_functions.**BuiltinToday**

today() -- kembalikan retetan tarikh untuk hari ini. Nilai ini direka untuk kegunaan dalam format_date atau days_between, tetapi boleh dimanipulasi seperti mana-mana rentetan. Tarikh dalam format ISO.

Formatting values

finish_formatting(val, fmt, prefix, suffix)

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

finish_formatting(val, fmt, prefix, suffix) -- laksanakan format, awalan, dan akhiran pada nilai dalam cara yang sama dibuat dalam templat seperti *{series_index:05.2f*| - |- *}*. Sebagai contoh, program berikut hasilkan output yang sama seperti di atas template: program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")

format_date(val, format_string)

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

format_date(val, format_string) -- formatkan nilai, iaitu tarikh, menggunakan format_string, mengembalikan rentetan. Kod pemformatan adalah: d : hari dalam nombor tanpa sifar dihadapan (1 hingga 31) dd : hari sebagai nombor dengan sifar dihadapan (01 hingga 31) ddd : singkatan nama hari (contoh. "Isn" hingga "Ahd"). dddd : nama hari tempatan panjang (contoh "Isnin" hingga "Ahad"). M : bulan sebagai nombor tanpa sifar dihadapan (1 hingga 12). MM : bulan sebagai nombor dengan sifat dihadapan (01 hingga 12) MMM : singkatan nama bulan tempatan (contoh. "Jan" hingga "Dis"). MMMM : nama bulan tempatan panjang (contoh "Januari" hingga "Disember"). yy : tahun dalam dua nombor digit (00 hingga 99). yyyy : tahun dalam empat nombor digit. h : jam tanpa sifar dihadapan 0 (0 hingga 11 atau 0 hingga 23, bergantung pada am/pm) hh : jam dengan sifar dihadapan 0 (00 hingga 11 atau 00 hingga 23, bergantung pada am/pm) m : minit tanpa sifar dihadapan 0 (0 hingga 59) mm : minit dengan sifar dihadapan 0 (00 hingga 59) s : saat tanpa sifar dihadapan 0 (0 hingga 59) ss : saat tanpa sifar dihadapan 0 (00 hingga 59) ap : guna jam 12 selain dari jam 24, dengan "ap" diganti dengan rentetan tempatan bagi am atau pm AP : guna jam 12 selain dari jam 2, dengan "AP" diganti oleh rentetan tempatan untuk AM atau PM iso : tarikh dengan masa dan zon waktu. Mesti dalam format semasa sahaja

format_number(v, template)

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

format_number(v, template) -- format the number v using a Python formatting template such as "{0:5.2f}" or "{0:,d}" or "${0:5,.2f}". The field_name part of the template must be a 0 (zero) (the "{0:" in the above examples). See the template language and Python documentation for more examples. You can leave off the leading "{0:" and trailing "}" if the template contains only a format. Returns the empty string if formatting fails.

human_readable(v)

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

human_readable(v) -- kembalikan rentetan yang diwakili dengan nombor v dalam KB, MB, GB, dan lain-lain.

rating_to_stars(value, use_half_stars)

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

rating_to_stars(value, use_half_stars) -- Returns the rating as string of star characters. The value is a number between 0 and 5. Set use_half_stars to 1 if you want half star characters for custom ratings columns that support non-integer ratings, for example 2.5.

Get values from metadata

annotation_count()

class calibre.utils.formatter_functions.**BuiltinAnnotationCount**

annotation_count() -- return the total number of annotations of all types attached to the current book. This function works only in the GUI.

approximate_formats()

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

approximate_formats() -- kembalikan senarai dipisah-tanda-koma bagi format yang mana satu keadaan dikaitkan dengan buku. Tiada jaminan senarai ini adalah betul, walaupun ia mungkin betul. Fungsi ini boleh dipanggil dalam mod program templat menggunakan template "{:'approximate_formats()'}". Perhatian nama format sentiasa berhuruf besar, seperti dalam EPUB. Fungsi hanya berfungsi dalam GUI. Jika anda mahu guna nilai ini dalam templat save-to-disk atau send-to-device maka anda mesti buat "Column built from other columns" suai, guna fungsi dalam templat lajur, dan guna nilai lajur tersebut dalam templat simpan/hantar anda

author_links(val_separator, pair_separator)

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

author_links(val_separator, pair_separator) -- kembalikan rentetan mengandungi senarai pengarang dan nilai pautan pengarang dalam bentuk author1 val_separator author1link pair_separator author2 val_separator author2link dan lain-lain. Pengarang dipisah dari nilai pautannya oleh rentetan val_separator tanpa jarak ditambah kepadanya. Pasangan author:linkvalue dipisah dengan argumen rentetan pair_separator string tanda jarak ditambah kepadanya. Ia bergantung kepada anda untuk memilih rentetan pemisah yang tidak muncul dalam nama atau pautan pengarang. Pengarang disertakan walaupun jika pautan pengarang kosong.

author_sorts(val_separator)

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

author_sorts(val_separator) -- kembalikan rentetan mengandungi senarai nilai isih pengarang untuk pengarang buku. Isih adalah salah satu data meta pengarang (berbeza dari author_sort di dalam buku). Senarai yang dikembalikan mempunyai bentuk author isih 1 val_separator author isih 2 dan lain-lain. Nilai isih pengarang dalam senarai ini adalah dalam tertib yang sama seperti pengarang buku. Jika anda mahu jarak disekitar val_separator maka sertakannya di dalam rentetan pemisah

booksize()

class calibre.utils.formatter_functions.**BuiltinBooksize**

booksize() -- kembalikan nilai medan saiz. Fungsi ini hanya berkesan dalam GUI. Jika anda mahu guna nilai ini dalam templat save-to-disk atau send-to-device maka anda mesti buat "Column built from other columns" suai, gunakan fungsi yang mana dalam lajur templat, dan gunan ilai lajur dalam templat simpan/hantar anda

connected_device_name(storage_location)

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceName**

connected_device_name(storage_location) -- if a device is connected then return the device name, otherwise return the empty string. Each storage location on a device can have a different name. The location names are 'main', 'carda' and 'cardb'. This function works only in the GUI.

current_library_name()

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**

current_library_name() -- kembalikan nama terakhir pada laluan ke pustaka calibre semasa. Fungsi ini boleh dipanggil dalam mod program templat menggunakan templat "{:'current_library_name()'}".

current_library_path()

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryPath**

current_library_path() -- kembalikan laluan ke pustaka calibre semasa. Fungsi ini boleh dikenali dalam mod program templat menggunakan templat "{:'current_library_path()'}".

field(name)

class calibre.utils.formatter_functions.**BuiltinField**

field(name) -- kembalikan medan data meta dinamakan mengikut nama

formats_modtimes(date_format)

class `calibre.utils.formatter_functions.`**`BuiltinFormatsModtimes`**

formats_modtimes(date_format) -- kembalikan senarai dipisah-tanda-koma bagi item dipisah tanda-titik-bertindih yang mewakili waktu pengubahsuaian untuk format sesebuah buku. Parameter date_format menyatakan bagaimana tarikh diformatkan. Sila rujuk fungsi format_date untuk perincian. Anda boleh guna fungsi pilih untuk dapatkan waktu mod bagi format tertentu. Perhatian nama format sentiasa berhuruf-besar, sepertimana dalam EPUB.

formats_paths()

class `calibre.utils.formatter_functions.`**`BuiltinFormatsPaths`**

formats_paths() -- kembalikan senarai dipisah tanda-koma bagi item dipisah tanda-titik-bertindih mewakili laluan penuh ke format sesebuah buku. Anda boleh guna fungsi pilih untuk dapatkan laluan untuk format tertentu. Perhatian nama format sentiasa berhuruf besar, sepertimana dalam EPUB.

formats_sizes()

class `calibre.utils.formatter_functions.`**`BuiltinFormatsSizes`**

formats_sizes() -- kembalikan senarai dipisah tanda-koma bagi item dipisah tanda-titik-bertindih mewakili saiz dalam baik bagi format sesebuah buku. Anda boleh guna fungsi pilih untuk dapatkan saiz bagi format tertentu. Perhatian nama format sentiasa berhuruf besar, sepertimana dalam EPUB.

has_cover()

class `calibre.utils.formatter_functions.`**`BuiltinHasCover`**

has_cover() -- kembalikan Yes jika buku mempunyai kulit buku, jika tidak kembalikan rentetan kosong

language_codes(lang_strings)

class `calibre.utils.formatter_functions.`**`BuiltinLanguageCodes`**

language_codes(lang_strings) -- kembalian kod bahasa bagi rentetan yang dilepasi dalam lang_strings. Rentetan mestilah dalam bahasa lokal semasa. Lang_strings adalah senarai dipisah-tanda-koma.

language_strings(lang_codes, localize)

class `calibre.utils.formatter_functions.`**`BuiltinLanguageStrings`**

language_strings(lang_codes, localize) -- kembalikan rentetan kod bahasa yang dilepasi dalam lang_codes. Jika dilokalkan adalah sifar, kembalikan rentetan dalam bahasa Inggeris. Jika dilokalkan tidak benilai sifar, maka kembalikan rentetan dalam bahasa lokal semasa. Lang_codes adalah senarai dipisah-tanda-koma.

ondevice()

class calibre.utils.formatter_functions.**BuiltinOndevice**

ondevice() -- kembalikan Ya jika ondevice ditetapkan, jika tidak kembalikan rentetan kosong. Fungsi ini hanya berkesan dalam GUI. Jika anda mahu guna nilai ini dalam templat save-to-disk atau send-to-device maka anda mesti buat "Column built from other columns" suai, gunakan fungsi yang mana dalam lajur templat, dan gunan ilai lajur dalam templat simpan/hantar anda

raw_field(name)

class calibre.utils.formatter_functions.**BuiltinRawField**

raw_field(name) -- kembalikan medan data meta dinamakan mengikut nama tanpa melaksanakan apa-apa pemformatan.

raw_list(name, separator)

class calibre.utils.formatter_functions.**BuiltinRawList**

raw_list(name, separator) -- kembalikan senarai data meta dinamakan mengikut nama tanpa melaksanakan apa-apa pemformatan atau mengisih dan dengan item diasing dengan pemisah.

series_sort()

class calibre.utils.formatter_functions.**BuiltinSeriesSort**

series_sort() -- kembalikan siri isih nilai

user_categories()

class calibre.utils.formatter_functions.**BuiltinUserCategories**

user_categories() -- kembalikan senarai dipisah-tanda-koma bagi kategori pengguna yang terkandung dalam buku ini. Fungsi ini hanya berfungsi dalam GUI. Jika anda mahu guna nilai ini dalam templat save-to-disk atau send-to-device maka anda mesti buat "Column built from other columns" suai, guna fungsi dalam templat lajur, dan guna nilai lajur dalam templat simpan/hantar anda

virtual_libraries()

class calibre.utils.formatter_functions.**BuiltinVirtualLibraries**

virtual_libraries() -- return a comma-separated list of Virtual libraries that contain this book. This function works only in the GUI. If you want to use these values in save-to-disk or send-to-device templates then you must make a custom "Column built from other columns", use the function in that column's template, and use that column's value in your save/send templates

If-then-else

check_yes_no(field_name, is_undefined, is_false, is_true)

class calibre.utils.formatter_functions.**BuiltinCheckYesNo**

check_yes_no(field_name, is_undefined, is_false, is_true) -- checks the value of the yes/no field named by the lookup key field_name for a value specified by the parameters, returning "yes" if a match is found, otherwise returning an 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 undefined (neither True nor False) or True. More than one of is_undefined, is_false, or is_true can be set to 1. This function is usually used by the test() or is_empty() functions.

contains(val, pattern, text if match, text if not match)

class calibre.utils.formatter_functions.**BuiltinContains**

contains(val, pattern, text if match, text if not match) -- periksa jika val mengandungi padanan untuk ungkapan nalar *pattern*. Kembalikan *text if match* jika padanan ditemui, jika tidak kembalikan *text if no match*

ifempty(val, text if empty)

class calibre.utils.formatter_functions.**BuiltinIfempty**

ifempty(val, text if empty) -- kembalikan val jika val tidak kosong, jika tidak kembalikan *text if empty*

test(val, text if not empty, text if empty)

class calibre.utils.formatter_functions.**BuiltinTest**

test(val, text if not empty, text if empty) -- kembalikan *text if not empty* jika val tidak kosong, jika tidak kembalikan *text if empty*

Iterating over values

first_non_empty(value, value, ...)

class calibre.utils.formatter_functions.**BuiltinFirstNonEmpty**

first_non_empty(value, value, ...) -- kembalikan nilai pertama yang tidak kosong. Jika semua nilai adalah kosong, maka nilai kosong dikembalikan. Anda boleh adakan sebanyak mana nilai yang anda mahu.

lookup(val, pattern, field, pattern, field, ..., else_field)

class calibre.utils.formatter_functions.**BuiltinLookup**

lookup(val, pattern, field, pattern, field, ..., else_field) -- seperti switch, kecuali argumen adalah nama medan (data meta), bukan teks. Nilai medan bersesuaian akan diperoleh dan digunakan. Perhatian oleh kerana lajur komposit adalah medan, anda boleh guna fungsi ini dalam satu medan komposit untuk guna nila bagi beberapa medan komposit yang lain. Ia sangat-sangat berguna ketika membina laluan penyimpanan pembolehubah

switch(val, pattern, value, pattern, value, ..., else_value)

class calibre.utils.formatter_functions.**BuiltinSwitch**

switch(val, pattern, value, pattern, value, ..., else_value) -- bagi setiap *pattern, value* pair, semak jika val sepadan dengan ungkapan nalar *pattern* dan jika ia, kembalikan *value* tersebut. Jika tiada pola dipadankan, maka else_value dikembalikan. Anda boleh adakan sebanyak mana pasangan *pattern, value* yang anda kehendaki

List lookup

_identifier_in_list(val, id, found_val, not_found_val)

class calibre.utils.formatter_functions.**BuiltinIdentifierInList**

_identifier_in_list(val, id, found_val, not_found_val) -- anggap val sebagai senarai pengecam yang dipisah dengan tanda koma, membandingkan rentetan terhadap setiap nilai dalam senarai. Satu pengecam mempunyai format "identifier:value". Parameter id patut sama ada "id" atau "id:regexp". Padanan kata pertama jika terdapat mana-mana pengecam yang id tersebut. Padanan kata kedua jika ungkapan nalar sepadan dengan nilai pengecam. Jika terdapat satu padanan kembalikan found_val, jika tidak kembalikan not_found_val.

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) -- treat val as a list of items separated by separator, evaluating the pattern against each value in the list. If the pattern matches a value, 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.

list_item(val, index, separator)

class calibre.utils.formatter_functions.**BuiltinListitem**

list_item(val, index, separator) -- tafsir nilai sebagai senarai item dipisah dengan *separator*, kembalikan item ke-*index*. Item pertam adalah nombor sifar. Item terakhir boleh dikembalikan menggunakan *list_item(-1,separator)*. Jika item tidak berada dalam senarai, nilai kosong dikembalikan. Pemisah mempunyai maksud yang sama dalam fungsi kiraan.

select(val, key)

class calibre.utils.formatter_functions.**BuiltinSelect**

select(val, key) -- tafsir nilai sebagai senarai item dipisah-tanda-koma, dengan item sebagai "id:value". Cari pasangan dengan id menyamai key, dan kembalikan nilai berkaitan.

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) -- treat val as a list of items separated by separator, comparing the string against each value in the list. If the string matches a value (ignoring case) then return found_val, otherwise return not_found_val. If the string contains separators, then it is also treated as a list and each value is checked. The string and found_value can be repeated as many times as desired, permitting returning different values depending on the search. The strings are checked in order. The first match is returned.

List manipulation

count(val, separator)

class calibre.utils.formatter_functions.**BuiltinCount**

count(val, separator) -- tafsir nilai sebagai senarai item dipisah dengan tanda *separator*, mengembalikan sebilangan item di dalam senarai. Kebanyakan senarai gunakan tanda koma sebagai pemisah, tetapi pengarang boleh gunakan tanda ampersan. Contoh: {tags:count(,)}, {authors:count(&)}

list_difference(list1, list2, separator)

class calibre.utils.formatter_functions.**BuiltinListDifference**

list_difference(list1, list2, separator) -- kembalikan senarai dengan membuang item list1 yang ditemui dalam list2, menggunakan perbandingan tak-sensitif-kata. Item dalam list1 dan list2 diasing dengan tanda-pemisah, begitu juga item dalam senarai dikembalikan.

list_equals(list1, sep1, list2, sep2, yes_val, no_val)

class calibre.utils.formatter_functions.**BuiltinListEquals**

list_equals(list1, sep1, list2, sep2, yes_val, no_val) -- kembalikan yes_val jika list1 dan list2 mengandungi item yang sama, jika tidak kembalikan no_val. Item ditentukan oleh pemisahan setiap senarai menggunakan aksara pemisah yang sesuai (sep1 atau sep2). Tertib item dalam senarai tidak relevan. Perbandingan adalah tidak-sensitif-kata.

list_intersection(list1, list2, separator)

class calibre.utils.formatter_functions.**BuiltinListIntersection**

list_intersection(list1, list2, separator) -- kembalikan senarai dengan membuang item list1 yang ditemui dalam list2, menggunakan perbandingan tak-sensitif-kata. Item dalam list1 dan list2 diasing dengan tanda-pemisah, begitu juga item dalam senarai dikembalikan.

list_re(src_list, separator, include_re, opt_replace)

class calibre.utils.formatter_functions.**BuiltinListRe**

list_re(src_list, separator, include_re, opt_replace) -- Biina senarai dengan pemisahan pertama src_list ke dalam item menggunakan aksara pemisah. Bagi setiap item di dalam senarai, tandakan jika memadani include_re. Jika ia berlaku, maka tambah ia ke dalam senarai yang dikembalikan. Jika opt_replace bukan rentetan kosong, maka laksana penggantian sebelum menambah item ke dalam senarai dikembalikan.

list_re_group(src_list, separator, include_re, search_re, group_1_template, ...)

class calibre.utils.formatter_functions.**BuiltinListReGroup**

list_re_group(src_list, separator, include_re, search_re, group_1_template, ...) -- Seperti list_re pengecualian penggantian bukan pilihan. Ia menggunakan re_group(list_item, search_re, group_1_template, ...) bila membuat penggantian pada senarai hasil.

list_sort(list, direction, separator)

class calibre.utils.formatter_functions.**BuiltinListSort**

list_sort(list, direction, separator) -- kembalikan senarai terisih menggunakan isih sensitif-kata. Jika arah adalah sifat, senarai diisih menaik, jika tidak menurun. Item senarai diasing dengan tanda-pemisah, begitu juga item dalam senarai dikembalikan.

list_union(list1, list2, separator)

class calibre.utils.formatter_functions.**BuiltinListUnion**

list_union(list1, list2, separator) -- kembalikan senarai yang dibuat dengan menggabungkan item di dalam list1 dan list2, membuang item pendua menggunakan perbandingan tak-sensitif-kata. Jika item berbeza kata, yang berada dalam list1 digunakan. Item dalam list1 dan list2 diasing dengan tanda-pemisah, begitu juga item dalam senarai dikembalikan.

subitems(val, start_index, end_index)

class calibre.utils.formatter_functions.**BuiltinSubitems**

subitems(val, start_index, end_index) -- Fungsi ini digunakan untuk pecahkan senarai item seperti genre. Ia tafsir nilai sebagai senarai item dipisah tanda-koma, yang mana setiap item adalah senarai dipisah tanda-noktah. Kembalikan senarai baharu dibuat oleh pencarian pertama semua item dipisah tanda-noktah, kemudian bagi setiap item tersebut, ekstrak komponen *start_index* hingga the *end_index*, seterusnya gabungkan keputusan bersama-sama. Komponen pertama ialah senarai dipisah tanda-noktah yang mempunyai indeks sifar. Jika indeks adalah negatif, maka ia dikira dari penghujung senarai. Dalam kes khas, end_index sifar dianggap sebagai panjang senarai. Contoh penggunaan mod templat asas dan anggap nilai #genre bagi "A.B.C": {#genre:subitems(0,1)} kembalikan "A". {#genre:subitems(0,2)} kembalikan "A.B". {#genre:subitems(1,0)} kembalikan "B.C". Menganggap nilai #genre bagi "A.B.C, D.E.F", {#genre:subitems(0,1)} kembalikan "A, D". {#genre:subitems(0,2)} kembalikan "A.B, D.E"

sublist(val, start_index, end_index, separator)

class calibre.utils.formatter_functions.**BuiltinSublist**

sublist(val, start_index, end_index, separator) -- interpret the value as a list of items separated by *separator*, returning a new list made from the *start_index* to the *end_index* item. 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 using basic template mode and assuming that the tags column (which is comma-separated) contains "A, B, C": {tags:sublist(0,1,\,)} returns "A". {tags:sublist(-1,0,\,)} returns "C". {tags:sublist(0,-1,\,)} returns "A, B".

Other

assign(id, val)

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

assign(id, val) -- umpuk val ke id, kemudian kembalikan val. id mestilah pengecam, bukan ungkapan

print(a, b, ...)

class calibre.utils.formatter_functions.**BuiltinPrint**

print(a, b, ...) -- cetak argumen ke output piawai. Melainkan anda mulakan calibre melalui baris perintah (calibre-debug -g), output akan pergi ke lubang hitam.

Recursion

eval(template)

class calibre.utils.formatter_functions.**BuiltinEval**

eval(template) -- nilaikan templat, melepasi pembolehubah setempat (yang telah diumpuk) selain dari data meta buku. Ia membenarkan penggunaan pemproses templat untuk bina keputusan yang lebih kompleks dari pembolehubah semasa. Oleh kerana aksara { and } adalah istimewa, anda mesti gunakan [[untuk aksara { dan]] untuk aksara }; ia ditukar secara automatik. Perhatian awalan dan akhiran (sintaks |*prefix*|*suffix*) tidak dapat digunakan dalam argumen bagi fungsi ini bila menggunakan mod program templat.

template(x)

class calibre.utils.formatter_functions.**BuiltinTemplate**

template(x) -- nilaikan x sebagai templat. Penilaian ini selesai dalam konteks ia sendiri, bermaksud pembolehubah tidak berkongsi diantara pemanggil dengan penilaian templat. Oleh kerana aksara { and } adalah istimewa, anda mesti gunakan [[untuk aksara { dan]] untuk aksara }; ia ditukar secara automatik. Sebagai contoh, template('[[title_sort]]') akan nilaikan templat {title_sort} dan kembalikan nilainya. Perhatian awalan dan akhiran (sintaks |*prefix*|*suffix*) tidak daapt digunakan dalam argumen bagi fungsi ini bila menggunakan mod program templat.

Relational

cmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinCmp**

cmp(x, y, lt, eq, gt) -- buat perbandingan selepas menukar kedua-dua kepada nombor. Kembalikan lt jika x < y. Kembalikan eq jika x == y. Jika tidak kembalikan gt.

first_matching_cmp(val, cmp1, result1, cmp2, r2, ..., else_result)

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

first_matching_cmp(val, cmp1, result1, cmp2, r2, ..., else_result) -- bandingkan "val < cmpN" dalam jujukan, mengembalikan resultN bagi perbandingan pertama yang berjaya. Kembalikan else_result jika tiada perbandingan yang berjaya. Contohnya: first_matching_cmp(10,5,"small",10,"middle",15,"large","giant") kembalikan "large". Contohnya yang sama dengan nilai pertama 16 kembalikan "giant".

strcmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinStrcmp**

strcmp(x, y, lt, eq, gt) -- buat perbandingan tidak-sensitif-kata bagi x dan y sebagai rentetan. Kembalikan lt jika x < y. Kembalikan eq jika x == y. Jika tidak kembalikan gt.

String case changes

capitalize(val)

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

capitalize(val) -- kembalikan val dibesarkan hurufnya

lowercase(val)

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

lowercase(val) -- kembalikan val dengan huruf kecil

titlecase(val)

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

titlecase(val) -- kembalikan val dengan kata-tajuk

uppercase(val)

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

uppercase(val) -- kembalikan val dengan huruf besar

String manipulation

re(val, pattern, replacement)

class calibre.utils.formatter_functions.**BuiltinRe**

re(val, pattern, replacement) -- return val after applying the regular expression. All instances of *pattern* are replaced with *replacement*. As in all of calibre, these are Python-compatible regular expressions

re_group(val, pattern, template_for_group_1, for_group_2, ...)

class calibre.utils.formatter_functions.**BuiltinReGroup**

re_group(val, pattern, template_for_group_1, for_group_2, ...) -- kembalikan rentetan yang dibuat dengan melaksanakan pola ungkapan nalar pada val dan ganti setiap kejadian yang sepadan dengan rentetan yang dikira dengan menggantikan setiap kumpulan sepadan dengan nilai dikembalikan oleh templat berkenaan. Nilai sepadan asal bagi kumpulan tersedia sebagai $. Dalam mod program templat, seperti untuk templat dan fungsi eval, anda guna [[{ dan]] untuk }. Contoh berikut dalam mod program templat adalah untuk siri lebih dari satu perkataan dan huruf besar perkataan pertama: {series:'re_group($, "(S*)(.*)", "[[$:uppercase()]]", "[[$]]")'}

shorten(val, left chars, middle text, right chars)

class calibre.utils.formatter_functions.**BuiltinShorten**

shorten(val, left chars, middle text, right chars) -- Kembalikan versi pendek val, mengandungi aksara *left chars* dari permulaan val ,diikuti dengan *middle text*, diikuti dengan aksara *right chars* dari penghujung rentetan. *Left chars* dan`right chars` mestilah integer. Contoh, anggap tajuk buku ialah *Ancient English Laws in the Times of Ivanhoe*, dan anda mahu ia dimuat-suakan dengan jarak maksimum 15 aksara. Jika anda guna {title:shorten(9,-,5)}, hasil akan jadi *Ancient E-nhoe*. Jika panjang medan kurang dari aksara kiri + aksara kanan + panjang teks tengah bagi *middle text*, maka medan akan guna yang asal. Contohnya, tajuk *The Dome* tidak akan berubah.

strcat(a, b, ...)

class calibre.utils.formatter_functions.**BuiltinStrcat**

strcat(a, b, ...) -- boleh ambil mana-mana bilangan argumen. Kembalikan rentetan yang terbentuk dengan memangkas semua argumen

strcat_max(max, string1, prefix2, string2, ...)

class `calibre.utils.formatter_functions.`**`BuiltinStrcatMax`**

strcat_max(max, string1, prefix2, string2, ...) -- Kembalikan rentetan yang terbentuk oleh pemangkasan argumen. Nilai dikembalikan adalah diawali pada string1. Pasangan *Prefix, string* ditambah ke penghujung nilai asalkan panjang rentetan yang terhasil adalah kurang dari *max*. String1 dikembalikan walaupun jika string1 lebih panjang berbanding max. Anda boleh lepasi sebanyak mana pasangan *prefix, string* yang anda kehendaki.

strlen(a)

class `calibre.utils.formatter_functions.`**`BuiltinStrlen`**

strlen(a) -- Kembalikan panjang rentetan yang dilepasi sebagai argumen

substr(str, start, end)

class `calibre.utils.formatter_functions.`**`BuiltinSubstr`**

substr(str, start, end) -- kembalikan aksara start hingga end bagi str. Aksara pertama dalam str ialah aksara sifar. Jika penghujung adalah negatif, maka ia menunjukkan kebanyakan aksara dikira dari sebelah kanan. Jika penghujung adalah sifar, maka ia menunjukkan aksara terakhir. Sebagai contoh, substr('12345', 1, 0) kembalikan '2345', dan substr('12345', 1, -1) kembalikan '234'.

swap_around_articles(val, separator)

class `calibre.utils.formatter_functions.`**`BuiltinSwapAroundArticles`**

swap_around_articles(val, separator) -- returns the val with articles moved to the end. The value can be a list, in which case each member of the list is processed. If the value is a list then you must provide the list value separator. If no separator is provided then the value is treated as being a single value, not a list.

swap_around_comma(val)

class `calibre.utils.formatter_functions.`**`BuiltinSwapAroundComma`**

swap_around_comma(val) -- berikan nilai dalam bentuk "B, A", kembalikan "A B". Ia merupakan pertukaran nama paling berguna dalam format LN, FN ke FN LN. Jika tiada tanda koma, fungsi mengembalikan val tanpa perubahan

transliterate(a)

class `calibre.utils.formatter_functions.`**`BuiltinTransliterate`**

transliterate(a) -- Kembalikan rentetan dalam abjad latin terbentuk oleh penganggaran sebutan perkataan dalam rentetan sumber. Sebagai contoh, jika sumber ialah " a " fungsi kembalikan "Fiodor Mikhailovich Dostoievskii".

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=('Tidak Diketahui')*, *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()` (halaman 180) 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` (halaman 181).

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.

```
__docformat__ = 'restructuredtext en'

'''
All fields must have a NULL value represented as None for simple types,
an empty list/dictionary for complex types and (None, None) for cover_data
'''

SOCIAL_METADATA_FIELDS = frozenset((
    'tags',             # Ordered list
    'rating',           # A floating point number between 0 and 10
    'comments',         # A simple HTML enabled string
    'series',           # A simple string
    'series_index',     # A floating point number
    # Of the form { scheme1:value1, scheme2:value2}
    # For example: {'isbn':'123456789', 'doi':'xxxx', ... }
    'identifiers',
))

'''
The list of names that convert to identifiers when in get and set.
'''

TOP_LEVEL_IDENTIFIERS = frozenset((
```

(continues on next page)

(continued from previous page)

```
    '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, for e.g. 'jpeg', 'png', 'gif'...
    'cover_data',
    # Either thumbnail data, or an object with the attribute
    # image_path which is the path to an image file, encoded
    # in filesystem_encoding
    'thumbnail',
))

BOOK_STRUCTURE_FIELDS = frozenset((
    # These are used by code, Null values are None.
    'toc', 'spine', 'guide', 'manifest',
))

USER_METADATA_FIELDS = frozenset((
    # A dict of dicts similar to field_metadata. Each field description dict
    # also contains a value field with the key #value#.
    'user_metadata',
))

DEVICE_METADATA_FIELDS = frozenset((
    'device_collections',   # Ordered list of strings
    'lpath',                # Unicode, / separated
    'size',                 # In bytes
    'mime',                 # Mimetype of the book file being represented
))

CALIBRE_METADATA_FIELDS = frozenset((
    'application_id',   # An application id, currently set to the db_id.
    'db_id',            # the calibre primary key of the item.
    'formats',          # list of formats (extensions) for this book
    # a dict of user category names, where the value is a list of item names
```

(continues on next page)

(continued from previous page)

```
    # from the book that are in that category
    'user_categories',
    # a dict of author to an associated hyperlink
    'author_link_map',
))

ALL_METADATA_FIELDS =      SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           USER_METADATA_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)

# All fields except custom fields
STANDARD_METADATA_FIELDS = SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)

# Metadata fields that smart update must do special processing to copy.
SC_FIELDS_NOT_COPIED =     frozenset(('title', 'title_sort', 'authors',
                                      'author_sort', 'author_sort_map',
                                      'cover_data', 'tags', 'languages',
                                      'identifiers'))

# Metadata fields that smart update should copy only if the source is not None
SC_FIELDS_COPY_NOT_NULL =  frozenset(('device_collections', 'lpath', 'size', 'comments
↪', 'thumbnail'))

# Metadata fields that smart update should copy without special handling
SC_COPYABLE_FIELDS =       SOCIAL_METADATA_FIELDS.union(
                           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 Semua berkenaan penggunaan ungkapan nalar di dalam calibre

Ungkapan nalar merupakan fitur yang banyak digunakan di dalam calibre untuk membuat manipulasi rumit bagi kandungan serta data meta e-buku. Tutorial ini merupakan pengenalan asasi untuk anda memulakan penggunaan ungkapan nalar di dalam calibre.

Kandungan

- *Pertama sekali, untuk pembakar semangat dan peringatan mesra* (halaman 184)
- *Kenapa calibre menggunakan ungkapan nalar?* (halaman 185)
- *Apa* itu *ungkapan nalar atau regular expression?* (halaman 185)
- *Teruskan penjelasan?* (halaman 185)
- *Ok nampak jelas, berikutnya?* (halaman 185)
- *Hey, neat! This is starting to make sense!* (halaman 186)
- *Well, these special characters are very neat and all, but what if I wanted to match a dot or a question mark?* (halaman 186)
- *So, what are the most useful sets?* (halaman 186)
- *But if I had a few varying strings I wanted to match, things get complicated?* (halaman 187)
- *You missed...* (halaman 187)
- *In the beginning, you said there was a way to make a regular expression case insensitive?* (halaman 187)
- *I think I'm beginning to understand these regular expressions now... how do I use them in calibre?* (halaman 188)
  - *Conversions* (halaman 188)
  - *Menambah buku* (halaman 188)
  - *Bulk editing metadata* (halaman 189)
- *Rujukan pantas* (halaman 189)
- *Penghargaan* (halaman 194)

10.4.1 Pertama sekali, untuk pembakar semangat dan peringatan mesra

Perbincangan ini akan menjadi lebih teknikal, ungkapan nalar atau regular expression ialah alat teknikal untuk membuat pekara-perkara teknikal :P. Terdapat beberapa istilah atau jargon dan konsep mungkin rumit atau berbelit (tak pe kita teruskan). Walaubagaimanapun, penerangan akan dibuat semudah yang mungkin, kerana istilah-istilah ini wajib digunakan. Jom kita teruskan.

10.4.2 Kenapa calibre menggunakan ungkapan nalar?

There are a few places calibre uses regular expressions. There's the *Search & replace* in conversion options, metadata detection from filenames in the import settings and Search & replace when editing the metadata of books in bulk. The calibre book editor can also use regular expressions in its *Search and replace* feature. Finally, you can use regular expressions when searching the calibre book list and when searching inside the calibre E-book viewer.

10.4.3 Apa *itu* ungkapan nalar atau regular expression?

Ungkapan nalar ialah satu cara menjelaskan set-set rentetan. Ungkapan nalar boleh *padankan* sejumlah rentetan yang berbeza. Ia menjadikan ungkapan nalar sangat hebat -- ia merupakan cara teringkas dan padat untuk menjelaskan sejumlah besar variasi.

Note: Retetan yang digunakan di sini adalah berbentuk bahasa pengaturcaraan: satu rentetan yang mengandungi lebih daripada satu aksara termasuklah aksara, tanda baca dan juga ruang putih atau whitespace (hentian baris, tabulator dan lain-lain). Perhatian aksara huruf besar dan kecil juga tidak dianggap sama, maka "a" adalah berbeza dangan "A". Dalam calibre, ungkapan nalar adalah peka kata, yang mana akan dibincangkan kemudian. Ia menjadi lebih rumit kerana ungkapan nalar membenarkan variasi dalam rentetan yang disepadankan, jadi satu ungkapan boleh dipadankan dengan berbilang rentetan.

10.4.4 Teruskan penjelasan?

Konsep terpenting dalam ungkapan nalar: *rentetan A itu sendiri ialah ungkapan nalar yang disepadankan dengan dirinya sendiri*/ Jika mahu padankan dengan rentetan *"Hello, World!"*` menggunakan ungkapan nalar, ia menjadi `Hello, World!`. Ya nampak mudah. Anda dapati ia *hanya* disepadankan dengan rentetan "`Hello, World!`" dengan tepat, bukannya "`Hello, wOrld!`" atau "`hello, world!`" atau lain-lain variasi sebegitu.

10.4.5 Ok nampak jelas, berikutnya?

Next is the beginning of the really good stuff. Remember where I said that regular expressions can match multiple strings? This is where it gets a little more complicated. Say, as a somewhat more practical exercise, the e-book you wanted to convert had a nasty footer counting the pages, like "Page 5 of 423". Obviously the page number would rise from 1 to 423, thus you'd have to match 423 different strings, right? Wrong, actually: regular expressions allow you to define sets of characters that are matched: To define a set, you put all the characters you want to be in the set into square brackets. So, for example, the set `[abc]` would match either the character "a", "b" or "c". *Sets will always only match one of the characters in the set*. They "understand" character ranges, that is, if you wanted to match all the lower case characters, you'd use the set `[a-z]` for lower- and uppercase characters you'd use `[a-zA-Z]` and so on. Got the idea? So, obviously, using the expression `Page [0-9] of 423` you'd be able to match the first 9 pages, thus reducing the expressions needed to three: The second expression `Page [0-9][0-9] of 423` would match all two-digit page numbers, and I'm sure you can guess what the third expression would look like. Yes, go ahead. Write it down.

10.4.6 Hey, neat! This is starting to make sense!

I was hoping you'd say that. But brace yourself, now it gets even better! We just saw that using sets, we could match one of several characters at once. But you can even repeat a character or set, reducing the number of expressions needed to handle the above page number example to one. Yes, ONE! Excited? You should be! It works like this: Some so-called special characters, "+", "?" and "*", *repeat the single element preceding them.* (Element means either a single character, a character set, an escape sequence or a group (we'll learn about those last two later)- in short, any single entity in a regular expression.) These characters are called wildcards or quantifiers. To be more precise, "?" matches *0 or 1* of the preceding element, "*" matches *0 or more* of the preceding element and "+" matches *1 or more* of the preceding element. A few examples: The expression `a?` would match either "" (which is the empty string, not strictly useful in this case) or "a", the expression `a*` would match "", "a", "aa" or any number of a's in a row, and, finally, the expression `a+` would match "a", "aa" or any number of a's in a row (Note: it wouldn't match the empty string!). Same deal for sets: The expression `[0-9]+` would match *every integer number there is*! I know what you're thinking, and you're right: If you use that in the above case of matching page numbers, wouldn't that be the single one expression to match all the page numbers? Yes, the expression `Page [0-9]+ of 423` would match every page number in that book!

Note: A note on these quantifiers: They generally try to match as much text as possible, so be careful when using them. This is called "greedy behaviour"- I'm sure you get why. It gets problematic when you, say, try to match a tag. Consider, for example, the string `"<p class="calibre2">Title here</p>"` and let's say you'd want to match the opening tag (the part between the first pair of angle brackets, a little more on tags later). You'd think that the expression `<p.*>` would match that tag, but actually, it matches the whole string! (The character "." is another special character. It matches anything *except* linebreaks, so, basically, the expression `.*` would match any single line you can think of.) Instead, try using `<p.*?>` which makes the quantifier `"*"` non-greedy. That expression would only match the first opening tag, as intended. There's actually another way to accomplish this: The expression `<p[^>]*>` will match that same opening tag- you'll see why after the next section. Just note that there quite frequently is more than one way to write a regular expression.

10.4.7 Well, these special characters are very neat and all, but what if I wanted to match a dot or a question mark?

You can of course do that: Just put a backslash in front of any special character and it is interpreted as the literal character, without any special meaning. This pair of a backslash followed by a single character is called an escape sequence, and the act of putting a backslash in front of a special character is called escaping that character. An escape sequence is interpreted as a single element. There are of course escape sequences that do more than just escaping special characters, for example `"\t"` means a tabulator. We'll get to some of the escape sequences later. Oh, and by the way, concerning those special characters: Consider any character we discuss in this introduction as having some function to be special and thus needing to be escaped if you want the literal character.

10.4.8 So, what are the most useful sets?

Knew you'd ask. Some useful sets are `[0-9]` matching a single number, `[a-z]` matching a single lowercase letter, `[A-Z]` matching a single uppercase letter, `[a-zA-Z]` matching a single letter and `[a-zA-Z0-9]` matching a single letter or number. You can also use an escape sequence as shorthand:

```
\d is equivalent to [0-9]
\w is equivalent to [a-zA-Z0-9_]
\s is equivalent to any whitespace
```

Note: "Whitespace" is a term for anything that won't be printed. These characters include space, tabulator, line feed, form feed and carriage return.

As a last note on sets, you can also define a set as any character *but* those in the set. You do that by including the character "`^`" as the *very first character in the set*. Thus, `[^a]` would match any character excluding "a". That's called complementing the set. Those escape sequence shorthands we saw earlier can also be complemented: "`\D`" means any non-number character, thus being equivalent to `[^0-9]`. The other shorthands can be complemented by, you guessed it, using the respective uppercase letter instead of the lowercase one. So, going back to the example `<p[^>]*>` from the previous section, now you can see that the character set it's using tries to match any character except for a closing angle bracket.

10.4.9 But if I had a few varying strings I wanted to match, things get complicated?

Fear not, life still is good and easy. Consider this example: The book you're converting has "Title" written on every odd page and "Author" written on every even page. Looks great in print, right? But in e-books, it's annoying. You can group whole expressions in normal parentheses, and the character "`|`" will let you match *either* the expression to its right *or* the one to its left. Combine those and you're done. Too fast for you? Okay, first off, we group the expressions for odd and even pages, thus getting `(Title)(Author)` as our two needed expressions. Now we make things simpler by using the vertical bar ("`|`" is called the vertical bar character): If you use the expression `(Title|Author)` you'll either get a match for "Title" (on the odd pages) or you'd match "Author" (on the even pages). Well, wasn't that easy?

You can, of course, use the vertical bar without using grouping parentheses, as well. Remember when I said that quantifiers repeat the element preceding them? Well, the vertical bar works a little differently: The expression "Title|Author" will also match either the string "Title" or the string "Author", just as the above example using grouping. *The vertical bar selects between the entire expression preceding and following it*. So, if you wanted to match the strings "Calibre" and "calibre" and wanted to select only between the upper- and lowercase "c", you'd have to use the expression `(c|C)alibre`, where the grouping ensures that only the "c" will be selected. If you were to use `c|Calibre`, you'd get a match on the string "c" or on the string "Calibre", which isn't what we wanted. In short: If in doubt, use grouping together with the vertical bar.

10.4.10 You missed...

... wait just a minute, there's one last, really neat thing you can do with groups. If you have a group that you previously matched, you can use references to that group later in the expression: Groups are numbered starting with 1, and you reference them by escaping the number of the group you want to reference, thus, the fifth group would be referenced as `\5`. So, if you searched for `([^ ]+) \1` in the string "Test Test", you'd match the whole string!

10.4.11 In the beginning, you said there was a way to make a regular expression case insensitive?

Yes, I did, thanks for paying attention and reminding me. You can tell calibre how you want certain things handled by using something called flags. You include flags in your expression by using the special construct `(?flags go here)` where, obviously, you'd replace "flags go here" with the specific flags you want. For ignoring case, the flag is `i`, thus you include `(?i)` in your expression. Thus, `(?i)test` would match "Test", "tEst", "TEst" and any case variation you could think of.

Another useful flag lets the dot match any character at all, *including* the newline, the flag `s`. If you want to use multiple flags in an expression, just put them in the same statement: `(?is)` would ignore case and make the dot match all. It doesn't matter which flag you state first, `(?si)` would be equivalent to the above.

10.4.12 I think I'm beginning to understand these regular expressions now... how do I use them in calibre?

Conversions

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). You'd have to remove some of the tags as well. In this example, I'd recommend beginning with the tag `<b class="calibre2">`, now you have to end with the corresponding closing tag (opening tags are `<tag>`, closing tags are `</tag>`), which is simply the next `</b>` in this case. (Refer to a good HTML manual or ask in the forum if you are unclear on this point.) The opening tag can be described using `<b.*?>`, the closing tag using `</b>`, thus we could remove everything between those tags using `<b.*?>.*?</b>`. But using this expression would be a bad idea, because it removes everything enclosed by <b>- tags (which, by the way, render the enclosed text in bold print), and it's a fair bet that we'll remove portions of the book in this way. Instead, include the beginning of the enclosed string as well, making the regular expression `<b.*?>\s*Generated\s+by\s+ABC\s+Amber\s+LIT.*?</b>` The `\s` with quantifiers are included here instead of explicitly using the spaces as seen in the string to catch any variations of the string that might occur. Remember to check what calibre will remove to make sure you don't remove any portions you want to keep if you test a new expression. If you only check one occurrence, you might miss a mismatch somewhere else in the text. Also note that should you accidentally remove more or fewer tags than you actually wanted to, calibre tries to repair the damaged code after doing the removal.

Menambah buku

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.

Bulk editing metadata

The last part is regular expression *Search and replace* in metadata fields. You can access this by selecting multiple books in the library and using bulk metadata edit. Be very careful when using this last feature, as it can do **Very Bad Things** to your library! Doublecheck that your expressions do what you want them to using the test fields, and only mark the books you really want to change! In the regular expression search mode, you can search in one field, replace the text with something and even write the result into another field. A practical example: Say your library contained the books of Frank Herbert's Dune series, named after the fashion `Dune 1 - Dune`, `Dune 2 - Dune Messiah` and so on. Now you want to get `Dune` into the series field. You can do that by searching for `(.*?) \d+ - .*` in the title field and replacing it with `\1` in the series field. See what I did there? That's a reference to the first group you're replacing the series field with. Now that you have the series all set, you only need to do another search for `.*? -` in the title field and replace it with `""` (an empty string), again in the title field, and your metadata is all neat and tidy. Isn't that great? By the way, instead of replacing the entire field, you can also append or prepend to the field, so, if you *wanted* the book title to be prepended with series info, you could do that as well. As you by now have undoubtedly noticed, there's a checkbox labeled *Case sensitive*, so you won't have to use flags to select behaviour here.

Well, that just about concludes the very short introduction to regular expressions. Hopefully I'll have shown you enough to at least get you started and to enable you to continue learning by yourself- a good starting point would be the Python documentation for regexps.

Amaran terakhir, walaupun: Regexps hebat, tetapi sangat mudah tersalah langkah. calibre menyediakan keupayaan pengujian yang terbaik jika mahu melihat ungkapan anda berkelakuan sepertimana yang dikehendaki. Sila gunakannya. Akan tetapi cuba elakkan kesilapan-kesilapan. Sepatutnya anda teruskan usaha dan cuba pelajari darinya.

10.4.13 Rujukan pantas

Quick reference for regexp syntax

This checklist summarizes the most commonly used/hard to remember parts of the regexp engine available in most parts of calibre.

Kandungan

- *Character classes* (halaman 190)
- *Shorthand character classes* (halaman 190)
- *The quantifiers* (halaman 190)
- *Greed* (halaman 191)
- *Alternation* (halaman 191)
- *Exclusion* (halaman 191)
- *Anchors* (halaman 191)
- *Groups* (halaman 192)
- *Lookarounds* (halaman 192)
- *Recursion* (halaman 193)
- *Special characters* (halaman 193)
- *Meta-characters* (halaman 193)
- *Modes* (halaman 194)

Character classes

Character classes are useful to represent different groups of characters, succinctly.

Examples:

Representation	Class
`[a-z]`	Lowercase letters. Does not include characters with accent mark and ligatures
`[a-z0-9]`	Lowercase letters from a to z or numbers from 0 to 9
`[A-Za-z-]`	Uppercase or lowercase letters, or a dash. To include the dash in a class, you must put it at the beginning or at the end so as not to confuse it with the hyphen that specifies a range of characters
`[^0-9]`	Any character except a digit. The caret (^) placed at the beginning of the class excludes the characters of the class (complemented class)
`[[a-z]--[aeiouy]]`	The lowercase consonants. A class can be included in a class. The characters `--` exclude what follows them
`[\w--[\d_]]`	All letters (including foreign accented characters). Abbreviated classes can be used inside a class

Example:

```
<[^<>]+> to select an HTML tag
```

Shorthand character classes

Representation	Class
`\d`	A digit (same as `[0-9]`)
`\D`	Any non-numeric character (same as `[^0-9]`)
`\w`	An alphanumeric character (`[a-zA-Z0-9]`) including characters with accent mark and ligatures
`\W`	Any non-word character
`\s`	Space, non-breaking space, tab, return line
`\S`	Any non-whitespace character
`.`	Any character except newline. Use the dot all checkbox or the `(?s)` regexp modifier to include the newline character.

The quantifiers

Quantifier	Number of occurrences of the expression preceding the quantifier
`?`	0 or 1 occurrence of the expression. Same as `{0,1}`
`+`	1 or more occurrences of the expression. Same as `{1,}`
`*`	0, 1 or more occurrences of the expression. Same as `{0,}`
`{n}`	Exactly n occurrences of the expression
`{min,max}`	Number of occurrences between the minimum and maximum values included
`{min,}`	Number of occurrences between the minimum value included and the infinite
`{,max}`	Number of occurrences between 0 and the maximum value included

Greed

By default, with quantifiers, the regular expression engine is greedy: it extends the selection as much as possible. This often causes surprises, at first. `?` follows a quantifier to make it lazy. Avoid putting two in the same expression, the result can be unpredictable.

Beware of nesting quantifiers, for example, the pattern `(a*)*`, as it exponentially increases processing time.

Alternation

The `|` character in a regular expression is a logical `OR`. It means that either the preceding or the following expression can match.

Exclusion

Methodă1

`pattern_to_exclude(*SKIP)(*FAIL)|pattern_to_select`

Example:

`"Blabla"(*SKIP)(*FAIL)|Blabla`

selects Blabla, in the strings Blabla or "Blabla or Blabla", but not in "Blabla".

Methodă2

`pattern_to_exclude\K|(pattern_to_select)`

`"Blabla"\K|(Blabla)`

selects Blabla, in the strings Blabla or "Blabla or Blabla", but not in "Blabla".

Anchors

An anchor is a way to match a logical location in a string, rather than a character. The most useful anchors for text processing are:

\b Designates a word boundary, i.e. a transition from space to non-space character. For example, you can use `\bsurd` to match `the surd` but not `absurd`.

^ Matches the start of a line (in multi-line mode, which is the default)

$ Matches the end of a line (in multi-line mode, which is the default)

\K Resets the start position of the selection to its position in the pattern. Some regexp engines (but not calibre) do not allow lookbehind of variable length, especially with quantifiers. When you can use `\K` with these engines, it also allows you to get rid of this limit by writing the equivalent of a positive lookbehind of variable length.

Groups

`(expression)` Capturing group, which stores the selection and can be recalled later in the *search* or *replace* patterns with `\n`, where `n` is the sequence number of the capturing group (starting at 1 in reading order)

`(?:expression)` Group that does not capture the selection

`(?>expression)` Atomic Group: As soon as the expression is satisfied, the regexp engine passes, and if the rest of the pattern fails, it will not backtrack to try other combinations with the expression. Atomic groups do not capture.

`(?|expression)` Branch reset group: the branches of the alternations included in the expression share the same group numbers

`(?<name>expression)` Group named name. The selection can be recalled later in the *search* pattern by `(?P=name)` and in the *replace* by `\g<name>`. Two different groups can use the same name.

Lookarounds

Lookaround	Meaning
`?=`	Positive lookahead (to be placed after the selection)
`?!`	Negative lookahead (to be placed after the selection)
`?<=`	Positive lookbehind (to be placed before the selection)
`?<!`	Negative lookbehind (to be placed before the selection)

Lookaheads and lookbehinds do not consume characters, they are zero length and do not capture. They are atomic groups: as soon as the assertion is satisfied, the regexp engine passes, and if the rest of the pattern fails, it will not backtrack inside the lookaround to try other combinations.

When looking for multiple matches in a string, at the starting position of each match attempt, a lookbehind can inspect the characters before the current position. Therefore, on the string 123, the pattern `(?<=\d)\d` (a digit preceded by a digit) should, in theory, select 2 and 3. On the other hand, `\d\K\d` can only select 2, because the starting position after the first selection is immediately before 3, and there are not enough digits for a second match. Similarly, `\d(\d)` only captures 2. In calibre's regexp engine practice, the positive lookbehind behaves in the same way, and selects only 2, contrary to theory.

Groups can be placed inside lookarounds, but capture is rarely useful. Nevertheless, if it is useful, it will be necessary to be very careful in the use of a quantifier in a lookbehind: the greed associated with the absence of backtracking can give a surprising capture. For this reason, use `\K` rather than a positive lookbehind when you have a quantifier (or worse, several) in a capturing group of the positive lookbehind.

Example of negative lookahead:

```
(?![^<>{}]*[>}])
```

Placed at the end of the pattern prevents to select within a tag or a style embedded in the file.

Whenever possible, it is always better to "anchor" the lookarounds, to reduce the number of steps necessary to obtain the result.

Recursion

Representation	Meaning
`(?R)`	Recursion of the entire pattern
`(?1)`	Recursion of the only pattern of the numbered capturing group, here group 1

Recursion is calling oneself. This is useful for balanced queries, such as quoted strings, which can contain embedded quoted strings. Thus, if during the processing of a string between double quotation marks, we encounter the beginning of a new string between double quotation marks, well we know how to do, and we call ourselves. Then we have a pattern like:

```
start-pattern(?>atomic sub-pattern|(?R))*end-pattern
```

To select a string between double quotation marks without stopping on an embedded string:

```
((?>[^]+|(?R))*[^]+)
```

This template can also be used to modify pairs of tags that can be embedded, such as `<div>` tags.

Special characters

Representation	Character
`\t`	tabulation
`\n`	line break
`\x20`	(breakable) space
`\xa0`	no-break space

Meta-characters

Meta-characters are those that have a special meaning for the regexp engine. Of these, twelve must be preceded by an escape character, the backslash (\), to lose their special meaning and become a regular character again:

```
^ . [ ] $ ( ) * + ? | \
```

Seven other meta-characters do not need to be preceded by a backslash (but can be without any other consequence):

```
{ } ! < > = :
```

Special characters lose their status if they are used inside a class (between brackets `[]`). The closing bracket and the dash have a special status in a class. Outside the class, the dash is a simple literal, the closing bracket remains a meta-character.

The slash (/) and the number sign (or hash character) (#) are not meta-characters, they dont need to be escaped.

In some tools, like regex101.com with the Python engine, double quotes have the special status of separator, and must be escaped, or the options changed. This is not the case in the editor of calibre.

Modes

(?s) Causes the dot (.) to match newline characters as well

(?m) Makes the ^ and $ anchors match the start and end of lines instead of the start and end of the entire string.

10.4.14 Penghargaan

Terima kasih atas bantuan petua, pembetulan dan juga:

- ldolse
- kovidgoyal
- chaley
- dwanthny
- kacir
- Starson17
- Orpheu

For more about regexps see The Python User Manual. The actual regular expression library used by calibre is: regex which supports several useful enhancements over the python standard library one.

10.5 Menulis pemalam anda sendiri untuk menambahbaik kefungsian calibre

calibre mempunyai reka bentuk yang sangat modular. Hampir semua kefungsian dalam calibre berasal dalam bentuk pemalam. Pemalam digunakan untuk pertukaran format, untuk memuat turun berita (dikenali sebagai resepi), untuk pelbagai komponen antaramuka pengguna, untuk sambung ke peranti yang berbeza, untuk proses fail bila menambahnya ke calibre dan seterusnya. Anda boleh dapatkan senarai semua pemalam terbina-dalam di dalam calibre dengan pergi ke *Keutamaan→Lanjutan→Pemalam*.

Here, we will teach you how to create your own plugins to add new features to calibre.

Kandungan

- *Anatomy of a calibre plugin* (halaman 195)
- *A User Interface plugin* (halaman 196)
  - *__init__.py* (halaman 197)
  - *ui.py* (halaman 198)
  - *main.py* (halaman 199)
  - *Getting resources from the plugin ZIP file* (halaman 202)
  - *Enabling user configuration of your plugin* (halaman 202)
- *Edit book plugins* (halaman 204)
  - *main.py* (halaman 205)

- *Running User Interface plugins in a separate process* (halaman 207)
- *Adding translations to your plugin* (halaman 208)
- *The plugin API* (halaman 209)
- *Debugging plugins* (halaman 209)
- *More plugin examples* (halaman 209)
- *Sharing your plugins with others* (halaman 209)

Note: This only applies to calibre releases >= 0.8.60

10.5.1 Anatomy of a calibre plugin

A calibre plugin is very simple, it's just a ZIP file that contains some Python code and any other resources like image files needed by the plugin. Without further ado, let's see a basic example.

Suppose you have an installation of calibre that you are using to self publish various e-documents in EPUB and MOBI formats. You would like all files generated by calibre to have their publisher set as "Hello world", here's how to do it. Create a file named `__init__.py` (this is a special name and must always be used for the main file of your plugin) and enter the following Python code into it:

```
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          = set(['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
        file = open(path_to_ebook, 'r+b')
        ext  = os.path.splitext(path_to_ebook)[-1][1:].lower()
        mi = get_metadata(file, ext)
        mi.publisher = 'Hello World'
        set_metadata(file, mi, ext)
        return path_to_ebook
```

That's all. To add this code to calibre as a plugin, simply run the following in the directory in which you created `__init__.py`:

```
calibre-customize -b .
```

Note: Pada MacOS, alat baris perintah disediakan dalam berkas calibre, contohnya, jika anda memasang calibre di dalam `/Applications` alat baris perintah berada di dalam `/Applications/calibre.app/Contents/MacOS/`.

You can download the Hello World plugin from helloworld_plugin.zip.

Every time you use calibre to convert a book, the plugin's `run()` method will be called and the converted book will have its publisher set to "Hello World". This is a trivial plugin, lets move on to a more complex example that actually adds a component to the user interface.

10.5.2 A User Interface plugin

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

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-directory), 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 This file defines the interface part of the plugin

images/icon.png The icon for this plugin

about.txt A text file with information about the plugin

translations A folder containing .mo files with the translations of the user interface of your plugin into different languages. See below for details.

Now let's look at the code.

__init__.py

First, the obligatory `__init__.py` to define the plugin metadata:

```python
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)

(continued from previous page)

```
        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.
        #
        # Note that if you are loading more than one icon, for performance, you
```

(continues on next page)

(continued from previous page)

```
        # 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')

        # 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)

(continued from previous page)

```
        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)

(continued from previous page)

```
        # 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)

(continued from previous page)

```
        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 directory 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 dict mapping the names to bytes. If a name is not found, it will not be present in the returned dict.

get_icons(name_or_list_of_names) A convenience 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.

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
        aborted.

        The base class implementation of this method raises NotImplementedError
```

(continues on next page)

(continued from previous page)

```
        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)
        self.label.setBuddy(self.msg)

```

(continues on next page)

(continued from previous page)

```
    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.

The first step, as for all plugins is to create the import name empty txt file, as described *above* (halaman 196). 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* (halaman 326) 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* (halaman 326) class for more details.

main.py

Here we will see the definition of a single tool that will multiply all font sizes in the book by a number provided by the user. This tool demonstrates various important concepts that you will need in developing your own plugins, so you should read the (heavily commented) source code carefully.

```
import re
from PyQt5.Qt import QAction, QInputDialog
from css_parser.css import CSSRule

# The base class that all tools must inherit from
from calibre.gui2.tweak_book.plugin import Tool

from calibre import force_unicode
from calibre.gui2 import error_dialog
from calibre.ebooks.oeb.polish.container import OEB_DOCS, OEB_STYLES, serialize


class DemoTool(Tool):

    #: Set this to a unique name it will be used as a key
    name = 'demo-tool'

    #: If True the user can choose to place this tool in the plugins toolbar
    allowed_in_toolbar = True

    #: If True the user can choose to place this tool in the plugins menu
    allowed_in_menu = True

    def create_action(self, for_toolbar=True):
        # Create an action, this will be added to the plugins toolbar and
        # the plugins menu
        ac = QAction(get_icons('images/icon.png'), 'Magnify fonts', self.gui)  # noqa
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
            # register it for the action created for the menu, not the toolbar,
            # to avoid a double trigger
            self.register_shortcut(ac, 'magnify-fonts-tool', default_keys=(
↪'Ctrl+Shift+Alt+D',))
        ac.triggered.connect(self.ask_user)
        return ac

    def ask_user(self):
        # Ask the user for a factor by which to multiply all font sizes
        factor, ok = QInputDialog.getDouble(
            self.gui, 'Enter a magnification factor', 'Allow font sizes in the book␣
↪will be multiplied by the specified factor',
            value=2, min=0.1, max=4
        )
        if ok:
            # Ensure any in progress editing the user is doing is present in the␣
↪container
            self.boss.commit_all_editors_to_container()
            try:
                self.magnify_fonts(factor)
            except Exception:
                # Something bad happened report the error to the user
                import traceback
```

(continues on next page)

(continued from previous page)

```
                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)

(continued from previous page)

```
        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()* (halaman 327). 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* (halaman 327) class.

Finally, there is `self.current_container` which is a reference to the book being edited as a *calibre.ebooks.oeb.polish.container.Container* (halaman 319) object. This represents the book as a collection of its constituent HTML/CSS/image files and has convenience methods for doing many useful things. The container object and various useful utility functions that can be reused in your plugin code are documented in *API documentation for the e-book editing tools* (halaman 319).

10.5.4 Running User Interface plugins in a separate process

If you are writing a user interface plugin that needs to make use of Qt WebEngine, it cannot be run in the main calibre process as it is not possible to use WebEngine there. Instead you can copy the data your plugin needs to a temporary directory and run the plugin with that data in a separate process. A simple example plugin follows that shows how to do this.

You can download the plugin from webengine_demo_plugin.zip.

The important part of the plugin is in two functions:

```

    def show_dialog(self):
        # Ask the user for a URL
        url, ok = QInputDialog.getText(self.gui, 'Enter a URL', 'Enter a URL to␣
↪browse below', text='https://calibre-ebook.com')
```

(continues on next page)

(continued from previous page)

```
        if not ok or not url:
            return
        # Launch a separate process to view the URL in WebEngine
        self.gui.job_manager.launch_gui_app('webengine-dialog', kwargs={
            'module':'calibre_plugins.webengine_demo.main', 'url':url})
```

```
def main(url):
    # This function is run in a separate process and can do anything it likes,
    # including use QWebEngine. Here it simply opens the passed in URL
    # in a QWebEngineView
    app = Application([])
    w = QWebEngineView()
    w.setUrl(QUrl(url))
    w.show()
    w.raise_()
    app.exec_()
```

The `show_demo()` function asks the user for a URL and then runs the `main()` function passing it that URL. The `main()` function displays the URL in a `QWebEngineView`.

10.5.5 Adding translations to your 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 program for this.

Send these .po files to your translators. Once you get them back, compile them into .mo files. You can again use Poedit for that, or just do:

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

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 like this:

```
CALIBRE_OVERRIDE_LANG=de calibre
```

Replace `de` with the language code of the language you want to test.

10.5.6 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 *API documentation for plugins* (halaman 225).

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 *Bentangan kod* (halaman 306).

10.5.7 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/directory; calibre
```

This will shutdown a running calibre, wait for the shutdown to complete, then update your plugin in calibre and relaunch calibre.

10.5.8 More plugin examples

You can find a list of many, sophisticated calibre plugins here.

10.5.9 Sharing your plugins with others

If you would like to share the plugins you have created with other users of calibre, post your plugin in a new thread in the calibre plugins forum.

10.6 Mengatur Huruf Matematik di dalam e-buku

Pelihat E-buku calibre mempunyai keupayaan untuk memaparkan ungkapan matematik yang terbenam dalam e-buku (fail ePUB dan HTML). Anda boleh mengatur huruf matematik secara terus dengan TeX atau MathML atau AsciiMath. Pelihat calibre menggunakan pustaka MathJax yang sesuai untuk memaparkannya. Tutorial ini merupakan tutorial ringkas bagaimana mencipta e-buku dengan ungkapan matematik yang mana dapat dipapar dengan baik dengan pelihat E-buku calibre.

10.6.1 Fail HTML mudah yang ada ungkapan matematik

Anda boleh tulis ungkapan matematik di dalam fail HTML mudah dan pelihat calibre akan menerapkannya dalam aturan huruf matematik yang dikehendaki. Contohnya seperti di bawah, kita gunakan notasi TeX untuk ungkapan matematik. Didapati anda boleh guna perintah TeX biasa, dengan tanda caveat kecil, ampersand, tanda lebih besar atau kurang dari yang boleh ditulis sebagai &amp; &lt; dan juga &gt;.

Langkah pertama untuk memberitahu calibre terdapat ungkapan matematik. Apa yang perlu dibuat adalah dengan menambah kod snippet ke dalam seksyen <head> dalam fail HTML:

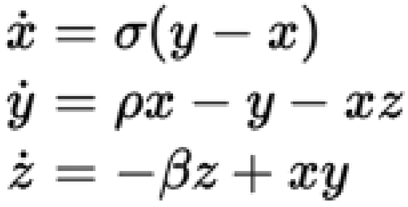
```
<script type="text/x-mathjax-config"></script>
```

Begitulah, kini anda boleh menaip ungkapan matematik seperti mana anda buat dalam fail .tex/ Contohnya, di bawah merupakan persamaan Lorentz:

```
<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} &amp; = \sigma(y-x) \\
\dot{y} &amp; = \rho x - y - xz \\
\dot{z} &amp; = -\beta z + xy
\end{align}
</p>
```

Snippet ini kelihatan seperti cekupan skrin berikut dalam pelihat E-buku calibre.



$$\begin{aligned}\dot{x} &= \sigma(y-x)\\ \dot{y} &= \rho x - y - xz\\ \dot{z} &= -\beta z + xy\end{aligned}$$

Figure1: *Persamaan Lorenz*

Fail HTML lengkap, lebih banyak persamaan dan garis-dalaman matematik yang terhasil seperti dibawah. Anda boleh tukar fail HTML menjadi EPUB di dalam calibre yang mana e-buku yang mudah diedar kepada orang lain.

```
<!DOCTYPE html>
<html>
<!-- Copyright (c) 2012 Design Science, Inc. -->
<head>
<title>Math Test Page</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<!-- This script tag is needed to make calibre's ebook-viewer recpgnize that this␣
↪file needs math typesetting -->
<script type="text/x-mathjax-config">
    // This line adds numbers to all equations automatically, unless explicitly␣
↪suppressed.
    MathJax.tex = {tags: 'all'};
</script>

```

(continues on next page)

(continued from previous page)

```
<style>
h1 {text-align:center}
h2 {
  font-weight: bold;
  background-color: #DDDDDD;
  padding: .2em .5em;
  margin-top: 1.5em;
  border-top: 3px solid #666666;
  border-bottom: 2px solid #999999;
}
</style>
</head>
<body>

<h1>Sample Equations</h1>

<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} &amp; = \sigma(y-x) \label{lorenz}\\
\dot{y} &amp; = \rho x - y - xz \\
\dot{z} &amp; = -\beta z + xy
\end{align}
</p>

<h2>The Cauchy-Schwarz Inequality</h2>

<p>\[
\left( \sum_{k=1}^n a_k b_k \right)^{\!\!2} \leq
 \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
\]</p>

<h2>A Cross Product Formula</h2>

<p>\[
  \mathbf{V}_1 \times \mathbf{V}_2 =
   \begin{vmatrix}
    \mathbf{i} &amp; \mathbf{j} &amp; \mathbf{k} \\
    \frac{\partial X}{\partial u} &amp; \frac{\partial Y}{\partial u} &amp; 0 \\
    \frac{\partial X}{\partial v} &amp; \frac{\partial Y}{\partial v} &amp; 0 \\
   \end{vmatrix}
\]</p>

<h2>The probability of getting \(k\) heads when flipping \(n\) coins is:</h2>

<p>\[P(E) = {n \choose k} p^k (1-p)^{ n-k} \]</p>

<h2>An Identity of Ramanujan</h2>

<p>\[
   \frac{1}{(\sqrt{\phi \sqrt{5}}-\phi) e^{\frac25 \pi}} =
     1+\frac{e^{-2\pi}} {1+\frac{e^{-4\pi}} {1+\frac{e^{-6\pi}}
      {1+\frac{e^{-8\pi}} {1+\ldots} } } }
\]</p>

<h2>A Rogers-Ramanujan Identity</h2>
```

(continues on next page)

(continued from previous page)

```
<p>\[
  1 +  \frac{q^2}{(1-q)}+\frac{q^6}{(1-q)(1-q^2)}+\cdots =
    \prod_{j=0}^{\infty}\frac{1}{(1-q^{5j+2})(1-q^{5j+3})},
     \quad\quad \text{for $|q|&lt;1$}.
\]</p>

<h2>Maxwell's Equations</h2>

<p>
\begin{align}
  \nabla \times \vec{\mathbf{B}} -\, \frac1c\, \frac{\partial\vec{\mathbf{E}}}{\
↪partial t} &amp; = \frac{4\pi}{c}\vec{\mathbf{j}} \\
  \nabla \cdot \vec{\mathbf{E}} &amp; = 4 \pi \rho \\
  \nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial\vec{\mathbf{B}}}{\
↪partial t} &amp; = \vec{\mathbf{0}} \\
  \nabla \cdot \vec{\mathbf{B}} &amp; = 0
\end{align}
</p>

<h2>In-line Mathematics</h2>

<p>While display equations look good for a page of samples, the
ability to mix math and text in a paragraph is also important.  This
expression \(\sqrt{3x-1}+(1+x)^2\) is an example of an inline equation.  As
you see, equations can be used this way as well, without unduly
disturbing the spacing between lines.</p>

<h2>References to equations</h2>

<p>Here is a reference to the Lorenz Equations (\ref{lorenz}). Clicking on the␣
↪equation number will take you back to the equation.</p>

</body>
</html>
```

10.6.2 Maklumat lanjut

Oleh kerana pelihat E-buku calibre menggunakan pustaka MathJax untuk menerap matematik, tempat terbaik untuk mengetahui berkenaan matematik dalam e-buku atau dapatkan bantuan adalah di laman sesawang MathJax.

10.7 Penciptaan Katalog AZW3 EPUB MOBI

Fitur Cipta katalog calibre membolehkan anda cipta satu katalog bagi pustaka anda dalam pelbagai format. Ia membantu fail menjelaskan pilihan katalog bila menjana katalog dalam format AZW3, EPUB dan MOBI.

- *Memilih buku untuk dikatalogkan* (halaman 213)
- *Seksyen disertakan* (halaman 214)
- *Awalan* (halaman 214)

- *Buku diasingkan* (halaman 215)
- *Genre diasingkan* (halaman 215)
- *Pilihan lain* (halaman 216)
- *Kulit buku katalog suai* (halaman 217)
- *Sumber bantuan tambahan* (halaman 217)

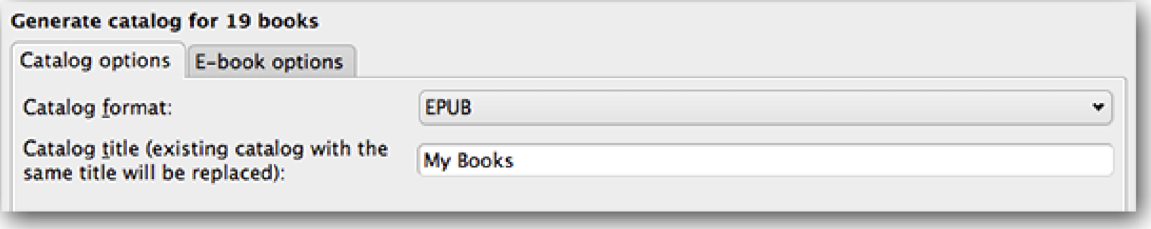
10.7.1 Memilih buku untuk dikatalogkan

Jika anda mahu *semua* pustaka anda dikatalogkan, buang mana-mana kriteria gelintar atau penapisan dalam tetingkap utama. Dengan satu buku terpilih, semua buku dalam pustaka anda akan menjadi calon untuk disertakan dalam katalog yang hendak dijana. Buku secara individu boleh diasingkan dengan pelbagai kriteria; sila rujuk seksyen *Genre diasingkan* (halaman 215) di bawah untuk maklumat lanjut.

Jika anda mahu hanya *sebahagian* dari pustaka anda dikatalogkan, anda ada dua pilihan:

- Cipta pemilihan berbilang bagi buku yang mahu dikatalogkan. Jika lebih dari sebuah buku terpilih dalam tetingkap utama calibre, hanya buku terpilih ini sahaja akan dikatalogkan.
- Guna medan Gelintar atau pelayar Tag untuk menapis buku yang dipapar. Hanya buku dipapar akan dikatalogkan.

To begin catalog generation, select the menu item *Convert books > Create a catalog of the books in your calibre library*. You may also add a *Create catalog* button to a toolbar in *Preferences > Interface > Toolbars & menus* for easier access to the Generate catalog dialog.

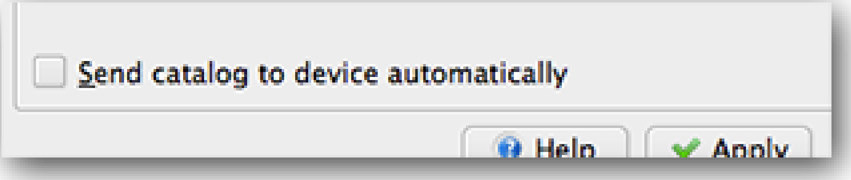


Generate catalog for 19 books

Catalog format: EPUB

Catalog title (existing catalog with the same title will be replaced): My Books

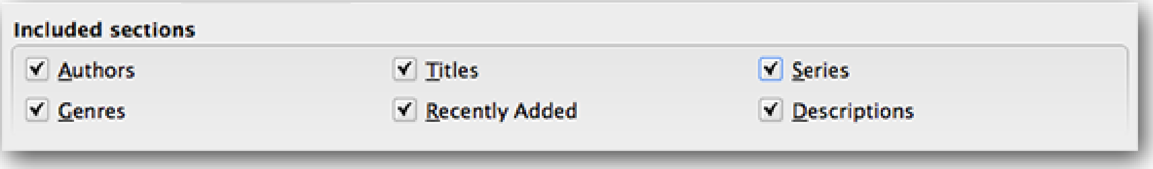
Dalam *Pilian katalog*, pilih **AZW3, EPUB atau MOBI** sebagai Format katalog. Pada medan *Tajuk katalog*, masukkan nama yang akan digunakan untuk katalog yang dijana. Jika terdapat katalog dengan nama dan format yang sama, ia akan ditulis-ganti dengan katalog terjana yang baharu.



☐ Send catalog to device automatically

Membenarkan *Hantar katalog ke peranti secara automatik* akan memuat turun katalog terjana ke peranti bersambung.

10.7.2 Seksyen disertakan



Included sections

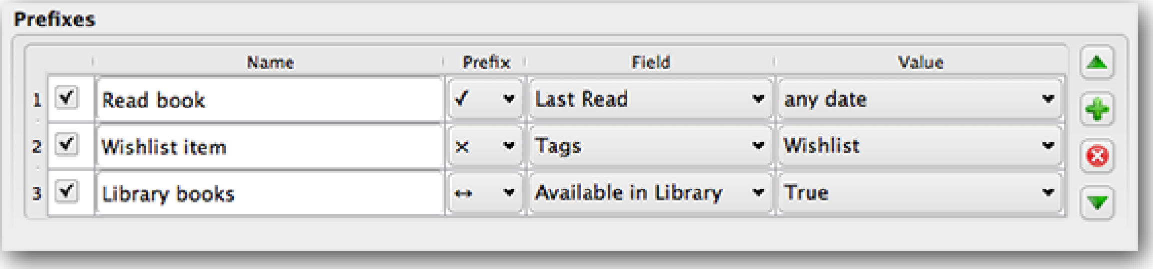
☑ Authors ☑ Titles ☑ Series

☑ Genres ☑ Recently Added ☑ Descriptions

Seksyen yang dibenarkan dengan tanda semak akan disertakan dalam katalog terjana:

- *Pengarang* - semua buku, diisih oleh pengarang, hadir dalam format senarai. Buku bukan-bersiri tersenarai sebelum buku bersiri.
- *Tajuk* - semua buku, diisih mengikut tajuk, hadir dalam format senarai.
- *Siri* - semua buku yang merupakan sebahagian dari siri, hadir dalam format siri.
- *Genre* - genre secara individu hadir dalam senarai, diisih mengikut Pengarang dan Siri.
- *Ditambah Baru-Baru Ini* - semua buku, diisih dalam tertib kronologi songsang. Senarai termasuklah buku ditambah dalam tempoh 30 terakhir, kemudian mengikut penyenaraian bulanan bagi buku yang ditambah.
- *Keterangan* - halaman keterangan terperinci bagi setiap buku, termasuklah lakaran kenit kulit buku dan ulasan. Diisih mengikut pengarang, dengan buku tanpa-siri tersenarai sebelum buku bersiri.

10.7.3 Awalan



Prefixes

		Name	Prefix	Field	Value
1	☑	Read book	✓	Last Read	any date
2	☑	Wishlist item	×	Tags	Wishlist
3	☑	Library books	↔	Available in Library	True

Peraturan awalan membolehkan anda tambah awalan ke penyenaraian buku bila kriteria tertentu dipenuhi. Sebagai contoh, anda mungkin mahu tambah buku yang telah dibaca dengan tanda semak, atau buku dalam senarai impian anda dengan tanda X.

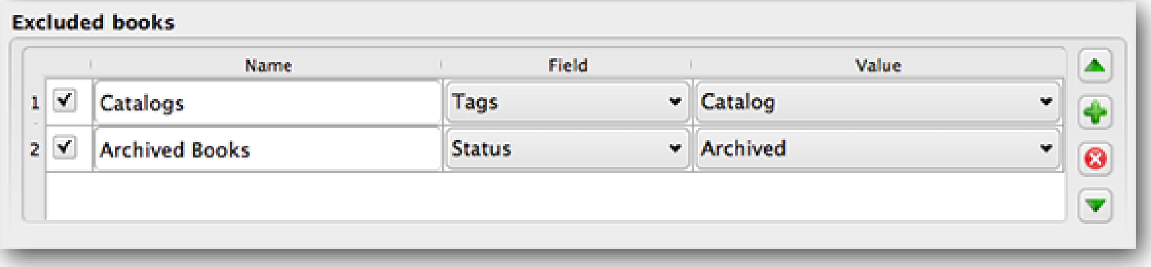
Kotak tanda pada lajur pertama membenarkan peraturan. *Nama* ialah nama peraturan yang anda sediakan. *Medan* adalah sama ada *Tag* atau ajur suai dari pustaka anda. *Nilai* pula merupakan kandungan *Medan* yang dipadankan. Bila peraturan awalan dipenuhi, buku akan ditanda dengan *Awalan* terpilih.

Tiga peraturan awalan telah dinyatakan dalam contoh di atas:

1. *Baca buku* menyatakan buku dengan mana-mana tarikh dalam lajur suai bernama *Terakhir dibaca* akan diawali dengan simbol tanda semak.
2. Item *Senarai Impian* menyatakan mana-mana buku dengan tag *Senarai Impian* akan diawali dengan simbol X.
3. Buku *pustaka' menyatakan mana-mana buku dengan nilai Benar (atau Ya) dalam lajur suai :guilabel: `Tersedia dalam Pustaka* akan diawali dengan simbol anak panah berganda dua.

Peraturan pemadanan pertama bekalkan awalan. Peraturan dilumpuhkan atau tidak lengkap diabaikan.

10.7.4 Buku diasingkan



Excluded books

		Name	Field	Value
1	☑	Catalogs	Tags	Catalog
2	☑	Archived Books	Status	Archived

Peraturan pengasingan membolehkan anda nyatakan buku yang tidak dikatalogkan.

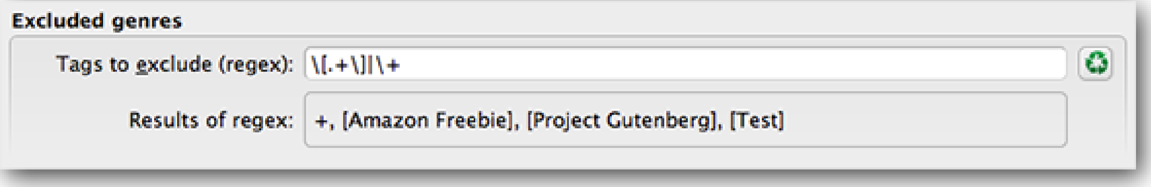
Kotak tanda pada lajur pertama membenarkan peraturan. *Nama* ialah nama peraturan yang anda sediakan. *Medan* adalah sama ada *Tag* atau lajur suai dari pustaka anda. *Nilai* pula merupakan kandungan *Medan* yang dipadankan. Bila peraturan pengecualian dipenuhi, buku akan diasingkan dari katalog yang dijana.

Dua peraturan telah dinyatakan dalam contoh di atas:

1. Item *Katalog* menyatakan mana-mana buku dengan tag *Katalog* akan diasingkan dari katalog yang dijana.
2. Peraturan buku *Diarkib' menyatakan mana-mana buku dengan nilai :guilabel: `Diarkib* dalam lajur suai *Status* akan diasing dari katalog yang dijana.

Semua peraturan dinilai bagi setiap buku. Peraturan dilumpuhkan atau tidak lengkap akan diabaikan.

10.7.5 Genre diasingkan



Excluded genres

Tags to exclude (regex): \[.+\]|\+

Results of regex: +, [Amazon Freebie], [Project Gutenberg], [Test]

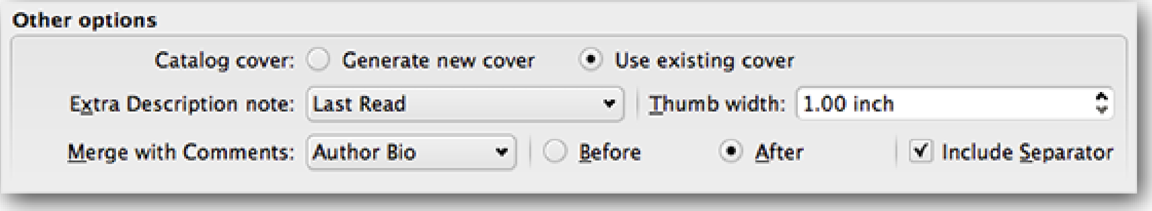
Bila katalog dijana, tag dalam pangkalan data anda digunakan sebagai genre. Sebagai contoh, anda boleh guna tag `Fiksyen` dan `Bukan Fiksyen`. Tag ini menjadi genre dalam katalog yang dijana, dengan buku tersenarai di bawah senarai genre mereka masing-masing yang berdasarkan pada tag yang telah diumpukkan. Satu buku akan disenarai dalam setiap seksyen genre yang mana ia mempunyai tag berkenaan.

Anda boleh guna beberapa tag untuk lain-lain tujuan, mungkin tanda + untuk menunjukkan buku sedang dibaca, atau tag kurungan seperti `[Amazon Freebie]` untuk menunjukkan sumber buku. Ungkapan nalar *Genre diasing* membolehkan anda nyatakan tag yang anda tidak mahu guna sebagai genre di dalam katalog yang dijana. Pola ungkapan nalar pengecualian lalai `\[.+\]\+` asingkan mana-mana tag dalam bentuk `[tag]`, begitu juga pengecualian +, tag lalai untuk buku dibaca, daripada digunakan sebagai genre dalam katalog yang dijana.

Anda juga boleh guna nama tag sebenar dalam ungkapan nalar. Sebagai contoh, `[Amazon Freebie]` atau `[Project Gutenberg]`. Jika anda mahu senaraikan tag sebenar berbilang untuk pengecualian, letak aksara paip (garis menegaker) diantara mereka: `[Amazon Freebie]|[Project Gutenberg]`.

Keputusan ungkapan nalar menunjukkan tag yang manakah akan diasing bila katalog dibina, berdasarkan pada tag dalam pangkalan data anda dan juga pola ungkapan nalar yang anda masukkan. Keputusan dikemaskini bilamana anda mengubahsuai pola ungkapan nalar.

10.7.6 Pilihan lain



Other options

Catalog cover: ○ Generate new cover ● Use existing cover

Extra Description note: Last Read Thumb width: 1.00 inch

Merge with Comments: Author Bio ○ Before ● After ☑ Include Separator

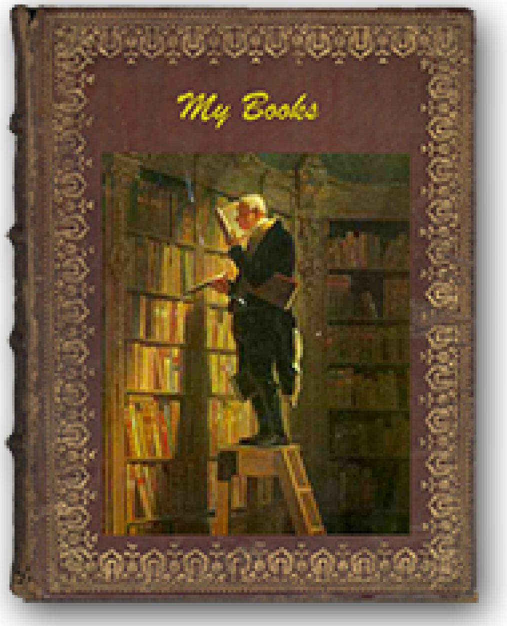
Kulit hadapan katalog menyatakan sama ada mahu jana kulit buku baharu atau guna kulit buku sedia ada. Selain boleh juga hasilkan kulit buku suai untuk katalog anda - sila rujuk Kulit buku katalog suai untuk maklumat lanjut. Jika anda telah hasilkan kulit buku suai yang mana anda mahu guna semula, pilih *Guna kulit buku sedia ada*. Jika tidak, pilih *Jana kulit buku baharu*.

Nota keterangan ekstra menyatakan kandungan lajur suai yang disisip ke dalam laman Keterangan, disebelah lakaran kenit kulit buku. Sebagai contoh, anda mungkin mahu paparkan tarikh terakhir buku dibaca menggunakan lajur suai *Terakhir Dibaca*. Untuk penggunaan lanjutan fitur nota Keterangan, sila rujuk pos ini dalam forum calibre.

Lebar lakaran kenit menyatakan keutamaan lebar untuk lakaran kenit kulit buku yang disertakan dengan laman Keterangan. Lakaran kenit dicache untuk menambahbaik prestasi. Walaubagaimanapun, untuk menguji lebar yang berlainan, cuba jana katalog dengan beberapa buah buku sehingga anda puashati dengan lebar pilihan anda, kemudian jana katalog lengkap anda. Kali pertama katalog dijana dengan lebar lakaran kenit baharu, prestasi calibre menjadi sedikit perlahan, lama-kelamaan binaan katalog yang berikutnya akan gunakan kelebihan cache lakaran kenit.

Merge with comments specifies a custom column whose content will be non-destructively merged with the comments metadata during catalog generation. For example, you might have a custom column *Author bio* that you'd like to append to the comments metadata. You can choose to insert the custom column contents *before or after* the comments section, and optionally separate the appended content with a horizontal rule separator. Eligible custom column types include `text, comments, and composite`.

10.7.7 Kulit buku katalog suai



Dengan Pemalam Jana Kulit Buku dipasang, anda boleh hasilkan kulit buku suai untuk katalog anda. Untuk memasang pemalam, pergi ke *Keutamaan* > *Lanjutan* > *Pemalam* > *Dapatkan pemalam baharu*.

10.7.8 Sumber bantuan tambahan

Untuk maklumat lanjut berkenaan fitur Katalog calibre, sila rujuk sticky forum MobileRead Creating Catalogs - Mula di sini, di mana anda boleh cari maklumat bagaimana hendak suaikan templat katalog, dan bagaimana hendak menyerahkan laporan pepijat.

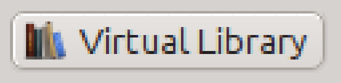
Untuk menanya soalan atau berbincang tentang fitur Katalog calibre dengan lain-lain pengguna, lawati forum MobileRead Katalog Calibre.

10.8 Pustaka maya

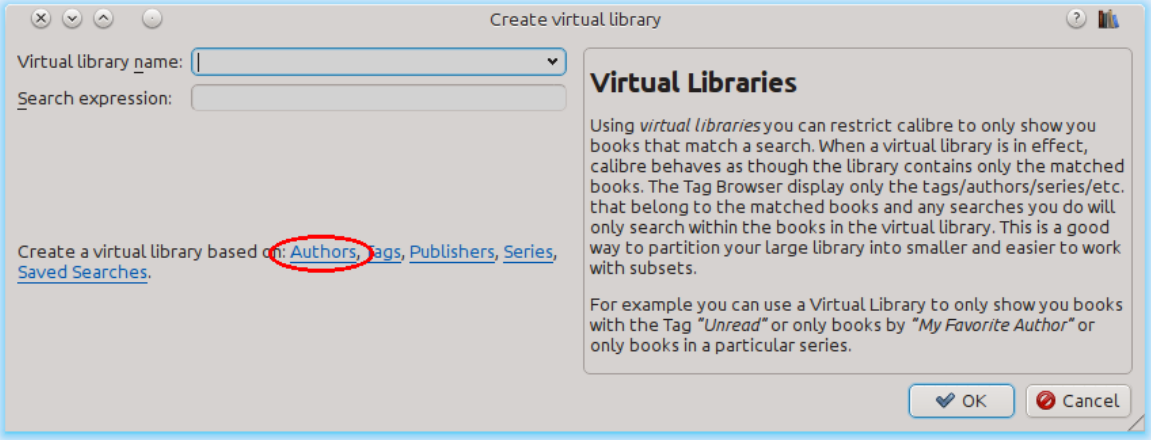
In calibre, a Virtual library is a way to tell calibre to open only a subset of a normal library. For example, you might want to only work with books by a certain author, or books having only a certain tag. Using Virtual libraries is the preferred way of partitioning your large book collection into smaller sub collections. It is superior to splitting up your library into multiple smaller libraries as, when you want to search through your entire collection, you can simply go back to the full library. There is no way to search through multiple separate libraries simultaneously in calibre.

A Virtual library is different from a simple search. A search will only restrict the list of books shown in the book list. A Virtual library does that, and in addition it also restricts the entries shown in the *Tag browser* to the left. The Tag browser will only show tags, authors, series, publishers, etc. that come from the books in the Virtual library. A Virtual library thus behaves as though the actual library contains only the restricted set of books.

10.8.1 Mencipta Pustaka maya



To use a Virtual library click the *Virtual library* button located to the left of the Search bar and select the *Create Virtual library* option. As a first example, let's create a Virtual library that shows us only the books by a particular author. Click the *Authors* link as shown in the image below and choose the author you want to use and click OK.



Virtual library name:

Search expression:

Create a virtual library based on: Authors, Tags, Publishers, Series, Saved Searches.

Virtual Libraries

Using *virtual libraries* you can restrict calibre to only show you books that match a search. When a virtual library is in effect, calibre behaves as though the library contains only the matched books. The Tag Browser display only the tags/authors/series/etc. that belong to the matched books and any searches you do will only search within the books in the virtual library. This is a good way to partition your large library into smaller and easier to work with subsets.

For example you can use a Virtual Library to only show you books with the Tag *"Unread"* or only books by *"My Favorite Author"* or only books in a particular series.

Dialog Cipta pustaka Maya telah diisikan untuk anda. Klik OK dan anda akan lihat Pustaka maya baharu telah dicipta, ditukar secara automatik kepadanya, yang hanya papar buku mengikut pengarang terpilih. Sepertimana yang calibre buat, jika pustaka anda hanya mengandungi buku mengikut pengarang terpilih.

Anda boleh tukar kembali ke pustaka penuh pada bila-bila masa dengan mengklik sekali *Pustaka maya* dan pilih masukan bernama <*None*>.

Virtual libraries are based on *searches*. You can use any search as the basis of a Virtual library. The Virtual library will contain only the books matched by that search. First, type in the search you want to use in the Search bar or build a search using the *Tag browser*. When you are happy with the returned results, click the *Virtual library* button, choose *Create library* and enter a name for the new Virtual library. The Virtual library will then be created based on the search you just typed in. Searches are very powerful, for examples of the kinds of things you can do with them, see *Antaramuka gelintar* (halaman 13).

Contoh pustaka Maya yang berguna

- **Buku ditambah ke dalam calibre pada hari terakhir:** date:>1daysago
- **Buku ditambah ke dalam calibre pada bulan terakhir:** date:>30daysago
- **Buku dengan penarafan 5 bintang:** rating:5
- **Buku dengan penarafan sekurang-kurangnya 4 bintang:** rating:>=4
- **Buku tanpa penarafan:** rating:false
- **Muat turun berkala oleh fungsi Dapatkan Berita di dalam calibre:** tags:=News and author:=calibre
- **Buku tanpa tag:** tags:false
- **Buku tanpa kulit hadapan:** cover:false

10.8.2 Mengendalikan Pustaka maya

You can edit a previously created Virtual library or remove it, by clicking the *Virtual library* and choosing the appropriate action.

You can tell calibre that you always want to apply a particular Virtual library when the current library is opened, by going to *Preferences→Interface→Behavior*.

You can quickly use the current search as a temporary Virtual library by clicking the *Virtual library* button and choosing the **current search* entry.

You can display all available Virtual libraries as tabs above the book list. This is particularly handy if you like switching between Virtual libraries very often. Click the *Virtual library* button and select *Show Virtual libraries as tabs*. You can re-arrange the tabs by drag and drop and close ones you do not want to see. Closed tabs can be restored by right-clicking on the tab bar.

10.8.3 Menggunakan Pustaka maya dalam penggelintaran

You can search for books that are in a Virtual library using the `vl:` prefix. For example, `vl:Read` will find all the books in the *Read* Virtual library. The search `vl:Read and vl:"Science Fiction"` will find all the books that are in both the *Read* and *Science Fiction* Virtual libraries.

The value following `vl:` must be the name of a Virtual library. If the Virtual library name contains spaces then surround it with quotes.

One use for a Virtual library search is in the content server. In *Preferences→Sharing over the net→Require username and password* you can limit the calibre libraries visible to a user. For each visible library you can specify a search expression to further limit which books are seen. Use `vl:"Virtual library name"` to limit the books to those in a Virtual library.

10.8.4 Menggunaan sekatan tambahan

You can further restrict the books shown in a Virtual library by using *Additional restrictions*. An additional restriction is saved search you previously created that can be applied to the current Virtual library to further restrict the books shown in a Virtual library. For example, say you have a Virtual library for books tagged as *Historical Fiction* and a saved search that shows you unread books, you can click the *Virtual Library* button and choose the *Additional restriction* option to show only unread Historical Fiction books. To learn about saved searches, see *Menyimpan gelintar* (halaman 16).

CHAPTER 11

The calibre:// URL scheme

calibre registers itself as the handler program for calibre:// URLs. So you can use these to perform actions like opening books, searching for books, etc from other programs/documents or via the command line. For example, running the following at the command line:

```
calibre calibre://switch-library/Some_Library
```

Will open calibre with the library named `Some Library`. Library names are the folder name of the library folder with spaces replaced by underscores. The special value _ means the current library. The various types of URLs are documented below.

You can even place these links inside HTML files or Word documents or similar and the operating system will automatically run calibre to perform the specified action.

- *Switch to a specific library* (halaman 222)
- *Show a specific book in calibre* (halaman 222)
- *Open a specific book in the E-book viewer at a specific position* (halaman 222)
- *Searching for books* (halaman 222)
- *Hex encoding of URL parameters* (halaman 223)

11.1 Switch to a specific library

The URL syntax is:

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

Library names are the folder name of the library with spaces replaced by underscores. The special value _ means the current library. You can also use *hex encoding* (halaman 223) for the library names, useful if the library names have special characters that would otherwise require URL encoding. Hex encoded library names look like:

```
_hex_-AD23F4BC
```

Where the part after the `_hex_-` prefix is the library name encoded as UTF-8 and every byte represented by two hexadecimal characters.

11.2 Show a specific book in calibre

The URL syntax is:

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

This will show the book with `book_id` (a number) in calibre. The ids for books can be seen in the calibre interface by hovering over the *Click to open* link in the book details panel, it is the number in brackets at the end of the path to the book folder.

You can copy a link to the current book displayed in calibre by right clicking the *Book details panel* and choosing *Copy link to book*.

11.3 Open a specific book in the E-book viewer at a specific position

The URL syntax is:

```
calibre://view-book/Library_Name/book_id/book_format?open_at=location
```

Here, `book_format` is the format of the book, for example, `EPUB` or `MOBI` and the `location` is an optional location inside the book. The easiest way to get these links is to open a book in the viewer, then in the viewer controls select *Go to→Location* and there such a link will be given that you can copy/paste elsewhere.

11.4 Searching for books

The URL syntax is:

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

Here query is any valid *search expression* (halaman 13). If the search expression is complicated, *encode it as a hex string* (halaman 223) and use `eq` instead. Leaving out the query will cause the current search to be cleared.

By default, if a Virtual library is selected, calibre will clear it before doing the search to ensure all books are found. If you want to preserve the Virtual library, use:

```
calibre://search/Library_Name?q=query&virtual_library=_
```

If you want to switch to a particular Virtual library, use:

```
calibre://search/Library_Name?virtual_library=Library%20Name
or
calibre://search/Library_Name?encoded_virtual_library=hex_encoded_virtual_library_name
```

replacing spaces in the Virtual library name by `%20`.

If you perform a search in calibre and want to generate a link for it you can do so by right clicking the search bar and choosing *Copy search as URL*.

11.5 Hex encoding of URL parameters

Hex encoding of URL parameters is done by first encoding the parameter as UTF-8 bytes, and then replacing each byte by two hexadecimal characters representing the byte. For example, the string `abc` is the bytes `0x61 0x62 and 0x63` in UTF-8, so the encoded version is the string: `616263`.

CHAPTER 12

Penyesuaian calibre.

calibre mempunyai reka bentuk yang sangat bermodular. Pelbagai bahagiannya boleh disuaikan. Anda boleh ketahui lebih lanjut bagaimana hendak mencipta *resepi* untuk tambah sumber baharu kandungan atas-talian ke dalam calibre dalam Seksyen *Menambah laman sesawang berita kegemaran anda* (halaman 27). Di sini, anda akan ketahui, bagaimana hendak guna pembolehubah persekitaran dan *tweak* untuk suaikan kelakuan calibre's, kemudian bagaimana hendak nyatakan sumber statik anda sendiri seperti ikon dan templat untuk membatalkan lalai dan akhir sekali bagaimana hendak guna *pemalam* untuk tambah lagi kefungsian calibre.

- *Pembolehubah persekitaran* (halaman 253)
- *Pulas* (halaman 254)
- *Membatalkan ikon, templat, dan lain-lain* (halaman 264)
- *Mencipta tema ikon anda sendiri untuk calibre* (halaman 264)
- *Menyuaikan calibre dengan pemalam* (halaman 265)

12.1 API documentation for plugins

Takrif pelbagai kelas asas abstrak yang boleh disub-kelaskan untuk menghasilkan pemalam yang hebat. Kelas-kelas yang berguna adalah:

- *Plugin* (halaman 226)
- *FileTypePlugin* (halaman 228)
- *Metadata plugins* (halaman 229)
- *Catalog plugins* (halaman 230)
- *Metadata download plugins* (halaman 230)

- *Conversion plugins* (halaman 234)
- *Device drivers* (halaman 236)
- *User Interface Actions* (halaman 248)
- *Preferences plugins* (halaman 251)

12.1.1 Plugin

class calibre.customize.**Plugin**(*plugin_path*)

A calibre plugin. Useful members include:

- **self.plugin_path: Stores path to the ZIP file that contains** this plugin or None if it is a builtin plugin
- **self.site_customization: Stores a customization string entered** by the user.

Methods that should be overridden in sub classes:

- *initialize()* (halaman 226)
- *customization_help()* (halaman 227)

Useful methods:

- *temporary_file()* (halaman 227)
- __enter__()
- *load_resources()* (halaman 227)

supported_platforms = []

List of platforms this plugin works on. For example: ['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 = 'Langsung tidak melakukan apa-apa'

A short string describing what this plugin does

author = 'Tidak Diketahui'

The author of this plugin

priority = 1

When more than one plugin exists for a filetype, the plugins are run in order of decreasing priority. Plugins with higher priority will be run first. The highest possible priority is sys.maxsize. Default priority is 1.

minimum_calibre_version = (0, 4, 118)

The earliest version of calibre this plugin requires

can_be_disabled = True

If False, the user will not be able to disable this plugin. Use with care.

type = 'Asas'

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()* (halaman 227) in your plugin to use a custom configuration dialog, rather then relying on the simple string based default customization.

This method, if implemented, must return a QWidget. The widget can have an optional method validate() that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns True.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

save_settings(*config_widget*)

Save the settings specified by the user with config_widget.

Parameters **config_widget** -- The widget returned by *config_widget()* (halaman 227).

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)
```

Parameters **names** -- List of paths to resources in the ZIP file using / as separator

Returns 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.

Parameters **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.

Parameters **`suffix`** -- The suffix that the temporary file will have.

`cli_main`(*args*)

This method is the main entry point for your plugins command line interface. It is called when the user does: calibre-debug -r "Plugin Name". Any arguments passed are present in the args variable.

12.1.2 FileTypePlugin

class `calibre.customize.`**`FileTypePlugin`**(*plugin_path*)

Bases: *`calibre.customize.Plugin`* (halaman 226)

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_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.

`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.

Parameters **`path_to_ebook`** -- Absolute path to the e-book.

Returns 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()`* (halaman 228) 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.

Parameters

- **`book_id`** -- Database id of the added book.
- **`book_format`** -- The file type of the book that was added.
- **`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()* (halaman 228), 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.

Parameters

- **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*)

Bases: *calibre.customize.Plugin* (halaman 226)

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'])`

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.

Parameters type -- The type of file. Guaranteed to be one of the entries in *file_types* (halaman 229).

Returns A `calibre.ebooks.metadata.book.Metadata` object

class calibre.customize.**MetadataWriterPlugin**(**args*, ***kwargs*)

Bases: *calibre.customize.Plugin* (halaman 226)

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'])`

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.

Parameters

- **type** -- The type of file. Guaranteed to be one of the entries in *file_types* (halaman 229).
- **mi** -- A `calibre.ebooks.metadata.book.Metadata` object

12.1.4 Catalog plugins

class calibre.customize.**CatalogPlugin**(*plugin_path*)

Bases: *calibre.customize.Plugin* (halaman 226)

A plugin that implements a catalog generator.

file_types = {}

Output file type for which this plugin should be run. For example: 'epub' or 'xml'

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.

Parameters

- **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*)

Bases: *calibre.customize.Plugin* (halaman 226)

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.

Parameters **gui** -- If True return HTML help, otherwise return plain text help.

config_widget()
Implement this method and *save_settings()* (halaman 231) in your plugin to use a custom configuration dialog, rather then relying on the simple string based default customization.

This method, if implemented, must return a QWidget. The widget can have an optional method validate() that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns True.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

save_settings(*config_widget*)
Save the settings specified by the user with config_widget.

Parameters **config_widget** -- The widget returned by *config_widget()* (halaman 231).

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 browseable 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()` (halaman 232).

get_cached_cover_url(*identifiers*)
Return cached cover URL for the book identified by the identifiers dict 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()` (halaman 232).

For details on the default algorithm see `InternalMetadataCompareKeyGen` (halaman 233). 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.

Parameters

- **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.

Returns 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()` (halaman 232). Put (self, cover_data) into result_queue.

This method should use cached cover URLs for efficiency whenever possible. When cached data is not present, most plugins simply call identify and use its results.

If the parameter get_best_cover is True and this plugin can get multiple covers, it should only get the "best" one.

class calibre.ebooks.metadata.sources.base.**InternalMetadataCompareKeyGen**(*mi*, *source_plugin*, *title*, *authors*, *identifiers*)

Generate a sort key for comparison of the relevance of Metadata objects, given a search query. This is used only to compare results from the same metadata source, not across different sources.

The sort key ensures that an ascending order sort is a sort by order of decreasing relevance.

The algorithm is:

- Prefer results that have at least one identifier the same as for the query
- Prefer results with a cached cover URL
- Prefer results with all available fields filled in
- Prefer results with the same language as the current user interface language
- Prefer results that are an exact title match to the query
- Prefer results with longer comments (greater than 10% longer)
- **Use the relevance of the result as reported by the metadata source's search** engine

12.1.6 Conversion plugins

class calibre.customize.conversion.**InputFormatPlugin**(**args*)

Bases: `calibre.customize.Plugin` (halaman 226)

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()` (halaman 234).

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` (halaman 234) instead. Every option must be an instance of `OptionRecommendation`.

options = {}

Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = {}

A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

get_images()

Return a list of absolute paths to the images, if this input plugin represents an image collection. The list of images is in the same order as the spine and the TOC.

convert(*stream*, *options*, *file_ext*, *log*, *accelerators*)

This method must be implemented in sub-classes. It must return the path to the created OPF file or an `OEBBook` instance. All output should be contained in the current directory. If this plugin creates files outside the current directory they must be deleted/marked for deletion before this method returns.

Parameters

- **stream** -- A file like object that contains the input file.
- **options** -- Options to customize the conversion process. Guaranteed to have attributes corresponding to all the options declared by this plugin. In addition, it will have a verbose attribute that takes integral values from zero upwards. Higher numbers mean be more verbose. Another useful attribute is `input_profile` that is an instance of `calibre.customize.profiles.InputProfile`.
- **file_ext** -- The extension (without the .) of the input file. It is guaranteed to be one of the *file_types* supported by this plugin.
- **log** -- A `calibre.utils.logging.Log` object. All output should use this object.

- **accelarators** -- A dictionary of various information that the input plugin can get easily that would speed up the subsequent stages of the conversion.

postprocess_book (*oeb*, *opts*, *log*)
Called to allow the input plugin to perform postprocessing after the book has been parsed.

specialize (*oeb*, *opts*, *log*, *output_fmt*)
Called to allow the input plugin to specialize the parsed book for a particular output format. Called after postprocess_book and before any transforms are performed on the parsed book.

gui_configuration_widget (*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)
Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin input plugins for examples.

class `calibre.customize.conversion.`**OutputFormatPlugin** (**args*)
Bases: `calibre.customize.Plugin` (halaman 226)

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()` (halaman 235).

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` (halaman 235) 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.

Parameters

- **output** -- Either a file like object or a string. If it is a string it is the path to a directory that may or may not exist. The output plugin should write its output into that directory. 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.

Parameters

- **item** -- The item (HTML file) being processed
- **stylizer** -- A Stylizer object containing the flattened styles for item. You can get the style for any element by stylizer.style(element).

gui_configuration_widget (*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)
Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin output plugins for examples.

12.1.7 Device drivers

The base class for all device drivers is *DevicePlugin* (halaman 236). However, if your device exposes itself as a USBMS drive to the operating system, you should use the USBMS class instead as it implements all the logic needed to support these kinds of devices.

class calibre.devices.interface.**DevicePlugin** (*plugin_path*)
Bases: *calibre.customize.Plugin* (halaman 226)

Defines the interface that should be implemented by backends that communicate with an e-book reader.

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 = '/home/kovid/work/calibre/resources/images/reader.png'
Icon for this device

UserAnnotation
alias of `Annotation`

OPEN_FEEDBACK_MESSAGE = None
GUI displays this as a message if not None. 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()`* (halaman 239) 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()`* (halaman 241) and *`ignore_connected_device()`* (halaman 241) and *`get_user_blacklisted_devices()`* (halaman 242) and *`set_user_blacklisted_devices()`* (halaman 242)

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.

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.

Parameters **devices_on_system** -- List of devices currently connected

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.

Parameters

- **`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*)

Parameters

- **`key`** -- The key to unlock the device
- **`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()`* (halaman 238).

Note that for devices based on USBMS this method by default delegates to *`can_handle()`* (halaman 238). So you only need to override *`can_handle()`* (halaman 238) in your subclass of USBMS.

Parameters **`usbdevice`** -- A usbdevice as returned by `calibre.devices.winusb.scan_usb_devices()`

`can_handle`(*device_info*, *debug=False*)

Unix version of *`can_handle_windows()`* (halaman 238).

Parameters **`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()`* (halaman 246) 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.

Parameters

- **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.

Parameters **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}.

Returns (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. Main memory
2. Memory Card A
3. Memory Card B

Returns 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. Main memory
2. Card A
3. Card B

Returns 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.

Parameters **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.

Returns 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".

Parameters

- **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).

Returns A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (halaman 240).

classmethod add_books_to_metadata (*locations*, *metadata*, *booklists*)
Add locations to the booklists. This function must not communicate with the device.

Parameters

- **locations** -- Result of a call to L{upload_books}
- **metadata** -- List of `Metadata` objects, same as for *upload_books()* (halaman 240).
- **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.

Parameters

- **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.

Parameters **`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.

Parameters **`outfile`** -- file object like `sys.stdout` or the result of an *`open()`* (halaman 238) 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()`* (halaman 241) 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).

Returns None or a single plugboard instance.

`set_driveinfo_name`(*location_code*, *name*)

Set the device name in the driveinfo file to 'name'. This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the get_device_information() method.

`prepare_addable_books`(*paths*)

Given a list of paths, returns another list of paths. These paths point to addable versions of the books.

If there is an error preparing a book, then instead of a path, the position in the returned list for that book should be a three tuple: (original_path, the exception instance, traceback)

`startup`()

Called when calibre is starting the device. Do any initialization required. Note that multiple instances of the class can be instantiated, and thus __init__ can be called multiple times, but only one instance will have this method called. This method is called on the device thread, not the GUI thread.

`shutdown`()

Called when calibre is shutting down, either for good or in preparation to restart. Do any cleanup required. This method is called on the device thread, not the GUI thread.

`get_device_uid`()

Must return a unique id for the currently connected device (this is called immediately after a successful call to open()). You must implement this method if you set ASK_TO_ALLOW_CONNECT = True

`ignore_connected_device`(*uid*)

Should ignore the device identified by uid (the result of a call to get_device_uid()) in the future. You must implement this method if you set ASK_TO_ALLOW_CONNECT = True. Note that this function is called immediately after open(), so if open() caches some state, the driver should reset that state.

get_user_blacklisted_devices()
Return map of device uid to friendly name for all devices that the user has asked to be ignored.

set_user_blacklisted_devices(*devices*)
Set the list of device uids that should be ignored by this driver.

specialize_global_preferences(*device_prefs*)
Implement this method if your device wants to override a particular preference. You must ensure that all call sites that want a preference that can be overridden use device_prefs['something'] instead of prefs['something']. Your method should call device_prefs.set_overrides(pref=val, pref=val, ...). Currently used for: metadata management (prefs['manage_device_metadata'])

set_library_info(*library_name*, *library_uuid*, *field_metadata*)
Implement this method if you want information about the current calibre library. This method is called at startup and when the calibre library changes while connected.

is_dynamically_controllable()
Called by the device manager when starting plugins. If this method returns a string, then a) it supports the device manager's dynamic control interface, and b) that name is to be used when talking to the plugin.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

start_plugin()
This method is called to start the plugin. The plugin should begin to accept device connections however it does that. If the plugin is already accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

stop_plugin()
This method is called to stop the plugin. The plugin should no longer accept connections, and should cleanup behind itself. It is likely that this method should call shutdown. If the plugin is already not accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

get_option(*opt_string*, *default=None*)
Return the value of the option indicated by opt_string. This method can be called when the plugin is not started. Return None if the option does not exist.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

set_option(*opt_string*, *opt_value*)
Set the value of the option indicated by opt_string. This method can be called when the plugin is not started.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

is_running()
Return True if the plugin is started, otherwise false

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

synchronize_with_db(*db*, *book_id*, *book_metadata*, *first_call*)
Called during book matching when a book on the device is matched with a book in calibre's db. The method is responsible for syncronizing data from the device to calibre's db (if needed).

The method must return a two-value tuple. The first value is a set of calibre book ids changed if calibre's database was changed or None if the database was not changed. If the first value is an empty set then the metadata for the book on the device is updated with calibre's metadata and given back to the device, but no GUI refresh of that book is done. This is useful when the calibre data is correct but must be sent to the device.

The second value is itself a 2-value tuple. The first value in the tuple specifies whether a book format should be sent to the device. The intent is to permit verifying that the book on the device is the same as the book in calibre. This value must be None if no book is to be sent, otherwise return the base file name on the device (a string like foobar.epub). Be sure to include the extension in the name. The device subsystem will construct a send_books job for all books with not- None returned values. Note: other than to later retrieve the extension, the name is ignored in cases where the device uses a template to generate the file name, which most do. The second value in the returned tuple indicated whether the format is future-dated. Return True if it is, otherwise return False. calibre will display a dialog to the user listing all future dated books.

Extremely important: this method is called on the GUI thread. It must be threadsafe with respect to the device manager's thread.

book_id: the calibre id for the book in the database. book_metadata: the Metadata object for the book coming from the device. first_call: True if this is the first call during a sync, False otherwise

class `calibre.devices.interface.`**`BookList`**(*oncard*, *prefix*, *settings*)

Bases: `list`

A list of books. Each Book object must have the fields

1. title
2. authors
3. size (file size of the book)
4. datetime (a UTC time tuple)
5. path (path on the device to the book)
6. thumbnail (can be None) thumbnail is either a str/bytes object with the image data or it should have an attribute image_path that stores an absolute (platform native) path to the image
7. tags (a list of strings, can be empty).

`supports_collections`()

Return True if the device supports collections for this book list.

`add_book`(*book*, *replace_metadata*)

Add the book to the booklist. Intent is to maintain any device-internal metadata. Return True if booklists must be sync'ed

`remove_book`(*book*)

Remove a book from the booklist. Correct any device metadata at the same time

`get_collections`(*collection_attributes*)

Return a dictionary of collections created from collection_attributes. Each entry in the dictionary is of the form collection name:[list of books]

The list of books is sorted by book title, except for collections created from series, in which case series_index is used.

Parameters **`collection_attributes`** -- A list of attributes of the Book object

USB Mass Storage based devices

The base class for such devices is *calibre.devices.usbms.driver.USBMS* (halaman 246). This class in turn inherits some of its functionality from its bases, documented below. A typical basic USBMS based driver looks like this:

```
from calibre.devices.usbms.driver import USBMS

class PDNOVEL(USBMS):
    name = 'Pandigital Novel device interface'
    gui_name = 'PD Novel'
    description = _('Communicate with the Pandigital Novel')
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'linux', 'osx']
    FORMATS = ['epub', 'pdf']

    VENDOR_ID   = [0x18d1]
    PRODUCT_ID  = [0xb004]
    BCD         = [0x224]

    THUMBNAIL_HEIGHT = 144

    EBOOK_DIR_MAIN = 'eBooks'
    SUPPORTS_SUB_DIRS = False

    def upload_cover(self, path, filename, metadata):
        coverdata = getattr(metadata, 'thumbnail', None)
        if coverdata and coverdata[2]:
            with open('%s.jpg' % os.path.join(path, filename), 'wb') as coverfile:
                coverfile.write(coverdata[2])
```

class calibre.devices.usbms.device.**Device**(*plugin_path*)

Bases: calibre.devices.usbms.deviceconfig.DeviceConfig, *calibre.devices.interface.DevicePlugin* (halaman 236)

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.

WINDOWS_MAIN_MEM = None

String identifying the main memory of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_A_MEM = None

String identifying the first card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_B_MEM = None

String identifying the second card of the device in the Windows PnP id strings This can be None, string, list of strings or compiled regex

OSX_MAIN_MEM_VOL_PAT = None

Used by the new driver detection to disambiguate main memory from storage cards. Should be a regular expression that matches the main memory mount point assigned by macOS

MAX_PATH_LEN = 250

The maximum length of paths created on the device

NEWS_IN_FOLDER = True

Put news in its own folder

reset (*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Parameters

- **key** -- The key to unlock the device
- **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.

Parameters **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. Main memory
2. Memory Card A
3. Memory Card B

Returns 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. Main memory
2. Card A
3. Card B

Returns A 3 element list with free space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return -1.

windows_sort_drives (*drives*)

Called to disambiguate main memory and storage card for devices that do not distinguish between them on the basis of *WINDOWS_CARD_NAME*. For e.g.: The EB600

can_handle_windows (*usbdevice*, *debug=False*)

Optional method to perform further checks on a device to see if this driver is capable of handling it. If it is not it should return False. This method is only called after the vendor, product ids and the bcd have matched, so it can do some relatively time intensive checks. The default implementation returns True. This method is called only on Windows. See also `can_handle()`.

Note that for devices based on USBMS this method by default delegates to `can_handle()`. So you only need to override `can_handle()` in your subclass of USBMS.

Parameters **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()`* (halaman 246) 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.

Parameters

- **connected_device** -- The device that we are trying to open. It is a tuple of (vendor id, product id, bcd, manufacturer name, product name, device serial number). However, some devices have no serial number and on Windows only the first three fields are present, the rest are None.
- **library_uuid** -- The UUID of the current calibre library. Can be None if there is no library (for example when used from the command line).

eject()

Un-mount / eject the device from the OS. This does not check if there are pending GUI jobs that need to communicate with the device.

NOTE: That this method may not be called on the same thread as the rest of the device methods.

post_yank_cleanup()

Called if the user yanks the device without ejecting it first.

sanitize_callback(*path*)

Callback to allow individual device drivers to override the path sanitization used by `create_upload_path()`.

filename_callback(*default*, *mi*)

Callback to allow drivers to change the default file name set by `create_upload_path()`.

sanitize_path_components(*components*)

Perform any device specific sanitization on the path components for files to be uploaded to the device

get_annotations(*path_map*)

Resolve path_map to annotation_map of files found on the device

add_annotation_to_library(*db*, *db_id*, *annotation*)

Add an annotation to the calibre library

class calibre.devices.usbms.cli.**CLI**

class calibre.devices.usbms.driver.**USBMS**(*plugin_path*)

Bases: *`calibre.devices.usbms.cli.CLI`* (halaman 246), *`calibre.devices.usbms.device.Device`* (halaman 244)

The base class for all USBMS devices. Implements the logic for sending/getting/updating metadata/caching metadata/etc.

booklist_class

alias of `calibre.devices.usbms.books.BookList`

book_class

alias of `calibre.devices.usbms.books.Book`

get_device_information(*end_session=True*)

Ask device for device information. See L{DeviceInfoQuery}.

Returns (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.

Parameters **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.

Returns 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".

Parameters

- **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).

Returns A list of 3-element tuples. The list is meant to be passed to `add_books_to_metadata()` (halaman 247).

upload_cover(*path*, *filename*, *metadata*, *filepath*)

Upload book cover to the device. Default implementation does nothing.

Parameters

- **path** -- The full path to the directory 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.

Parameters

- **locations** -- Result of a call to L{upload_books}

- **metadata** -- List of Metadata objects, same as for *upload_books()* (halaman 247).
- **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.

Parameters

- **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.

Parameters **booklists** -- A tuple containing the result of calls to (books(oncard=None)(), books(oncard='carda')(), :meth`books(oncard='cardb')`).

classmethod normalize_path(*path*)
Return path with platform native path separators

12.1.8 User Interface Actions

If you are adding your own plugin in a ZIP file, you should subclass both InterfaceActionBase and InterfaceAction. The load_actual_plugin() method of your InterfaceActionBase subclass must return an instantiated object of your InterfaceBase subclass.

class calibre.gui2.actions.**InterfaceAction**(*parent*, *site_customization*)
Bases: PyQt5.QtCore.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* (halaman 248) objects have the same name, the one with higher priority takes precedence.

Sub-classes should implement the *genesis()* (halaman 250), *library_changed()* (halaman 250), *location_selected()* (halaman 250) *shutting_down()* (halaman 251) and *initialization_complete()* (halaman 251) 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` (halaman 249) 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 shortcut must be a string, None or tuple of shortcuts. If None, a keyboard shortcut corresponding to the action is not registered. If you pass an empty tuple, then the shortcut is registered with no default key binding.

`action_add_menu = False`
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()` (halaman 249), :meth`:accept_drag_move_event`, `drop_event()` (halaman 249) 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.

Parameters

- **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 I() builtin, so you do not need to pass the full path to the images directory.
- **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)
```

Parameters **names** -- List of paths to resources in the ZIP file using / as separator

Returns 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* (halaman 249) 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_changed(*db*)

Called whenever the current library is changed.

Parameters **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()` (halaman 251).

initialization_complete()
Called once per action when the initialization of the main GUI is completed.

shutting_down()
Called once per plugin when the main GUI is in the process of shutting down. Release any used resources, but try not to block the shutdown for long periods of time.

class calibre.customize.**InterfaceActionBase**(**args*, ***kwargs*)
Bases: `calibre.customize.Plugin` (halaman 226)

load_actual_plugin(*gui*)
This method must return the actual interface action plugin object.

12.1.9 Preferences plugins

class calibre.customize.**PreferencesPlugin**(*plugin_path*)
Bases: `calibre.customize.Plugin` (halaman 226)

A plugin representing a widget displayed in the Preferences dialog.

This plugin has only one important method `create_widget()` (halaman 251). The various fields of the plugin control how it is categorized in the UI.

config_widget = None
Import path to module that contains a class named ConfigWidget which implements the ConfigWidgetInterface. Used by `create_widget()` (halaman 251).

category_order = 100
Where in the list of categories the `category` (halaman 251) of this plugin should be.

name_order = 100
Where in the list of names in a category, the `gui_name` (halaman 251) 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` (halaman 251).

The default implementation uses `config_widget` (halaman 251) 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` (halaman 252) 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 = 'Pilih tetapan ke nilai lalai. Anda perlu klik Laksana untuk s
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()`* (halaman 252) returns True.

genesis(*gui*)
Called once before the widget is displayed, should perform any necessary setup.

Parameters gui -- The main calibre graphical user interface

initialize()
Should set all config values to their initial values (the values stored in the config files).

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.

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()`* (halaman 252) 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`* (halaman 251) methods and call the base class methods inside the overrides.

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.

Parameters

- **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).

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 Pembolehubah persekitaran

- `CALIBRE_CONFIG_DIRECTORY` - tetapkan direktori yang mana fail konfiguration disimpan/dibaca.
- `CALIBRE_TEMP_DIR` - tetapkan direktori sementara yang digunakan oleh calibre
- `CALIBRE_CACHE_DIRECTORY` - tetapkan direktori yang mana calibre gunakan untuk cache data tekal diantara sesi
- `CALIBRE_OVERRIDE_DATABASE_PATH` - membolehkan anda nyatakan laluan penuh ke metadata.db. Dengan menggunakan pembolehubah ini anda boleh pastikan metadata.db berada di lokasi lain selain dari folder pustaka. Berguna jika folder pustaka anda berada pada pemacu rangkaian yang tidak menyokong penguncian fail.
- `CALIBRE_DEVELOP_FROM` - Digunakan untuk jalankan persekitaran pembangunan calibre. Rujuk *Menetapkan persekitaran pembangunan calibre* (halaman 305).
- `CALIBRE_OVERRIDE_LANG` - Digunakan untuk paksa bahasa yang digunakan oleh antaramuka (kod bahasa ISO 639)
- `CALIBRE_TEST_TRANSLATION` - Digunakan untuk menguji fail terjemahan .po (laluan ke fail .po)
- `CALIBRE_NO_NATIVE_FILEDIALOGS` - Menyebabkan calibre tidak gunakan dialog fail tabii ketika memilih fail/direktori.
- `CALIBRE_NO_NATIVE_MENUBAR` - Menyebabkan calibre tidak mencipta menu tabii (sejagat) pada Ubuntu Unity dan persekitaran desktop linux yang serupa. Menu akan diletak dalam tetingkap, seperti biasa.
- `CALIBRE_USE_SYSTEM_THEME` - Secara lalai, pada Linux, calibre menggunakan gaya Qt terbina-dalamnya sendiri. Ia bertujuan untuk menghindari kerosakan dan menjadi kaku yang disebabkan oleh ketidakserasian diantara versi Qt calibre dengan versi Qt sistem. Kelemahanya pula, calibre tidak mengikuti penampilan sistem. Jika anda tetapkan pembolehubah persekitaran ini pada Linux, ia akan menyebabkan calibre menggunakan tema sistem - tetapi hati-hati jika berlakunya kerosakan dan menjadi kaku.
- `CALIBRE_SHOW_DEPRECATION_WARNINGS` - Menyebabkan calibre cetak amaran lapuk ke stdout. Berguna untuk pembangun calibre.
- `CALIBRE_NO_DEFAULT_PROGRAMS` - Halang calibre daripada mendaftar jenis fail yang boleh dikendali dengan Windows.
- `CALIBRE_USE_DARK_PALETTE` - Set it to `1` to have calibre use dark colors and `0` for normal colors (ignored on macOS). On Windows 10 in the absence of this variable, the Windows system preference for dark colors is used.
- `SYSFS_PATH` - Guna jika sysfs dilekap selain dari /sys
- `http_proxy`, `https_proxy` - Used on Linux to specify an HTTP(S) proxy

See How to set environment variables in Windows. If you are on macOS you can set environment variables by creating the ~/Library/Preferences/calibre/macos-env.txt and putting the environment variables one per line in it, for example:

```
CALIBRE_DEVELOP_FROM=$HOME/calibre-src/src
CALIBRE_NO_NATIVE_FILEDIALOGS=1
CALIBRE_CONFIG_DIRECTORY=~/.config/calibre
```

12.3 Pulas

Tweak adalah perubahan kecil yang mana anda boleh tentukan untuk mengawal pelbagai aspek kelakuan calibre. Pergi ke Keutamaan->Lanjutan->Tweak. Nilai lalai untuk tweaks seperti berikut

```
#!/usr/bin/env python
# vim:fileencoding=UTF-8:ts=4:sw=4:sta:et:sts=4:ai
# License: GPLv3 Copyright: 2010, Kovid Goyal <kovid at kovidgoyal.net>


# Contains various tweaks that affect calibre behavior. Only edit this file if
# you know what you are doing. If you delete this file, it will be recreated from
# defaults.

#: Auto increment series index
# The algorithm used to assign a book added to an existing series a series number.
# New series numbers assigned using this tweak are always integer values, except
# if a constant non-integer is specified.
# Possible values are:
#   next - First available integer larger than the largest existing number
#   first_free - First available integer larger than 0
#   next_free - First available integer larger than the smallest existing number
#   last_free - First available integer smaller than the largest existing number.␣
↪Return largest existing + 1 if no free number is found
#   const - Assign the number 1 always
#   no_change - Do not change the series index
#   a number - Assign that number always. The number is not in quotes. Note that 0.0␣
↪can be used here.
# Examples:
#   series_index_auto_increment = 'next'
#   series_index_auto_increment = 'next_free'
#   series_index_auto_increment = 16.5
#
# Set the use_series_auto_increment_tweak_when_importing tweak to True to
# use the above values when importing/adding books. If this tweak is set to
# False (the default) then the series number will be set to 1 if it is not
# explicitly set during the import. If set to True, then the
# series index will be set according to the series_index_auto_increment setting.
# Note that the use_series_auto_increment_tweak_when_importing tweak is used
# only when a value is not provided during import. If the importing regular
# expression produces a value for series_index, or if you are reading metadata
# from books and the import plugin produces a value, than 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
```

(continues on next page)

(continued from previous page)

```
# Should the completion separator be append
# to the end of the completed text to
# automatically begin a new completion operation
# for authors.
# 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 prefixes.
# The author name copy words 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 name. This means that the sort for a string like Acme
# Inc. will be Acme Inc. instead of Inc., Acme
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 = ('Corporation', 'Company', 'Co.', 'Agency', 'Council',
        'Committee', 'Inc.', 'Institute', 'Society', 'Club', 'Team')

#: 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 tags panel (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 tags panel 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
```

(continues on next page)

(continued from previous page)

```
# 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}'

#: Control order of categories in the Tag browser
# Change the following dict to change the order that categories are displayed in
# the Tag browser. Items are named using their lookup name, and will be sorted
# using the number supplied. The lookup name '*' stands for all names that
# otherwise do not appear. Two names with the same value will be sorted
# using the default order; the one used when the dict is empty.
# Example: tag_browser_category_order = {'series':1, 'tags':2, '*':3}
# resulting in the order series, tags, then everything else in default order.
tag_browser_category_order = {'*':1}


#: Specify columns to sort the booklist by on startup
# Provide a set of columns to be sorted on when calibre starts.
# The argument is None if saved sort history is to be used
# otherwise it is a list of column,order pairs. Column is the
# lookup/search name, found using the tooltip for the column
# Order is 0 for ascending, 1 for descending.
# For example, set it to [('authors',0),('title',0)] to sort by
# title within authors.
sort_columns_at_startup = None

#: Control how dates are displayed
# Format to be used for publication date and the timestamp (date).
#  A string controlling how the publication date is displayed in the GUI
#  d     the day as number without a leading zero (1 to 31)
#  dd    the day as number with a leading zero (01 to 31)
#  ddd   the abbreviated localized day name (e.g. 'Mon' to 'Sun').
#  dddd  the long localized day name (e.g. 'Monday' to 'Sunday').
#  M     the month as number without a leading zero (1-12)
#  MM    the month as number with a leading zero (01-12)
#  MMM   the abbreviated localized month name (e.g. 'Jan' to 'Dec').
#  MMMM  the long localized month name (e.g. 'January' to 'December').
#  yy    the year as two digit number (00-99)
#  yyyy  the year as four digit number
#  h     the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm) '
```

(continues on next page)

(continued from previous page)

```
#  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 return
# without changing anything is sufficient to change the sort. Or you can use
# the 'Update title sort' action in the Bulk metadata edit dialog to update
# it for many books at once.
title_series_sorting = 'library_order'

#: Control formatting of title and series when used in templates
# Control how title and series names are formatted when saving to disk/sending
# to device. The behavior depends on the field being processed. If processing
# title, then if this tweak is set to 'library_order', the title will be
# replaced with title_sort. If it is set to 'strictly_alphabetic', then the
# title will not be changed. If processing series, then if set to
# 'library_order', articles such as 'The' and 'An' will be moved to the end. If
# set to 'strictly_alphabetic', the series will be sent without change.
# For example, if the tweak is set to library_order, "The Lord of the Rings"
# will become "Lord of the Rings, The". If the tweak is set to
# strictly_alphabetic, it would remain "The Lord of the Rings". Note that the
# formatter function raw_field will return the base value for title and
# series regardless of the setting of this tweak.
save_template_title_series_sorting = 'library_order'

#: Set the list of words considered to be "articles" for sort strings
# Set the list of words that are to be considered 'articles' when computing the
# title sort strings. The articles differ by language. By default, calibre uses
# a combination of articles from English and whatever language the calibre user
# interface is set to. In addition, in some contexts where the book language is
```

(continues on next page)

(continued from previous page)

```
# 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'", 'Lt’'),
        # 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'Lt’', u'L', r'Les\s+', r'Un\s+', r'Une\
↪s+',
                  r'Des\s+', r'De\s+La\s+', r'De\s+', r"D'", u'Dt’', u'L'),
        # Italian
        'ita': ('Lo\\s+', 'Il\\s+', "L'", 'Lt’', 'La\\s+', 'Gli\\s+',
                'I\\s+', 'Le\\s+', 'Uno\\s+', 'Un\\s+', 'Una\\s+', "Un'",
                'Unt’', 'Dei\\s+', 'Degli\\s+', 'Delle\\s+', 'Del\\s+',
                'Della\\s+', 'Dello\\s+', "Dell'", 'Dellt’'),
        # 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'Nite\s+', ),
        # German
        'deu'  : (r'Der\s+', r'Die\s+', r'Das\s+', r'Den\s+', r'Ein\s+',
                  r'Eine\s+', r'Einen\s+', r'Dem\s+', r'Des\s+', r'Einem\s+',
                  r'Eines\s+'),
        # Dutch
        'nld'  : (r'De\s+', r'Het\s+', r'Een\s+', r"'n\s+", r"'s\s+", r'Ene\s+',
                  r'Ener\s+', r'Enes\s+', r'Den\s+', r'Der\s+', r'Des\s+',
                  r"'t\s+"),
        # Swedish
        'swe'  : (r'En\s+', r'Ett\s+', r'Det\s+', r'Den\s+', r'De\s+', ),
        # Turkish
        'tur'  : (r'Bir\s+', ),
        # Afrikaans
        'afr'  : (r"'n\s+", r'Die\s+', ),
        # Greek
        'ell'  : (r'O\s+', r'I\s+', r'To\s+', r'Ta\s+', r'Tus\s+', r'Tis\s+',
                  r"'Enas\s+", r"'Mia\s+", r"'Ena\s+", r"'Enan\s+", ),
        # Hungarian
        'hun'  : (r'A\s+', r'Az\s+', r'Egy\s+',),
}
default_language_for_title_sort = None
title_sort_articles=r'^(A|The|An)\s+'

#: Specify a folder calibre should connect to at startup
# Specify a folder that calibre should connect to at startup using
# connect_to_folder. This must be a full path to the folder. If the folder does
# not exist when calibre starts, it is ignored.
# Example for Windows:
```

(continues on next page)

(continued from previous page)

```
#     auto_connect_to_folder = 'C:/Users/someone/Desktop/testlib'
# Example for other operating systems:
#     auto_connect_to_folder = '/home/dropbox/My Dropbox/someone/library'
auto_connect_to_folder = ''

#: Specify renaming rules for SONY collections
# Specify renaming rules for sony collections. This tweak is only applicable if
# metadata management is set to automatic. Collections on Sonys are named
# depending upon whether the field is standard or custom. A collection derived
# from a standard field is named for the value in that field. For example, if
# the standard 'series' column contains the value 'Darkover', then the
# collection name is 'Darkover'. A collection derived from a custom field will
# have the name of the field added to the value. For example, if a custom series
# column named 'My Series' contains the name 'Darkover', then the collection
# will by default be named 'Darkover (My Series)'. For purposes of this
# documentation, 'Darkover' is called the value and 'My Series' is called the
# category. If two books have fields that generate the same collection name,
# then both books will be in that collection.
# This set of tweaks lets you specify for a standard or custom field how
# the collections are to be named. You can use it to add a description to a
# standard field, for example 'Foo (Tag)' instead of the 'Foo'. You can also use
# it to force multiple fields to end up in the same collection. For example, you
# could force the values in 'series', '#my_series_1', and '#my_series_2' to
# appear in collections named 'some_value (Series)', thereby merging all of the
# fields into one set of collections.
# There are two related tweaks. The first determines the category name to use
# for a metadata field.  The second is a template, used to determines how the
# value and category are combined to create the collection name.
# The syntax of the first tweak, sony_collection_renaming_rules, is:
# {'field_lookup_name':'category_name_to_use', 'lookup_name':'name', ...}
# The second tweak, sony_collection_name_template, is a template. It uses the
# same template language as plugboards and save templates. This tweak controls
# how the value and category are combined together to make the collection name.
# The only two fields available are {category} and {value}. The {value} field is
# never empty. The {category} field can be empty. The default is to put the
# value first, then the category enclosed in parentheses, it isn't empty:
# '{value} {category:|(|)}'
# 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:| (|)}'
```

(continues on next page)

(continued from previous page)

```
#: Specify how SONY collections are sorted
# Specify how sony collections are sorted. This tweak is only applicable if
# metadata management is set to automatic. You can indicate which metadata is to
# be used to sort on a collection-by-collection basis. The format of the tweak
# is a list of metadata fields from which collections are made, followed by the
# name of the metadata field containing the sort value.
# Example: The following indicates that collections built from pubdate and tags
# are to be sorted by the value in the custom column '#mydate', that collections
# built from 'series' are to be sorted by 'series_index', and that all other
# collections are to be sorted by title. If a collection metadata field is not
# named, then if it is a series- based collection it is sorted by series order,
# otherwise it is sorted by title order.
# [(['pubdate', 'tags'],'#mydate'), (['series'],'series_index'), (['*'], 'title')]
# Note that the bracketing and parentheses are required. The syntax is
# [ ( [list of fields], sort field ) , ( [ list of fields ] , sort field ) ]
# Default: empty (no rules), so no collection attributes are named.
sony_collection_sorting_rules = []

#: Control how tags are applied when copying books to another library
# Set this to True to ensure that tags in 'Tags to add when adding
# a book' are added when copying books to another library
add_new_book_tags_when_importing_books = False

#: Set the maximum number of sort 'levels'
# Set the maximum number of sort 'levels' that calibre will use to resort the
# library after certain operations such as searches or device insertion. Each
# sort level adds a performance penalty. If the database is large (thousands of
# books) the penalty might be noticeable. If you are not concerned about multi-
# level sorts, and if you are seeing a slowdown, reduce the value of this tweak.
maximum_resort_levels = 5

#: Choose whether dates are sorted using visible fields
# Date values contain both a date and a time. When sorted, all the fields are
# used, regardless of what is displayed. Set this tweak to True to use only
# the fields that are being displayed.
sort_dates_using_visible_fields = False

#: Fuzz value for trimming covers
# The value used for the fuzz distance when trimming a cover.
# Colors within this distance are considered equal.
# The distance is in absolute intensity units.
cover_trim_fuzz_value = 10

#: Control behavior of the book list
# You can control the behavior of double clicks and pressing enter on the books list.
# Choices: open_viewer, do_nothing,
# edit_cell, edit_metadata. Selecting anything other than open_viewer 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 horizontal per column or
# per pixel. Default is per column.
doubleclick_on_library_view = 'open_viewer'
enter_key_behavior = 'do_nothing'
horizontal_scrolling_per_column = True
```

(continues on next page)

(continued from previous page)

```
#: 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 =  ''

#: Number of columns for custom metadata in the edit metadata dialog
# Set whether to use one or two columns for custom metadata when editing
# metadata  one book at a time. If True, then the fields are laid out using two
# columns. If False, one column is used.
metadata_single_use_2_cols_for_custom_fields = True

#: Order of custom column(s) in edit metadata
# Controls the order that custom columns are listed in edit metadata single
# and bulk. The columns listed in the tweak are displayed first and in the
# order provided. Any columns not listed are displayed after the listed ones,
# in alphabetical order. Do note that this tweak does not change the size of
# the edit widgets. Putting comments widgets in this list may result in some
# odd widget spacing when using two-column mode.
# Enter a comma-separated list of custom field lookup names, as in
# metadata_edit_custom_column_order = ['#genre', '#mytags', '#etc']
metadata_edit_custom_column_order = []

#: Edit metadata custom column label width and elision point
# Set the width of custom column labels shown in the edit metadata dialogs.
# If metadata_edit_elide_labels is True then labels wider than the width
# will be elided, otherwise they will be word wrapped. The maximum width is
# computed by multiplying the average width of characters in the font by the
# appropriate number.
# Set the elision point to 'middle' to put the ellipsis () in the middle of
# the label, 'right' to put it at the right end of the label, and 'left' to
# put it at the left end.
metadata_edit_elide_labels = True
metadata_edit_bulk_cc_label_length = 25
metadata_edit_single_cc_label_length = 12
metadata_edit_elision_point = 'right'

#: 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']

#: 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
```

(continues on next page)

(continued from previous page)

```
# 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 book details in the interface
# Change the font size at which book details are rendered in the side panel and
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0

#: What format to default to when using the "Unpack book" feature
# The "Unpack book" feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
# to that format rather than your output format preference.
# Set to a value of 'remember' to use whichever format you chose last time you
# used the "Unpack book" feature.
# Examples:
#   default_tweak_format = None       (Use output format)
#   default_tweak_format = 'EPUB'
#   default_tweak_format = 'remember'
default_tweak_format = None

#: Do not preselect a completion when editing authors/tags/series/etc.
# This means that you can make changes and press Enter and your changes will
# not be overwritten by a matching completion. However, if you wish to use the
```

(continues on next page)

(continued from previous page)

```
# 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.
preselect_first_completion = 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'

#: Recognize numbers inside text when sorting
# This means that when sorting on text fields like title the text "Book 2"
# will sort before the text "Book 100". If you want this behavior, set
# numeric_collation = True note that doing so will cause problems with text
# that starts with numbers and is a little slower.
numeric_collation = False

#: 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 = ()

#: Show the Saved searches box in the Search bar
# In newer versions of calibre, only a single button that allows you to add a
# new Saved search is shown in the Search bar. If you would like to have the
# old Saved searches box with its two buttons back, set this tweak to True.
show_saved_search_box = False

#: Exclude fields when copy/pasting metadata
```

(continues on next page)

(continued from previous page)

```
# You can ask calibre to not paste some metadata fields when using the
# Edit metadata->Copy metadata/Paste metadata actions. For example,
# exclude_fields_on_paste = ['cover', 'timestamp', '#mycolumn']
# to prevent pasting of the cover, Date and custom column, mycolumn.
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
```

12.4 Membatalkan ikon, templat, dan lain-lain

Note: calibre mempunyai sokongan langsung untuk tema ikon, terdapat beberapa tema ikon tersedia untuk calibre, yang anda boleh guna dengan pergi ke *Keutamaan→Antaramuka→Penampilan→Ubah tema Ikon*. Tema ikon menggunakan mekanisma yang serupa sepertimana yang telah dijelaskan di bawah untuk membatalkan sumber statik.

calibre allows you to override the static resources, like icons, JavaScript and templates for the metadata jacket, catalogs, etc. with customized versions that you like. All static resources are stored in the resources sub-folder of the calibre install location. On Windows, this is usually `C:\Program Files\Calibre2\app\resources`. On macOS, `/Applications/calibre.app/Contents/Resources/resources/`. On Linux, if you are using the binary installer from the calibre website it will be `/opt/calibre/resources`. These paths can change depending on where you choose to install calibre.

Anda sepatutnya tidak mengubah fail di daklam folder sumber, kerana perubahan yang anda buat akan ditulis-ganti pada kemaskini calibre yang berikutnya. Oleh itu, pergi ke *Keutamaan→Lanjutan→Pelbagai* kemudian klik *Buka direktori konfigurasi calibre*. Di dalam direktori konfigurasi ini, cipta satu sub-folder yang dikenali sebagai resources dan masukkan fail yang anda mahu gantikan di dalamnya. Letak fail di dala folder yang berkenaan, sebagai contoh letak imej di dalam `resources/images`, dan lain-lain dokumen. calibre akan guna secara automatik fail suai anda dalam keutamaan membatalkan fail terbina-dalam pada pembukaan berikunya.

Sebagai contoh, jika anda mahu mengubah ikon untuk tindakan *Buang buku*, anda perlu lihat dahulu folder sumber terbina-dalam dan lihat fail yang berkaitan ialah `resources/images/remove_books.png`. Anggap anda sudah memiliki ikon pengganti dalam format PNG iaitu `my_remove_books.png` yang mana anda mesti menyimpannya dalam direktori konfigurasi sebagai `resources/images/remove_books.png`. Semua ikon yang digunakan oleh antaramuka pengguna calibre berada di dalam `resources/images` dan juga sub-foldernya.

12.5 Mencipta tema ikon anda sendiri untuk calibre

JIka anda sudah menghasilkan satu set ikon yang cantik dan mahu kongsikanya dengan lain-lain pengguna calibre melalui sokongan tema ikon terbina-dalam calibre, anda boleh pakejkan ikon anda menjadi sebuah tema. Untuk membuatnya, pergi ke *Keutamaan→Pelbagai→Cipta tema ikon*, kemudian pilih folder yang mengandungi ikon tersebut (biasanya folder `resources/images` di dalam direktori konfig calibre, seperti yang telah djelaskan di atas). Seterusnya isikan maklumat data meta tema dan akhir sekali klik OK. Satu fail zip yang mengandungi ikon tema terhasil. Anda boleh muat naik tema ikon ini ke forum calibre di Mobileread dan saya akan jadikan tema anda tersedia melalui sistem tema ikon terbina-dalam calibre.

12.6 Menyuaikan calibre dengan pemalam

calibre mempunyai reka bentuk yang sangat modular. Hampir semua kefungsian dalam calibre berasal dalam bentuk pemalam. Pemalam digunakan untuk pertukaran format, untuk memuat turun berita (dikenali sebagai resepi), untuk pelbagai komponen antaramuka pengguna, untuk sambung ke peranti yang berbeza, untuk proses fail bila menambahnya ke calibre dan seterusnya. Anda boleh dapatkan senarai semua pemalam terbina-dalam di dalam calibre dengan pergi ke *Keutamaan→Lanjutan→Pemalam*.

Anda boleh menulis pemalam anda sendiri untuk suaikan dan tambah baik kelakuan calibre. Seni bina pemalam di dalam calibre adalah sangat mudah, sila rujuk tutorial

CHAPTER 13

Antaramuka Baris Perintah

```
kovid giskard ~/work/libprs500/src/libprs500/manual $ 
```

Note: Pada MacOS, alat baris perintah disediakan dalam berkas calibre, contohnya, jika anda memasang calibre di dalam `/Applications` alat baris perintah berada di dalam `/Applications/calibre.app/Contents/MacOS/`.

13.1 Documented commands

13.1.1 `calibre`

```
calibre [options] [path_to_ebook or calibre url ...]
```

Launch the main **`calibre`** Graphical User Interface and optionally add the e-book at path_to_ebook to the database. You can also specify **`calibre`** URLs to perform various different actions, than just adding books. For example:

`calibre`://view-book/test_library/1842/epub

Will open the book with id 1842 in the EPUB format from the library "test_library" in the **`calibre`** E-book viewer. Library names are the folder names of the libraries with spaces replaced by underscores. A full description of the various URL based actions is in the User Manual.

Whenever you pass arguments to **`calibre`** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

`--detach`
Tanggalkan dari terminal kawalan, jika ada (Linux sahaja)

`--help, -h`
tunjuk mesej bantuan ini dan keluar

`--ignore-plugins`
Abai pemalam suai, berguna jika anda memasang pemalam yang menghalang calibre bermula

`--no-update-check`
Jangan periksa kemaskini

`--shutdown-running-calibre, -s`
Menyebabkan calibre sedang dijalankan, jika ada, akan ditutup. Perhatian jika terdapat kerja berjalan, ia akan dihenti paksa secara senyap, oleh itu guna dengan hati-hati.

`--start-in-tray`
Mula diminimumkan pada talam sistem.

`--verbose, -v`
Diabaikan, jangan guna. Hanya untuk sebab legasi

`--version`
tunjuk nombor versi program dan keluar

`--with-library`
Guna pustaka yang berada pada laluan yang dinyatakan.

13.1.2 `calibre-customize`

```
pilihan calibre-customize
```

Suaikan calibre dengan memuatkan pemalam luaran.

Whenever you pass arguments to **`calibre-customize`** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

`--add-plugin, -a`
Tambah pemalam dengan menyatakan laluan ke fail ZIP dikandunginya.

`--build-plugin, -b`
Untuk pembangun pemalam: Laluan ke direktori yang mana anda bangunkan pemalam. Perintah ini akan zipkan pemalam secara automatik dan kemaskini ia didalam calibre.

`--customize-plugin`
Suaikan pemalam. Nyatakan nama pemalan dan rentetan penyuaian dipisah dengan tanda koma.

`--disable-plugin`
Lumpuhkan pemalam bernama

`--enable-plugin`
Benarkan pemalam bernama

`--help, -h`
tunjuk mesej bantuan ini dan keluar

--list-plugins, -l
Senarai semua pemalam dipasang

--remove-plugin, -r
Buang pemalam suaiz mengikut nama. Tidak mempunyai kesan terhadap pemalam terbina-dalam

--version
tunjuk nombor versi program dan keluar

13.1.3 calibre-debug

```
calibre-debug [pilihan]
```

Various command line interfaces useful for debugging calibre. With no options, this command starts an embedded Python interpreter. You can also run the main calibre GUI, the calibre E-book viewer and the calibre editor in debug mode.

It also contains interfaces to various bits of calibre that do not have dedicated command line tools, such as font subsetting, the E-book diff tool and so on.

You can also use **calibre-debug** to run standalone scripts. To do that use it like this:

calibre-debug myscript.py `--` `--option1` `--option2` file1 file2 ...

Everything after the `--` is passed to the script.

Whenever you pass arguments to **calibre-debug** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--add-simple-plugin
Tambah pemalam mudah (iaitu pemalam yang mengandungi fail .py sahaja), dengan menyatakan laluan ke fail py yang mengandungi kod pemalam.

--command, -c
Jalankan kod Python.

--debug-device-driver, -d
Nyahpepijat pengesanan peranti

--default-programs
(Nyah)daftar calibre dari Program Lalai Windows. *--default-programs* (halaman 269) = `(register|unregister)`

--diff
Jalankan alat diff calibre. Contohnya: calibre-debug *--diff* (halaman 269) file1 file2

--edit-book, -t
Lancarkan alat "Sunting buku" dalam mod nyahpepijat.

--exec-file, -e
Jalankan kod Python dalam fail.

--explode-book, -x
Pecahkan buku ke direktori yang dinyatakan. Penggunaan: -x file.epub output_dir Eksport buku sebagai satu koleksi fail dan data meta HTML, yang anda boleh sunting menggunakan alat penyuntingan HTML piawai. Berfungsi baik dengan fail EPUB, AZW3, HTMLZ dan DOCX.

--export-all-calibre-data
Eksport semua data calibre (buku/tetapan/pemalam). Biasanya, anda akan ditanya untuk mengeksport dir dan juga pustaka. Anda juga boleh tentukan ia sebagai argumen baris perintah untuk melangkau soalan. Guna laluan mutlak untuk eksport direktori dan pustaka. Kata kunci khas `"all"` boleh digunakan untuk mengeksport semua pustaka.

--fix-multiprocessing
For internal use

--gui, -g
Jalankan GUI dengan penyahpepijatan dibenarkan. Output nyahpepijat dicetak ke stdout dan stderr.

--gui-debug
Jalankan GUI dengan konsol nyahpepijat, mengelog ke laluan yang dinyatakan. Untuk kegunaan dalaman sahaja, guna pilihan -g untuk jalankan GUI dalam mod nyahpepijat

--help, -h
tunjuk mesej bantuan ini dan keluar

--implode-book, -i
Ledakkan buku terledak sebelum ini. Penggunaan: -i output_dir file.epub Import buku dari fail dalam output_dir yang telah dihasilkan oleh perintah terdahulu *`--explode-book`* (halaman 269). Pastikan jenis fail yang sama dinyatakan.

--import-calibre-data
Import data calibre diimport sebelum ini

--inspect-mobi, -m
Periksa fail MOBI pada laluan yang dinyatakan

--paths
Outputkan laluan yang perlu untuk pasang persekitaran calibre

--reinitialize-db
Awalkan-semula pangkalan data calibre sqlite pada laluan yang dinyatakan. Berguna untuk pulih dari kerosakan db.

--run-plugin, -r
Jalankan pemalam yang menyediakan antaramuka baris perintah. Contohnya calibre-debug -r `"Add Books"` `--` file1 `--option1` Segalanya selepas `--` akan dilepasi ke pemalam sebagai argumen.

--shutdown-running-calibre, -s
Menyebabkan calibre sedang dijalankan, jika ada, akan ditutup. Perhatian jika terdapat kerja berjalan, ia akan diheti paksa secara senyap, oleh itu guna dengan hati-hati.

--subset-font, -f
Subset fon yang dinyatakan. Guna `--` selepas pilihan ini untuk melepaskan pilihan ke program subsetting fon.

--test-build
Uji modul binari dalam binaan

--version
tunjuk nombor versi program dan keluar

--viewer, -w
Jalankan pelihat E-buku dalam mod nyahpepijat

13.1.4 calibre-server

```
calibre-server [options] [path to library folder...]
```

Start the calibre Content server. The calibre Content server exposes your calibre libraries over the internet. You can specify the path to the library folders as arguments to **calibre-server**. If you do not specify any paths, all the libraries that the main calibre program knows about will be used.

Whenever you pass arguments to **calibre-server** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--access-log
Laluan ke fail log capaian. Log ini mengandungi maklumat berkenaan klien yang menyambung ke pelayan dan membuat permintaan. Secara lalai tiada pengelogan capaian dibuat.

--ajax-timeout
Time (in seconds) to wait for a response from the server when making queries.

--auth-mode
Pilih jenis pengesahihan yang digunakan. Tetapkan mod pengesahihan HTTP yang digunakan oleh pelayan. Tetapkan ke "basic" jika anda mahu letak pelayan ini disebalik proksi SSL. Jika tidak, biarkannya sebagai "auto", yang mana akan gunakan "basic" jika SSL dikonfigur jika sebaliknya ia akan gunakan "digest".

--auto-reload
Muat semula pelayan secara automatik bila kod sumber berubah. Berguna untuk pembangunan. Anda patut juga nyatakan nilai kecil untuk had masa tamat dimatikan.

--ban-after
Bilangan kegagalan daftar masuk untuk disekat. The number of login failures after which an IP address is banned

--ban-for
Ban IP addresses that have repeated login failures. Temporarily bans access for IP addresses that have repeated login failures for the specified number of minutes. Useful to prevent attempts at guessing passwords. If set to zero, no banning is done.

--book-list-mode
Choose the default book list mode. Set the default book list mode that will be used for new users. Individual users can override the default in their own settings. The default is to use a cover grid.

--compress-min-size
Saiz minimum yang membalas penggunaan mampatan data (dalam bait).

--custom-list-template
Path to a JSON file containing a template for the custom book list mode. The easiest way to create such a template file is to go to Preferences-> Sharing over the net-> Book list template in calibre, create the template and export it.

--daemonize
Run process in background as a daemon (Linux only).

--displayed-fields
Restrict displayed user-defined fields. Comma separated list of user-defined metadata fields that will be displayed by the Content server in the /opds and /mobile views. If you specify this option, any fields not in this list will not be displayed. For example: my_rating,my_tags

--enable-allow-socket-preallocation, --disable-allow-socket-preallocation
Socket pre-allocation, for example, with systemd socket activation. Secara lalai, pilihan ini dibenarkan.

--enable-auth, --disable-auth
Pengesahihan berasaskan kata laluan untuk mencapai pelayan. Normally, the server is unrestricted, allowing anyone to access it. You can restrict access to predefined users with this option. Secara lalai, pilihan ini dilumpuhkan.

--enable-fallback-to-detected-interface, --disable-fallback-to-detected-interface
Jatuh-balik ke antaramuka auto-dikesan. Jika atas beberapa sebab pelayan tidak dapat mengikat ke antaramuka yang dinyatakan dalam pilihan listen_on, maka ia akan cuba mengesan satu antaramuka yang bersamnung ke dunia luar yang mengikatnya. Secara lalai, pilihan ini dibenarkan.

--enable-local-write, --disable-local-write
Allow un-authenticated local connections to make changes. Normally, if you do not turn on authentication, the server operates in read-only mode, so as to not allow anonymous users to make changes to your calibre libraries. This option allows anybody connecting from the same computer as the server is running on to make changes. This is useful if you want to run the server without authentication but still use calibredb to make changes to your calibre libraries. Note that turning on this option means any program running on the computer can make changes to your calibre libraries. Secara lalai, pilihan ini dilumpuhkan.

--enable-log-not-found, --disable-log-not-found
Log HTTP 404 (Not Found) requests. Normally, the server logs all HTTP requests for resources that are not found. This can generate a lot of log spam, if your server is targeted by bots. Use this option to turn it off. Secara lalai, pilihan ini dibenarkan.

--enable-use-bonjour, --disable-use-bonjour
Iklankan suapan OPDS melalui BonJour. Iklankan suapan OPDS melalui perkhidmatan BonJour, supaya apl bacaan berasaskan OPDS dapat mengesan dan sambung ke pelayan secara automatik. Secara lalai, pilihan ini dibenarkan.

--enable-use-sendfile, --disable-use-sendfile
Pemindahan fail salinan sifar untuk tingkatkan prestasi. Ia akan gunakan pemindahan dalam-kernel bersalinan-sifar bila menghantar fail menerusi rangkaian, yang dapat meningkatkan prestasi. Walaubagaimanapun, ia boleh menyebabkan pemindahan fail rosak pada beberapa sistem fail bermasalah. Jika anda menghadapi pemindahan fail rosak, matikannya. Secara lalai, pilihan ini dibenarkan.

--help, -h
tunjuk mesej bantuan ini dan keluar

--ignored-fields
Medan data meta ditakrif-pengguna yang diabaikan. Comma separated list of user-defined metadata fields that will not be displayed by the Content server in the /opds and /mobile views. For example: my_rating,my_tags

--listen-on
Antaramuka yang didengari untuk penyambungan. The default is to listen on all available IPv4 interfaces. You can change this to, for example, "127.0.0.1" to only listen for connections from the local machine, or to "::" to listen to all incoming IPv6 and IPv4 connections.

--log
Laluan ke fail log untuk log pelayan. Log ini mengandungi maklumat dan ralat pelayan, bukan log capaian. Secara lalai ia ditulis ke stdout.

--manage-users
Urus pangkalan data pengguna yang dibenarkan untuk sambung ke pelayan ini. Lihat juga pilihan *--userdb* (halaman 274).

--max-header-line-size
Saiz maksimum bagi pengepala HTTP (dalam KB).

--max-job-time
Masa maksimum untuk proses pekerja. Amaun masa maksimum proses pekerja dibenarkan untuk berjalan (dalam minit). Tetapkan pada sifar untuk tanpa had.

--max-jobs
Bilangan maksimum proses pekerja. Worker processes are launched as needed and used for large jobs such as preparing a book for viewing, adding books, converting, etc. Normally, the max. number of such processes is based on the number of CPU cores. You can control it by this setting.

--max-log-size
Saiz fail log maks. (dalam MB). Saiz maksimum fail log, dijana oleh pelayan. Bila log menjadi lebih besar berbanding saiz ini, ia diputarkan secara automatik. Tetapkan ke sifar untuk lumpuhkan putaran log.

--max-opds-items
Jumlah maksimum buku dalam suapan OPDS. Bilangan maksimum buku yang mana pelayan akan kembalikan dalam satu suapan permerolehan OPDS.

--max-opds-ungrouped-items
Jumlah maksimum item tidak dikelompok dalam suapan OPDS. Kelompok item dalam kategori seperti pengarang/tag mengikut huruf pertama yang mana lebih dari bilangan item ini. Tetapkan pada sifar untuk lumpuhkan.

--max-request-body-size
Saiz maksimum dibenarkan untuk memuat naik fail ke dalam pelayan (dalam MB).

--num-per-page
Bilangan buku yang ditunjukkan dalam satu halaman. Bilangan buku yang ditunjukkan dalam satu halaman di dalam pustaka.

--pidfile
Tulis PID proses ke fail tertentu

--port
Port yang didengari untuk penyambungan.

--search-the-net-urls
Path to a JSON file containing URLs for the `"Search the internet"` feature. The easiest way to create such a file is to go to Preferences-> Sharing over the net->Search the internet in calibre, create the URLs and export them.

--shutdown-timeout
Jumlah masa dalam saat untuk menunggu penutupan bersih.

--ssl-certfile
Laluan ke fail sijil SSL.

--ssl-keyfile
Laluan ke fail kunci persendirian SSL.

--timeout
Masa (dalam saat) selepas sambungan melahu ditutup.

--trusted-ips
Allow un-authenticated connections from specific IP addresses to make changes. Normally, if you do not turn on authentication, the server operates in read-only mode, so as to not allow anonymous users to make changes to your calibre libraries. This option allows anybody connecting from the specified IP addresses to make changes. Must be a comma separated list of address or network specifications. This is useful if you want to run the server without authentication but still use calibredb to make changes to your calibre libraries. Note that turning on this option means anyone connecting from the specified IP addresses can make changes to your calibre libraries.

--url-prefix
Awalan untuk ditambah kepada semua URL. Useful if you wish to run this server behind a reverse proxy. For

example use, /calibre as the URL prefix.

--userdb
Path to the user database to use for authentication. The database is a SQLite file. To create it use *--manage-users* (halaman 272). You can read more about managing users at: https://manual.calibre-ebook.com/ms/server.html#managing-user-accounts-from-the-command-line-only

--version
tunjuk nombor versi program dan keluar

--worker-count
Bilangan bebenang pekerja yang digunakan untuk memproses permintaan.

13.1.5 calibre-smtp

```
calibre-smtp [options] [dari teks]
```

Hantar mel menggunakan protokol SMTP. **calibre-smtp** mempunyai dua mod operasi. Dalam mod gubah anda nyatakan daripada dan teks serta ia digunakan untuk bina dan hantar mesej emel. Dalam mod penapis, **calibre-smtp** membaca mesej emel lengkap dari STDIN dan hantar ia.

teks ialah badan mesej emel. Jika teks tidak dinyataka, satu mesej emel lengkap dibaca dari STDIN. from ialah alamat emel pengirim dan to ialah alamat emel penerima. Bila satu emel lengkap telah dibaca dari STDIN, from dan to hanya digunakan dalam perundingan SMTP, pengepala mesej tidak diubahsuai.

Whenever you pass arguments to **calibre-smtp** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--fork, -f
Cabangkan dan hantar mesej disebalik tabir. Jika anda guna pilihan ini, anda patut guna *--outbox* (halaman 274) untuk mengendali kegagalan penghantaran.

--help, -h
tunjuk mesej bantuan ini dan keluar

--localhost, -l
Nama hos bagi localhost. Digunakan bila menyambung ke pelayan SMTP.

--outbox, -o
Laluan ke folder maildir untuk menyimpan mesej emel yang gagal.

--timeout, -t
Had masa tamat untuk sambungan

--verbose, -v
Jadi lebih berjela

--version
tunjuk nombor versi program dan keluar

COMPOSE MAIL

Pilih untuk gubah emel. Diabaikan jika teks tidak dinyatakan

`--attachment, -a`
Fail untuk dilampir ke emel

`--subject, -s`
Subjek emel

SMTP RELAY

Pilih untuk guna pelayan geganti SMTP untuk menghantar mel. Calibre akan cuba menghantar emel secara langsung melainkan --relay dinyatakan.

`--cafile`
Path to a file of concatenated CA certificates in PEM format, used to verify the server certificate when using TLS. By default, the system CA certificates are used.

`--dont-verify-server-certificate`
Do not verify the server certificate when connecting using TLS. This used to be the default behavior in calibre versions before 3.27. If you are using a relay with a self-signed or otherwise invalid certificate, you can use this option to restore the pre 3.27 behavior

`--encryption-method, -e`
Kaedah penyulitan yang digunakan bila menyambung ke geganti. Pilihan adalah TLS, SSL dan TIADA. Lalai ialah TLS. Amaran: Memilih TIADA adalah sangat tidak selamat

`--password, -p`
Kata laluan untuk geganti

`--port`
Port untuk sambung ke pelayan geganti. Lalai ialah guna 465 jika kaedah penyulitan ialah SSL dan 25 sebaliknya.

`--relay, -r`
Pelayan geganti SMTP yang digunakan unutk menghantar mel.

`--username, -u`
Nama pengguna untuk geganti

13.1.6 `calibredb`

```
calibredb command [options] [arguments]
```

`calibredb` is the command line interface to the calibre database. It has several sub-commands, documented below.

`calibredb` can be used to manipulate either a calibre database specified by path or a calibre *Content server* running either on the local machine or over the internet. You can start a calibre *Content server* using either the **`calibre-server`** program or in the main calibre program click *Connect/share → Start Content server*. Since **`calibredb`** can make changes to your calibre libraries, you must setup authentication on the server first. There are two ways to do that:

- If you plan to connect only to a server running on the same computer, you can simply use the `--enable-local-write` option of the content server, to allow any program, including calibredb, running on the local computer to make changes to your calibre data. When running the server from the main calibre program, this option is in *Preferences→Sharing over the net→Advanced*.

- If you want to enable access over the internet, then you should setup user accounts on the server and use the `--username` (halaman 277) and `--password` (halaman 277) options to **`calibredb`** to give it access. You can setup user authentication for **`calibre-server`** by using the `--enable-auth` option and using `--manage-users` to create the user accounts. If you are running the server from the main calibre program, use *Preferences→Sharing over the net→Require username/password*.

To connect to a running Content server, pass the URL of the server to the `--with-library` (halaman 277) option, see the documentation of that option for details and examples.

- *PILIHAN SEJAGAT* (halaman 277)
- *list* (halaman 277)
- *add* (halaman 278)
  - *DITAMBAH DARI DIREKTORI* (halaman 279)
- *remove* (halaman 279)
- *add_format* (halaman 279)
- *remove_format* (halaman 280)
- *show_metadata* (halaman 280)
- *set_metadata* (halaman 280)
- *export* (halaman 281)
- *catalog* (halaman 282)
  - *EPUB PILIHAN* (halaman 282)
- *saved_searches* (halaman 284)
- *add_custom_column* (halaman 284)
- *custom_columns* (halaman 284)
- *remove_custom_column* (halaman 285)
- *set_custom* (halaman 285)
- *restore_database* (halaman 285)
- *check_library* (halaman 286)
- *list_categories* (halaman 286)
- *backup_metadata* (halaman 287)
- *clone* (halaman 287)
- *embed_metadata* (halaman 287)
- *search* (halaman 288)

PILIHAN SEJAGAT

--help, -h
tunjuk mesej bantuan ini dan keluar

--library-path, --with-library
Laluan ke pustaka calibre. Lalai gunakan laluan tersimpan dalam tetapan. Anda juga boleh sambungkan dengan pelayan Kandungan calibre untuk membuat tindakan terhadap pustaka jauh. Untuk membuatnya gunakan URL dalam bentuk: http://hostname:port/#library_id sebagai contoh, http://localhost:8080/#mylibrary. library_id ialah id pustaka bagi pustaka yang mahu disambungkan dengan pelayan Kandungan. Anda boleh gunakan nilai khas library_id - untuk dapatkan senarai id pustaka yang tersedia dalam pelayan. Untuk perincian bagaimana hendak memasang melalui pelayan Kandungan, sila rujuk https://manual.calibre-ebook.com/ms/generated/ms/calibredb.html.

--password
Kata laluan untuk menyambung dengan pelayan Kandungan calibre. Untuk membaca kata laluan dari input piawai, gunakan nilai khas: <stdin>. Manakala, untuk membaca kata laluan dari fail, guna: <f:/path/to/file> (i.e. <f: diikuti dengan laluan penuh ke fail dan juga tanda >). Kurungan seperti di atas diperlukan, ingat kena kurungkan.

--username
Nama pengguna untuk menyambung dengan pelayan Kandungan calibre

--version
tunjuk nombor versi program dan keluar

list

```
calibredb list [options]
```

Senaraikan buku tersedia dalam pangkalan data calibre.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--ascending
Isih keputusan dalam tertib menaik

--fields, -f
Medan yang dipaparkan bila menyenaraikan buku dalam pangkalan data. Seharusnya senarai medan dipisah-tanda-koma. Medan tersedia: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, timestamp, title, uuid Lalai: % lalai. Medan khas "all" boleh digunakan untuk pilih semua medan. Selain dari medan terbina-dalam seperti di atas, medan suai juga tersedia seperti *field_name, sebagai contoh, untuk medan suai #rating, gunakan nama: *rating

--for-machine
Jalan output dalam format JSON, yang lebih sesuai untuk penghuraian mesin. Menyebabkan lebar garisan dan pilihan pemisah diabaikan.

--limit
Bilangan keputusan makismum untuk dipaparkan. Lalai: semua

--line-width, -w
Lebar maksimum baris tunggal dalam output. Lalai ialah mengesan saiz skrin.

--prefix
Awalan untuk semua laluan fail. Lalai ialah laluan mutlak ke folder pustaka.

--search, -s
Tapis keputusan dengan pertanyaan gelintar. Bagi format pertanyaan gelintar, sila rujuk dokumentasi berkaitan gelintar dalam Panduan Pengguna. Lalai adalah tidak membuat penapisan.

--separator
Rentetan yang digunakan untuk memisahkan medan. Lalai ialah jarak.

--sort-by
Medan yang mana mengisih keputusan. Medan tersedia: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, timestamp, title, uuid Lalai: id

add

```
calibredb add [options] fail1 fail2 fail3 ...
```

Tambah fail dinyatakan sebagai buku ke pangkalan data. Anda juga boleh nyatakan direktori, sila rujuk pilihan berkenaan direktori di bawah.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--authors, -a
Tetapkan pengarang buku yang ditambah

--automerge, -m
If books with similar titles and authors are found, merge the incoming formats (files) automatically into existing book records. A value of "ignore" means duplicate formats are discarded. A value of "overwrite" means duplicate formats in the library are overwritten with the newly added files. A value of "new_record" means duplicate formats are placed into a new book record.

--cover, -c
Laluan ke kulit buku digunakan untuk buku yang ditambah

--duplicates, -d
Add books to database even if they already exist. Comparison is done based on book titles and authors. Note that the *--automerge* (halaman 278) option takes precedence.

--empty, -e
Tambah buku kosong (buku tanpa format)

--identifier, -I
Tetapkan pengecam unutk buku ini, contohnya -I asin:XXX -I isbn:YYY

--isbn, -i
Tetapkan ISBN buku yang ditambah

--languages, -l
Senarai dipisah tanda koma bahasa (terbaik gunakan kod bahasa ISO639, walaupun sesetengah nama bahasa juga dikenalpasti)

--series, -s
Tetapkan siri bagi buku yang ditambah

--series-index, -S
Tetapkan nombor siri bagi buku yang ditambah

--tags, -T
Tetapkan tag bagi buku yang ditambah

--title, -t
Tetapkan tajuk buku yang ditambah

DITAMBAH DARI DIREKTORI

Pilihan untuk mengawal penambahan buku dari direktori. Secara lalai hanya fail yang mempunyai sambungan jenis fail e-buku yang diketahui boleh ditambah.

--add
Satu pola (glob) nama fail, fail yang sepadan dengan pola ini akan diabaikan bila mengimbas direktori untuk dapatkan fail, walaupun jika ia bukanlah jenis fail e-buku yang dikenali. Boleh dinyatakan lebih dari sekali untuk pola berbilang.

--ignore
Satu pola (glob) nama fail, fail yang sepadan dengan pola ini akan diabaikan bila mengimbas direktori untuk dapatkan fail. Boleh dinyatakan lebih dari sekali untuk pola berbilang. Sebagai contoh: *.pdf akan abaikan semua fail pdf

--one-book-per-directory, -1
Anggap setiap direktori hanya mempunyai satu buku logikal dan semua fail berada di dalamnya adalah format ebuku yang berlainan bagi buku tersebut

--recurse, -r
Proses direktori secara rekursif

remove

```
calibredb remove ids
```

Buang buku yang dikenalpasti dengan id dari pangkalan data. id seharusnya senarai terpisah-tanda-koma nombor id (anda boleh dapatkan nombor id dengan menggunakan perintah gelintar). Sebagai contoh, 23,34,57-85 (bila menyatakan julat, nombor terakhir dalam julat tidak disertakan).

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--permanent
Jangan guna tong sampah kitar semula

add_format

```
calibredb add_format [options] id fail_ebuku
```

Tambah e-buku dalam fail_ebuku kepada format tersedia untuk buku logikal yang dkenalpasti dengan id. Anda boleh dapatkan id menerusi perintah gelintar. Jika format sudah wujud, ia akan digantikan, melainkan pilihan tidak ganti dinyatakan.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--dont-replace
Jangan ganti format jika ia sudah wujud

remove_format

```
calibredb remove_format [options] id fmt
```

Buang format fmt dari buku logikal yang dikenalpasti dengan id. Anda boleh dapatkan id menerusi perintah gelintar. fmt seharusnya sambungan fail seperti LRF atau TXT atau EPUB. Jika buku logikal tidak mempunyai fmt, jangan buat apa-apa.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

show_metadata

```
calibredb show_metadata [options] id
```

Tunjuk data meta yang tersimpan dalam pangkalan data calibre bagi buku yang dikenalpasti oleh id. id ialah nombor id dari perintah gelintar.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--as-opf
Cetak data meta dalam bentuk OPF (XML)

set_metadata

```
calibredb set_metadata [options] id [/laluan/ke/metadata.opf]
```

Tetapan data meta yang tersimpan dalam pangkalan data calibre bagi buku yang dikenalpasti dengan id dari fail OPF metadata.opf. id ialah nombor id dari perintah gelintar. Anda boleh dapatkan format OPF dengan menggunakan --as-opf yang ditukar ke perintah show_metadata. Anda juga boleh tetapkan data meta bagi medan secara individu dengan pilihan --field. Jika anda guna pilihan --field, tidak perlu nyatakan fail OPF.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--field, -f
Medan yang ditetapkan. Format ialah field_name:value, contohnya: *--field* (halaman 280) tags:tag1,tag2. Guna *--list-fields* (halaman 280) untuk dapatkansenarai bagi semua nama medan. Anda boleh nyatakan pilihan ini berbilang kali untuk tetapkan medan berbilang. Perhatian: Bagi bahasa anda mesti guna kod bahasa ISO639 (contohnya en untuk bahasa Inggeris, ms untuk Bahasa Melayu dan sebagainya). Bagi pengecam, sintaks ialah *--field* (halaman 280) identifiers:isbn:XXXX,doi:YYYYY. Bagi medan boolean (yes/no) gunakan true dan false atau yes dan no.

--list-fields, -l
Senarai nama medan data meta yang boleh digunakan dengan pilihan *--field* (halaman 280)

export

```
calibredb export [options] ids
```

Eksport buku yang dinyatakan dengan id (senarai dipisah-tanda-koma) ke sistem fail. Operasi eksport simpan semua format buku, kulit bukunya dan data meta (dalam fail opf). Anda boleh dapatan nombor id melalui perintah gelintar.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--all
Eksport semua buku dalam pangkalan data, mengabaikan senarai id.

--dont-asciiize
Have calibre convert all non English characters into English equivalents for the file names. This is useful if saving to a legacy filesystem without full support for Unicode filenames. Menyatakan suis ini akan matikan kelakuan ini.

--dont-save-cover
Biasanya, calibre akan simpan kulit buku dalam fail berasingan bersama-sama dengan fail ebuku yang sebenarnya. Menyatakan suis ini akan matikan kelakuan ini.

--dont-update-metadata
Biasanya, calibre akan mengemaskini data meta dalam fail tersimpan dari pustaka calibre. Menjadikan proses penyimpanan ke dalam cakera lebih lambat. Menyatakan suis ini akan matikan kelakuan ini.

--dont-write-opf
Biasanya, calibre akan tulis data meta ke dalam fail OPS secara berasingan bersama-sama dengan fail ebuku yang sebenarnya. Menyatakan suis ini akan matikan kelakuan ini.

--formats
Senarai format dipisah-koma untuk simpan setiap buku. Secara lalai semua format yang tersedia disimpan.

--progress
Kemajuan laporan

--replace-whitespace
Ganti ruang putih dengan underscore.

--single-dir
Eksport semua buku ke dalam satu direktori

--template
Templat yang mengawal nama fail dan struktur fail tersimpan. Lalai ialah "{author_sort}/{title}/{title} - {authors}" yang akan menyimpan buku ke dalam sub-direktori per-pengarang dengan nama fail yang mengandungi tajuk dan pengarang. Kawalan yang tersedia adalah: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt
Format yang memaparkan tarikh. %d - hari, %b - bulan, %m - nombor bulan, %Y - tahun. Lalai ialah: %b, %Y

--to-dir
Eksport buku ke direktori yang dinyatakan, Lalai ialah .

--to-lowercase
Tukar laluan menjadi huruf kecil.

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.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--ids, -i
Senarai terpisah-koma bagi ID pangkalan data ke katalog. Jika diisytihar, `--search` (halaman 282) diabaikan. Lalai: all

--search, -s
Tapis keputusan dengan pertanyaan gelintar. Bagi format pertanyaan gelintar, sila rujuk dokumentasi berkaitan gelintar dalam Panduan Pengguna. Lalai: tanpa penapisan

--verbose, -v
Tunjuk maklumat output terperinci. Berguna untuk penyahpepijatan

EPUB PILIHAN

--catalog-title
Tajuk katalog terjana yang digunakan sebagai tajuk dalam data meta. Lalai: 'My Books' Dilaksana pada: format output AZW3, EPUB, MOBI

--cross-reference-authors
Cipta rujukan-silang pada seksyen Pengarang bagi buku dengan pengarang berbilang. Lalai: 'False' Dilaksana pada: format output AZW3, EPUB, MOBI

--debug-pipeline
Simpan output dari tahap berlainan bagi penukaran talian paip ke direktori yang dinyatakan. Berguna jika anda tidak pasti pada tahap manakah proses penukaran yang mana berlakunya pepijat. Lalai: 'None' Dilaksana pada: format output AZW3, EPUB, MOBI

--exclude-genre
Ungkapan nalar menjelaskan tag untuk diasing sebagai genre. Lalai: '[.+]|^+$' asingkan tag terkurung, seperti '[Project Gutenberg]', dan '+', tag lalai untuk baca buku. Dilaksana pada: format output AZW3, EPUB, MOBI

--exclusion-rules
Nyatakan peraturan yang digunakan untuk asingkan buku dari katalog terjana. Model peraturan pengecualian adalah sama ada ('<rule name>','Tags','<comma-separated list of tags>') atau ('<rule name>','<custom column>','<pattern>'). Contohnya: (('Archived books','#status','Archived'),) akan asingkan buku dengan nilai 'Archived' dalam lajur suai 'status'. Bila peraturan berbilang ditakrif, semua peraturan akan dilaksanakan. Lalai: "(('Catalogs','Tags','Catalog'),)" Dilaksana pada format output AZW3, EPUB, MOBI

--generate-authors
Sertakan seksyen 'Pengarang' dalam katalog. Lalai: 'False' Dilaksanakan pada: format output AZW3, EPUB, MOBI

--generate-descriptions
Sertakan seksyen 'Keterangan' dalam katalog. Lalai: 'False' Dilaksanakan pada: format output AZW3, EPUB, MOBI

--generate-genres
Sertakan seksyen 'Genre' dalam katalog. Lalai: 'False' Dilaksanakan pada: format output AZW3, EPUB, MOBI

--generate-recently-added
Sertakan seksyen 'Ditambah Baru-Baru Ini' dalam katalog. Lalai: 'False' Dilaksanakan pada: format output AZW3, EPUB, MOBI

--generate-series
Sertakan seksyen 'Siri' dalam katalog. Lalai: 'False' Dilaksanakan pada: format output AZW3, EPUB, MOBI

--generate-titles
Sertakan seksyen 'Tajuk' dalam katalog. Lalai: 'False' Dilaksanakan pada: format output AZW3, EPUB, MOBI

--genre-source-field
Medan sumber untuk seksyen 'Genre'. Lalai: 'Tag' Dilaksana pada: format output AZW3, EPUB, MOBI

--header-note-source-field
Medan suai mengandungi teks nota untuk disisip dalam Pengepala Keterangan. Lalai: '' Dilaksana pada: format output AZW3, EPUB, MOBI

--merge-comments-rule
#<custom field>:[before|after]:[True|False] specifying: <custom field> Custom field containing notes to merge with comments [before|after] Placement of notes with respect to comments [True|False] - A horizontal rule is inserted between notes and comments Default: '::' Applies to: AZW3, EPUB, MOBI output formats

--output-profile
Nyatakan profil output. Dalam sesetengah kes, satu profil output diperlukan untuk optimumkan katalog bagi peranti. Contohnya, 'kindle' atau 'kindle_dx' hasilkan Senarai Kandungan berstrukutr dengan Seksyen dan Artikel. Lalai: 'None' Dilaksana pada: format output AZW3, EPUB, MOBI

--prefix-rules
Nyatakan peraturan yang digunakan untuk disertakan awalan yang menunjukkan buku telah dibaca, item senarai idaman dan lain-lain awalan khusus-pengguna. Model untuk peraturan awalan ialah ('<rule name>','<source field>','<pattern>','<prefix>'). Bila peraturan berbilang ditakrif, peraturan pertama yang sepadan akan digunaakn. Lalai: "(('Read books','tags','+','✓'),('Wishlist item','tags','Wishlist','Œ'))" Dilaksana pada format output AZW3, EPUB, MOBI

--preset
Guna praset bernama yang dicipta dengan pembina katalog GUI. Satu praset menyatakan semua tetapan untuk membina katalog. Lalai: 'None' Dilaksana pada: format output AZW3, EPUB, MOBI

--thumb-width
Saiz pembayang (dalam inci) untuk kulit buku dalam katalog. Julat: 1.0 - 2.0 Lalai: '1.0' Dilaksanakan pada: format output AZW3, EPUB, MOBI

--use-existing-cover
Ganti kulit buku sedia ada bila menjana katalog. Lalai: 'False' Dilaksanakan pada: format output AZW3, EPUB, MOBI

saved_searches

```
calibredb saved_searches [options] (list|add|remove)
```

Urus gelintar tersimpan yang disimpan dalam pangkalan data ini. Jika anda cuba menambah pertanyaan dengan nama sedia ada, ia akan digantikan.

Sintaks untuk penambahan:

calibredb **saved_searches** add search_name search_expression

Sintaks untuk pembuangan:

calibredb **saved_searches** remove search_name

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

add_custom_column

```
calibredb add_custom_column [options] label name datatype
```

Cipta lajur suai. label ialah nama mesra mesin bagi lajur. Seharusnya tidak mengandungi jarak atau titik bertindih. nama adalah nama lajur yang boleh dibaca oleh manusia. jenis data adalah salah satu dari: bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--display

Satu kamus pilihan untuk suaikan bagaimana data dalam lajur ini ditafsir. Ia adalah rentetan JSON. Untuk lajur enumerasi, guna *--display* (halaman 284)"{\ "enum_values\ ":[\ "val1\ ", \ "val2\ "]}" Terdapat banyak pilihan yang boleh pergi ke dalam pembolehubah paparan. Pilihan mengikut lajur adalah: composite: composite_template, composite_sort, make_category,contains_html, use_decorations datetime: date_format enumeration: enum_values, enum_colors, use_decorations int, float: number_format text: is_names, use_decorations Cara terbaik mencari gabungan sah ialah dengan mencipta lajur suai bagi jenis bersesuaian dalam GUI kemudian cari dalam sandar OPF bagi sesebuah buku (pastikan OPF baharu telah dicipta semenjak lajur telah ditambah). Anda akan lihat JSON untuk "display" bagi lajur baharu dalam OPF.

--is-multiple

Lajur ini menyimpan tag seperti data (iaitu nilai berbilang dipisah-tanda-koma). Hanya dilaksana jika jenis data adalah teks.

custom_columns

```
calibredb custom_columns [options]


Senaraikan lajur suai tersedia. Tunjuk label dan id lajur.
```

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--details, -d

Tunjuk perincian untuk setiap lajur.

remove_custom_column

```
calibredb remove_custom_column [options] label


Buang lajur suai yang dikenalpasti oleh label. Anda boleh lihat lajur
tersedia dengan perintah custom_columns.
```

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--force, -f
Jangan tanya untuk pengesahan

set_custom

```
calibredb set_custom [options] column id value
```

Tetapkan nilai lajur suai bagi buku yang dikenalpasti oleh id. Anda boleh dapatkan senarai id menggunakan perintah gelintar. Anda boleh dapatkan senarai nama lajur suai menggunakan perintah custom_columns.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--append, -a
Jika lajur menyimpan nilai berbilang, tambah nilai yang dinyatakan pada mana-mana yang ada, selain dari menulis-ganti ia.

restore_database

```
calibredb restore_database [options]
```

Pulih pangkalan data ini dari data meta tersimpan dalam fail OPF dalam setiap direktori bagi pustaka calibre. Ia berguna jika fail metadat.db anda mengalami kerosakan.

AMARAN: Perintah ini sepenuhnya jana semula pangkalan data anda. Anda akan kehilangan semua gelintar tersimpan, kategori pengguna, papan palam, tetapan penukaran per-buku tersimpan, dan resepi suai. Data meta tersimpan hanya benar-benar tepat sepertimana yang ditemui dalam fail OPF.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--really-do-it, -r
Pasti mahu buat pemulihan. Perintah tidak akan berjalan melainkan pilihan ini dinyatakan.

check_library

```
calibredb check_library [options]
```

Buat beberapa semakan pada sistem fail yang mewakili pustaka. Laporan adalah invalid_titles, extra_titles, invalid_authors, extra_authors, missing_formats, extra_formats, extra_files, missing_covers, extra_covers, failed_folders

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--csv, -c
Output dalam CSV

--ignore_extensions, -e
Senarai sambungan dipisah-tanda-koma untuk diabaikan. Lalai: semua

--ignore_names, -n
Senarai nama dipisah-tanda-koma untuk diabaikan. Lalai: semua

--report, -r
Senarai laporan dipisah-tanda-koma. Lalai: semua

list_categories

```
calibredb list_categories [options]
```

Produce a report of the category information in the database. The information is the equivalent of what is shown in the Tag browser.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--categories, -r
Senarai nama carian kategori dipisah-tanda-koma. Lalai: semua

--csv, -c
Output dalam CSV

--dialect
Jenis fail CSV yang dihasilkan. Pilihan: excel, excel-tab, unix

--item_count, -i
Hanya output bilangan item dalam kategori selain dari kiraan per item di dalam kategori

--width, -w
Lebar maksimum baris tunggal dalam output. Lalai ialah mengesan saiz skrin.

backup_metadata

```
calibredb backup_metadata [options]
```

Sandar data meta tersimpan dalam pangkalan data ke dalam fail OPF secara individu dalam setiap direktori buku. Ia biasanya berlaku secara automatik, tetapi anda boleh jalankan perintah ini untuk paksa penjanaan-semula fail OPF, dengan pilihan --all.

Perhatian kebiasaannya tidak perlu membuatnya, kerana fail OPF disandar secara automatik, setiap kali data meta berubah.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--all

Biasanya, perintah ini hanya beroperasi pada buku yang mempunyai fail OPF yang telah luput. Pilihan ini menjadikannya beroperasi pada semua buku.

clone

```
calibredb clone path/to/new/library
```

Create a **clone** of the current library. This creates a new, empty library that has all the same custom columns, Virtual libraries and other settings as the current library.

The cloned library will contain no books. If you want to create a full duplicate, including all books, then simply use your filesystem tools to copy the library folder.

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

embed_metadata

```
calibredb embed_metadata [options] book_id
```

Kemaskin data meta dalam fail buku sebenar yang disimpan dalam pustaka calibre daripada data meta dalam pangkalan data calibre. Kebiasaannya, data meta hanya dikemaskini bila mengeksport fail dari calibre, perintah ini berguna jika anda mahu fail dikemaskini pada tempatnya. Perhatian format fail berbeza mentokong sejumlah data meta yang berbeza. Anda boleh guna nilai khas 'all' bagi book_id untuk kemaskini data meta dalam semua buku. Anda juga boleh nyatakan banyak id buku dengan dipisah tanda jarak dan julat id dipisah dengan tanda sengkang. Contohnya: calibredb **embed_metadata** 1 2 10-15 23

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--only-formats, -f

Hanya kemaskini data meta dalam fail bagi format yang dinyatakan. Nyatakan ia berbilang kali untuk format berbilang. Secara lalai, semua format dikemaskinikan.

search

```
calibredb search [options] search expression
```

Search the library for the specified **search** term, returning a comma separated list of book ids matching the **search** expression. The output format is useful to feed into other commands that accept a list of ids as input.

The **search** expression can be anything from calibre's powerful **search** query language, for example: calibredb **search** author:asimov 'title:"i robot"'

Whenever you pass arguments to calibredb that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

--limit, -l
Bilangan keputusan makismum yang dikembalikann. Lalai ialah semua keputusan.

13.1.7 ebook-convert

```
ebook-convert input_file output_file [options]
```

Tukar format e-buku dari satu format ke format yang lain.

input_file ialah input dan output_file ialah output. Kedua-dua mesti dinyatakan sebagai dua argumen pertama perintah.

Format e-buku output diteka dari sambungan fail output_file. output_file juga boleh menjadi format khas .EXT yang mana EXT ialah sambungan fail output file. Dalam kes ini, nama fail output diambil dari nama fail input. Perhatian nama fail mesti tidak bermula dengan tanda hyphen. Akhir sekali, jika output_file tidak mempunyai sambungan, maka ia dianggap sebagai direktori dan "open ebook" (OEB) yang mengandungi fail HTML yang ditulis ke dalam direktori tersebut. Fail ini biasanya telah dilepasi ke pemalam output.

Selepas menyatakan fail input dan output anda boleh suaikan penukaran dengan menyatakan pelbagai pilihan. Pilihan tersedia bergantung pada jenis fail input dan output. Untuk mendapatkan bantuan nyatakan fail input dan output dan kemudian guna pilihan -h.

Untuk dokumentasi penuh sistem penukaran, lihat *E-book conversion* (halaman 49)

Whenever you pass arguments to **ebook-convert** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

Pilihan dan nilai lalai bagi perubahan pilihan bergantung pada kedua-dua format input dan output, jadi anda seharusnya sentiasa semak dengan:

```
ebook-convert myfile.input_format myfile.output_format -h
```

Dibawah merupakan pilihan umum untuk semua penukaran, diikuti dengan pilihan khusus untuk setiap format input dan output.

- *PILIHAN INPUT* (halaman 289)
- *PILIHAN OUTPUT* (halaman 289)
- *PENAMPILAN* (halaman 290)
- *PEMPROSESAN HEURISTIK* (halaman 293)
- *GELINTAR DAN GANTI* (halaman 293)
- *PENGESANAN STRUKTUR* (halaman 294)

- *ISI KANDUNGAN* (halaman 295)
- *DATA META* (halaman 295)
- *NYAHPEPIJAT* (halaman 296)

`--help, -h`
: tunjuk mesej bantuan ini dan keluar

`--input-profile`
: Nyatakan profil input. Profil input memberikan maklumat sistem penukaran bagaimana hendak tafsir pelbagai maklumat di dalam dokumen input. Sebagai contoh resolusi bergantung pada panjang (iaitu panjang dalam piksel). Pilihan adalah:cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

`--list-recipes`
: Senaraikan nama resepi terbina-dalam. Anda boleh cipta satu e-buku melalui resepi terbina-dalam seperti ini: ebook-convert `"`Recipe Name.recipe`"` output.epub

`--output-profile`
: Nyatakan profil output. Profil output memberitahu sisten penukaran bagaimana hendak optimumkan dokumen yang dihasilkan untuk peranti tertentu (seperti saizkan semula imej untuk saiz skrin peranti). Dalam sesetengah situasi, satu profil output boleh digunakan untuk optimumkan output bagi peranti tersebut, tetapi ia jarang berlaku. Pilihan adalah: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_voyage, kobo, msreader, mobipocket, nook, nook_color, nook_hd_plus, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet

`--version`
: tunjuk nombor versi program dan keluar

PILIHAN INPUT

Pilihan untuk mengawal pemprosesan fail input mobi

`--input-encoding`
: Tentukan pengkodan aksara bagi dokumen yang di inputkan. Jika di setkan, pilihan ini akan membatalkan sebarang pengkodan yang diinyatakan oleh dokumen itu sendiri. Ia sangat berguna bagi dokumen-dokumen yang tidak menyatakan pengkodannya atau yang mempunyai kesilapan pernyataan pengkodan.

PILIHAN OUTPUT

Pilihan untuk mengawal pemprosesan fail output epub

`--dont-split-on-page-breaks`
: Matikan pemisahan pada hentian halaman. Biasanya fail input dipisah secara automatik pada setiap hentian halaman menjadi dua fail. Ini memberikan e-buku output yang boleh dihurai lebih pantas dengan kurang penggunaan sumber. Walaubagaimanapun, pemisahan adalah perlahan dan jika fail sumber anda mengandungi jumlah hentian halaman yang sangat besar, anda patut mematikan pemisahan hentian halaman.

`--epub-flatten`
: Pilihan ini hanya diperlukan jika anda berniat untuk guna EPUB dengan FBReaderJ. Ia akan ratakan sistem fail yang ada di dalam EPUB, meletakkan semua fail dalam aras tertinggi.

`--epub-inline-toc`
: Sisip Isi Kandunagan dalaman yang akan muncul sebagai sebahagian dari kandungan buku utama.

--epub-toc-at-end

Letak Isi Kandungan dalam garis dipenghujung buku selain di permulaan.

--epub-version

Versi fail EPUB yang dijana. EPUB 2 adalah yang paling serasi, hanya guna EPUB 3 jika anda benar-benar memerlukannya.

--extract-to

Ekstrak kandungan fail EPUB yang dijana ke direktori yang dinyatakan. Kandungan direktori adalah mula-mula dipadamkan, oleh itu hati-hati.

--flow-size

Pisah semua fail HTML yang lebih besar dari saiz ini (dalam KB). Ia perlu kerana kebanyakan pembaca EPUB tidak dapat kendalikan saiz fail besar. Lalai 260KB adalah saiz yang diperlukan untuk Adobe Digital Editions. Tetapkan pada 0 untuk lumpuhkan pembahagian berasaskan saiz.

--no-default-epub-cover

Biasanya jika fail input tidak mempunyai kulit luar dan anda tidak nyatakan, kulit lalai dijana dengan tajuk, pengarang, dll. Pilihan ini lumpuhkan penjanaan kulit buku ini.

--no-svg-cover

Jangan guna SVG untuk kulit buku. Guna pilihan ini jika EPUB anda akan diguna dalam peranti yang tidak menyokong format SVG, seperti iPhone atau JetBook Lite. Tanpa pilihan ini, peranti tersebut akan paparkan kulit buku sebagai muka kosong.

--preserve-cover-aspect-ratio

Bila menggunakan kulit hadapan SVG, pilihan ini akan menyebabkan kulit hadapan diskalakan untuk meliputi kawasan skrin yang ada, tetapi masih mengekalkan nisbah bidangnya (nisbah lebar terhadap tinggi). Ini bermaksud akan wujud sempadan putih disisi atau atas dan bawah imej, tetapi imej tidak akan diherotkan. Tanpa pilihan ini imej akan menjadi sedikit herot, tetapi sempadan tidak akan wujud.

--pretty-print

Jika dinyatakan, pemalam output akan cuba cipta output yang boleh dibaca manusia. Mungkin tidak memberi kesan kepada beberapa pemalam output.

--toc-title

Tajuk bagi mana-mana isi kandungan dalaman yang terjana.

PENAMPILAN

Pilihan untuk mengawal penampilan output

--asciiize

Transliterate Unicode characters to an ASCII representation. Use with care because this will replace Unicode characters with ASCII. For instance it will replace `" "` with `"Mikhail Gorbachiov"`. Also, note that in cases where there are multiple representations of a character (characters shared by Chinese and Japanese for instance) the representation based on the current calibre interface language will be used.

--base-font-size

The base font size in pts. All font sizes in the produced book will be rescaled based on this size. By choosing a larger size you can make the fonts in the output bigger and vice versa. By default, when the value is zero, the base font size is chosen based on the output profile you chose.

--change-justification

Ubah kewajaran teks. Nilai `"kiri"` tukar semua teks terwajar di dalam sumber menjadi teks terjajar kiri (iaitu. tidak terwajar). Nilai `"wajar"` tukar semua teks tidak terwajar menjadi terwajar. Nilai `"asal"` (lalai) tidak mengubah kewajaran fail sumber. Perhatian hanya beberapa format output menyokong kewajaran.

--disable-font-rescaling
Lumpuhkan semua penskalaan semula saiz fon.

--embed-all-fonts
Benamkan setiap fon yang dirujuk dalam dokumen input tetapi sebenarnya tidak dibenam. Ia akan menggelintar fon tersebut dalam sistem anda, dan jika temui, ia akan dibenamkan. Pembenaman hanya berlaku jika format yang anda tukar menyokong fon terbenam, seperti EPUB, AZW3, DOCX atau PDF. Sila pastikan anda mempunyai lesen untuk membenamkan fon tersebut untuk digunakan dalam dokumen ini.

--embed-font-family
Benamkan keluarga fon tertentu ke dalam buku. Ini menyatakan fon `"dasar"` yang digunakan untuk buku. Jika dokumen input menyatakan fonnya sendiri, ia mungkin membatalkan fon dasar ini. Anda boleh guna pilihan maklumat gaya penapis untuk membuang fon dari dokumen input. Perhatian pembenaman fon hanya berfungsi dengan beberapa format output, biasanya EPUB, AZW3 dan DOCX.

--expand-css
Secara lalai, calibre akan guna kependekan untuk pelbagai sifat CSS seperti margin, pemadatan, sempadan, dll. Pilihan ini akan menyebabkan ia gunakan bentuk penuh. Perhatian CSS sentiasa dikembangkan bila menjana fail EPUB dengan profil output ditetapkan pada salah satu profil Nook kerana Nook tidak dapat kendalikan kependekan CSS.

--extra-css
Sama ada laluan ke lembaran gaya CSS atau CSS mentah. CSS ini akan ditambah ke peraturan gaya dari fail sumber, supaya ia boleh digunakan untuk batalkan peraturan tersebut.

--filter-css
Satu senarai dipisah tanda koma sifat CSS akan dibuang dari semua peraturan gaya CSS. Ia berguna jika kehadiran beberapa maklumat gaya menghalang ia dari dibatalkan oleh peranti anda. Contohnya: font-family,color,margin-left,margin-right

--font-size-mapping
Pemetaan dari nama fon CSS ke saiz fon dalam pt. Contoh penetapan ialah 12,12,14,16,18,20,22,24. Ini merupakan pemetaan untuk saiz xx-small sehinggalah xx-large, dengan saiz akhir merupakan fon amat besar. Algoritma penskalaan semula fon gunakan saiz ini untuk skalakan semula fon secara pintar. Lalai ialah guna pemetaan berdasarkan pada profil output yang anda pilih.

--insert-blank-line
Sisip baris kosong diantara perenggan. Tidak akan berfungsi jika fail sumber tidak guna perenggan (tag <p> atau <div>).

--insert-blank-line-size
Tetapkan tinggi baris kosong yang disisip (dalam em). Tinggi baris diantara perenggan akan menjadi dua kali ganda nilai yang ditetapkan di sini.

--keep-ligatures
Kekalkan ligatur yang hadir di dalam dokumen input. Ligatur ialah penerapan khas bagi pasangan aksara seperti ff, fi, fl dan lain-lain. Kebanyakan pembaca tidak mempunyai sokongan bagi ligatur di dalam fon lalai mereka, jadi mereka tidak dapat diterap dengan betul. Secara lalai, calibre akan jadikan ligatur dalam bentu pasangan aksara biasa berkenaan. Pilihan ini akan dikekalkan.

--line-height
Tinggi baris dalam pt. Kawal jarak diantara baris berturutan teks. Hanya laksana pada unsur yang tidak takrif tinggi baris mereka sendiri. Dalam kebanyakan kes, pilihan tinggi baris minimum adalah lebih berguna. Secara lalai tiada manipulasi tinggi baris dibuat.

--linearize-tables
Sesetengah dokumen direka teruk dengan jadual yang mana ia mengawal bentangan teks dalam halaman. Bila menukar dokumen ini biasanya teks terkeluar dari halaman dan ada artifak lain. Pilihan ini akan mengekstrak kandungan dari jadual dan persembahkannya dalam fesyen linear.

--margin-bottom
Tetapkan margin bawah dalam pts. Lalai ialah 5.0. Tetapan ini adalah kurang dari sifar yang akan menyebabkan tiada margin ditetapkan (tetapan margin dalam dokumen asal akan dikekalkan). Perhatian: Format berorientasikan halmaan seperti PDF dan DOCX mempunyai tetapan margin mereka masing-masing.

--margin-left
Tetapkan margin kiri dalam pts. Lalai ialah 5.0. Tetapan ini adalah kurang dari sifar yang akan menyebabkan tiada margin ditetapkan (tetapan margin dalam dokumen asal akan dikekalkan). Perhatian: Format berorientasikan halmaan seperti PDF dan DOCX mempunyai tetapan margin mereka masing-masing.

--margin-right
Tetapkan margin kanan dalam pts. Lalai ialah 5.0. Tetapan ini adalah kurang dari sifar yang akan menyebabkan tiada margin ditetapkan (tetapan margin dalam dokumen asal akan dikekalkan). Perhatian: Format berorientasikan halmaan seperti PDF dan DOCX mempunyai tetapan margin mereka masing-masing.

--margin-top
Tetapkan margin atas dalam pts. Lalai ialah 5.0. Tetapan ini adalah kurang dari sifar yang akan menyebabkan tiada margin ditetapkan (tetapan margin dalam dokumen asal akan dikekalkan). Perhatian: Format berorientasikan halmaan seperti PDF dan DOCX mempunyai tetapan margin mereka masing-masing.

--minimum-line-height
Tinggi baris minimum, sebagai peratus unsur saiz fon yang dikira. calibre akan pastikan yang mana setiap unsur mempunyai tinggi baris sekurang-kurangnya ikut tetapan ini, tidak menuruti apakah input dokumen tentukan. Tetapkan sifar untuk lumpuhkan. Lalai ialah 120%. Guna tetapan ini dalam keutamaan ke spesifikasi tinggi baru, melainkan anda tahu apa yang anda buat. Contohnya, anda boleh dapat teks "berjarak ganda dua" dengan menetapkannya menjadi 240.

--remove-paragraph-spacing
Buang penjarakan diantara perenggan. Juga tetapkan inden pada perenggan sebanyak 1.5em. Pembuangan penjarakan tidak berfungsi jika fail sumber tidak guna perenggan (tag <p> atau <div>).

--remove-paragraph-spacing-indent-size
Bila calibre membuang baris kosong diantara perenggan, ia tetapkan inden perenggan secara automatik untuk memastikan perenggan mudah dikenalpasti. Pilihan ini mengawal lebar inden tersebut (dalam em). Jika anda tetapkan nilainya negatif, maka inden yang dinyatakan dalam dokumen input akan digunakan, iaitu calibre tidak mengubah indentasinya.

--smarten-punctuation
Tukar petikan, sengkang dan elipsis biasa pada yang setara setepatnya secara tipografi. Untuk perincian, sila rujuk https://daringfireball.net/projects/smartypants

--subset-embedded-fonts
Sebset semua fon terbenam. Setiap fon terbenam dikurangkan hanya untuk mengandungi glif yang diguna dalam dokumen ini. Ia kurangkan saiz fail fon. Berguna jika anda membenamkan fon besar tertentu yang mengandungi banyak glif yang tidak perlu.

--transform-css-rules
Laluan ke fail yang mengandungi peraturan pengubah gaya CSS dalam buku ini. Cara termudah mencipta fail sebegitu adalah dengan menggunakan bestari untuk mencipta peraturan di dalam GUI calibre. Capai ia dalam seksyen "Penampilan->Ubah gaya" bagi dialog pertukaran. Seusai anda mewujudkan peraturan, anda boleh guna butang "Eksport" untuk menyimpannya ke dalam fail.

--unsmarten-punctuation
Tukar petikan, sengkang dan elipsis yang pelik mengikut format piawainya.

PEMPROSESAN HEURISTIK

Ubah suai teks dan struktur dokumen menggunakan corak umum. Dilumpuhkan secara lalai. Guna --enable-heuristics untuk dibenarkan. Tindakan individu boleh dilumpuhkan dengan pilihan --disable-*.

--disable-dehyphenate
Analisis perkataan bersempang menyeluruhi dokumen. Dokumen itu sendiri digunakan sebagai kamus untuk tentukan sama ada tanda sempang seharusnya dikekalkan atau dibuang.

--disable-delete-blank-paragraphs
Buang perenggan kosong dari dokumen bila ia wujud diantara setiap perenggan lain

--disable-fix-indents
Hidupkan identasi yang dicipta dari entiti jarak tidak-henti berbilang ke dalam inden CSS.

--disable-format-scene-breaks
Penanda hentian adegan terjajar kiri adalah dijajar tengah. Ganti hentian adegan lembut yang guna baris kosong berbilang dengan peratusan mengufuk.

--disable-italicize-common-cases
Cari perkataan dan corak umum yang diwakili dengan condong dan kecondongan mereka.

--disable-markup-chapter-headings
Kesan pengepala dan sub pengepala bab. Ubah ia menjadi tag h2 dan h3. Tetapan ini tidak akan hasilkan TOC, tetapi boleh digunakan di dalam konjuksi dengan pengesanan struktur jika mahu membinanya.

--disable-renumber-headings
Cari kemunculan jujukan tag <h1> atau <h2>. Tag dinombor semula untuk menghindari pemisahan ditengah-tengah pengepala bab.

--disable-unwrap-lines
Nyahlilit baris menggunakan tanda baca dan lain-lain tanda pemformatan.

--enable-heuristics
Benarkan pemprosesan heuristik. Pilihan ini mesti ditetapkan untuk mana-mana pemprosesan heuristik untuk dijalankan.

--html-unwrap-factor
Skala yang digunakan untuk menentukan panjang yang mana baris patut tidak dililit. Nilai yang sah adalah desimal diantara 0 hingga 1. Lalai ialah 0.4, sedikit dibawah panjang baris median. Jika hanya beberapa baris di dalam dokumen memerlukan penyahlilitaan, nilai ini patut dikurangkan.

--replace-scene-breaks
Ganti hentian adegan dengan teks yang dinyatakan. Secara lalai, teks dari dokumen input digunakan.

GELINTAR DAN GANTI

Ubahsuai teks dan struktur dokumen menggunakan corak ditakrif pengguna.

--search-replace
Laluan ke fail yang mengandungi ungkapan nalar gelintar dan ganti. Fail mesti mengandungi bari alternatif bagi ungkapan nalar yang diikuti oleh corak penggantian (yang boleh jadi baris kosong). Ungkapan nalar mesti dalam sintaks ungkapan nalar Python dan fail mesti terenkod UTF-8.

--sr1-replace
Penggantian untuk ganti teks yang ditemui dengan sr1-search.

--sr1-search
Gelintar corak (ungkapan nalar) yang diganti dengan sr1-replace.

--sr2-replace
Penggantian untuk ganti teks yang ditemui dengan sr2-search.

--sr2-search
Gelintar corak (ungkapan nalar) yang diganti dengan sr2-replace.

--sr3-replace
Penggantian untuk ganti teks yang ditemui dengan sr3-search.

--sr3-search
Gelintar corak (ungkapan nalar) yang diganti dengan sr3-replace.

PENGESANAN STRUKTUR

Kawal pengesanan-automatik struktur dokumen.

--chapter
Ungkapan XPath untuk mengesan tajuk bab. Lalai ialah dengan menganggap tag <h1> atau <h2> yang mengandungi perkataan "chapter","book","section", "prologue", "epilogue", atau "part" sebahagian dari tajuk bab serta mana-mana tag yang mempunyai class="chapter". Ungkapan digunakan mesti menilai senarai unsur. Untuk lumpuhkan pengesanan bab, guna ungkapan "/". Sila rujuk Tutorial XPath di dalam Panduan Pengguna calibre untuk bantuan lanjut berkenaan penggunaan fitur ini.

--chapter-mark
Nyatakan bagaimana hendak tanda bab yang dikesan. Nilai "pagebreak" akan menyisip hentian halaman sebelum bab. Nilai "rule" akan menyisip satu baris sebelum bab. Nilai "none" akan lumpuhkan penandaan bab dan nilai "both" akan guna kedua-dua hentian halaman dan baris untuk menanda bab.

--disable-remove-fake-margins
Sesetengah dokumen nyatakan margin halaman dengan menyatakan margin kiri dan kanan pada setiap perenggan secara individu. Calibre akan cuba mengesan dan membuang margin ini. Kadangkala, ia menyebabkan pembuangan margin yang tidak sepatutnya dibuang. Dalam situasi ini anda boleh lumpuhkan pembuangan tersebut.

--insert-metadata
Sisip data meta buku pada permulaan buku. Ia berguna jika pembaca e-buku anda tidak menyokong papar/gelintar data meta secara langsung.

--page-breaks-before
Ungkapan XPath. Hentian halaman disisip sebelum unsur yang dinyatakan. Untuk lumpuhkan penggunaan ungkapan: /

--prefer-metadata-cover
Guna kulit hadapan yang dikesan dari fail sumber dalam keutamaan ke kulit hadapan yang dinyatakan.

--remove-first-image
Buang imej pertama dari e-buku input. Berguna jika dokumen input mempunyai imej kulit buku yang tidak dikenalpasti sebagai kulit buku. Dalam situasi ini, jika anda tetapkan kulit buku dalam calibre, dokumen output akan mempunyai dua imej kulit buku jika anda tidak nyatakan pilihan ini.

--start-reading-at
Ungkapan XPath untuk mengesan lokasi dalam dokumen untuk mula membaca. Sesetengah program membaca e-buku (umumnya Kindle) gunakan lokasi ini sebagai kedudukan untuk membuka buku. Sila rujuk tutorial XPath di dalam Panduan Pengguna calibre untuk bantuan lanjutan berkenaan penggunaan fitur ini.

ISI KANDUNGAN

Kawal penjanaan automatik Isi Kandungan. Secara lalai, jika fail sumber mempunyai Isi Kandungan, ia akan digunakan dalam keutamaan ke yang terjana secara automatik.

--duplicate-links-in-toc
Bila mencipta TOC dari pautan dalam dokumen input, benarkan masukan berganda, iaitu benarkan lebih dari satu masukan dengan teks yang sama, yang mana ia menuju ke lokasi yang berbeza.

--level1-toc
Ungkapan XPath yang menyatakan semua tag perlu ditambah ke dalam Senarai Kandungan pada aras satu. Jika ia dinyatakan, ia mendahului lain-lain bentuk auto-pengesanan. Sila rujuk Tutorial XPath dalam Panduan Pengguna calibre untuk lihat contoh.

--level2-toc
Ungkapan XPath yang menyatakan semua tag perlu ditambah ke dalam Senarai Kandungan pada aras dua. Setiap masukan ditambah di bawah masukan aras satu sebelum ini. Sila rujuk Tutorial XPath dalam Panduan Pengguna calibre untuk lihat contoh.

--level3-toc
Ungkapan XPath yang menyatakan semua tag perlu ditambah ke dalam Senarai Kandungan pada aras tiga. Setiap masukan ditambah di bawah masukan aras dua sebelum ini. Sila rujuk Tutorial XPath dalam Panduan Pengguna calibre untuk lihat contoh.

--max-toc-links
Bilangan pautan maksimum disisip ke dalam TOC. Tetapkan 0 untuk lumpuhkan. Lalai ialah: 50. Pautan hanya ditambah ke TOC jika kurang dari ambang bilangan bab telah dikesan.

--no-chapters-in-toc
Jangan tambah bab dikesan-sendiri ke Isi Kandungan.

--toc-filter
Buang masukan dari Isi Kandungan yang mana tajuk sepadan dengan ungkapan nalar yang dinyatakan. Masukan yang sepadan dan semua anaknya akan dibuang.

--toc-threshold
Jika lebih sedikit berbanding bilangan bab ini dikesan, maka pautan ditambah ke Isi Kandungan. Lalai: 6

--use-auto-toc
Biasanya, jika fail sumber sudah mempunyai Senarai Kandungan, ia digunakan dalam keutamaan untuk auto-jana. Dengan pilihan ini, auto-jana sentiasa digunakan.

DATA META

Pilihan untuk tetapkan data meta dalam output

--author-sort
Rentetan yang digunakan bila mengisih ikut pengarang.

--authors
Tetapkan pengarang. Pengarang berbilang seharusnya dipisah dengan tanda ampersand.

--book-producer
Tetapkan pengeluar buku.

--comments
Tetapkan keterangan e-buku.

--cover
Tetapkan kulit hadapan fail atau URL yang dinyatakan.

--isbn
Tetapkan ISBN buku.

--language
Tetapkan bahasa.

--pubdate
Tetapkan tarikh penerbitan (dianggap dalam zon waktu setempat, melainkan zon waktu dinyatakan secara eksplisit)

--publisher
Tetapkan penerbit e-buku.

--rating
Tetapkan penarafan. Seharusnya nombor diantara 1 hingga 5.

--read-metadata-from-opf, --from-opf, -m
Baca data meta dari fail OPF yang dinyatakan. Data meta dibaca dari fail ini akan membatalkan mana-mana data meta dalam fail sumber.

--series
Tetapkan siri e-buku ini berasal.

--series-index
Tetapkan indeks buku dalam siri ini.

--tags
Tetapkan tag untuk buku. Seharusnya senarai dipisah dengan tanda koma.

--timestamp
Tetapkan setem masa buku (tiada lagi digunakan)

--title
Tetapkan tajuk.

--title-sort
Versi tajuk yang digunakan untuk pengisihan.

NYAHPEPIJAT

Pilihan untuk bantu dengan menyahpepijat pertukaran

--debug-pipeline, -d
Simpan output dari tahap berlainan bagi penukaran talian paip ke direktori yang dinyatakan. Berguna jika anda tidak pasti pada tahap manakah proses penukaran yang mana berlakunya pepijat.

--verbose, -v
Aras kejelaan. Nyatakan berapa kali untuk lebih kejelaan. Menyatakan dua kali akan hasilkan kejelaan penuh, sekali kejelaan sederhana dan sifar paling kurang kejelaan.

13.1.8 ebook-edit

```
ebook-edit [opts] [path_to_ebook] [name_of_file_inside_book ...]
```

Launch the calibre Edit book tool. You can optionally also specify the names of files inside the book which will be opened for editing automatically.

Whenever you pass arguments to **ebook-edit** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--detach
Tanggalkan dari terminal kawalan, jika ada (Linux sahaja)

--help, -h
tunjuk mesej bantuan ini dan keluar

--version
tunjuk nombor versi program dan keluar

13.1.9 ebook-meta

```
ebook-meta ebook_file [options]
```

Baca/Tulis data meta dari/ke fail e-buku.

Format disokong untuk data meta pembacaan: azw, azw1, azw3, azw4, 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

Format disokong untuk data meta penulisan: azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

Jenis fail berbeza menyokong jenis data meta yang berlainan. Jika anda cuba tetapkan beberapa data meta pada satu jenis fail yang tidak menyokongnya,

> data meta tersebut akan diabaikan secara senyap.

Whenever you pass arguments to **ebook-meta** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--author-sort
Rentetan yang digunakan bila mengisih berdasarkan pengarang. Jika tidak dinyatakan dan pengarang dinyatakan, ia akan auto-jana daripada pengarang.

--authors, -a
Tetapkan pengarang. Pengarang berbilang seharusnya diasing dengan aksara &. Nama pengarang patut dalam tertib Nama Pertama Nama Akhir.

--book-producer, -k
Tetapkan pengeluar buku.

--category
Tetapkan kategori buku.

--comments, -c
Tetapkan keterangan e-buku.

--cover
Tetapkan kulit hadapan ke fail yang dinyatakan.

--date, -d
Tetapkan tarikh diterbitkan.

--from-opf
Baca data meta dari fail OPF yang dinyatakna dan guna ia untuk tetapkan data meta dalam e-buku. Data meta dinyatakan pada baris perintah akan membatalkan data meta yang dibaca dari fail OPF

--get-cover
Dapatkan kulit buku dari e-buku dan simpan ia seperti fail yang dinyatakan.

--help, -h
tunjuk mesej bantuan ini dan keluar

--identifier
Set the identifiers for the book, can be specified multiple times. For example: *--identifier* (halaman 298) uri:https://acme.com *--identifier* (halaman 298) isbn:12345 To remove an identifier, specify no value, *--identifier* (halaman 298) isbn: Note that for EPUB files, an identifier marked as the package identifier cannot be removed.

--index, -i
Tetapkan indeks buku dalam siri ini.

--isbn
Tetapkan ISBN buku.

--language, -l
Tetapkan bahasa.

--lrf-bookid
Tetapkan ID buku dalam fail LRF

--publisher, -p
Tetapkan penerbit e-buku.

--rating, -r
Tetapkan penarafan. Seharusnya nombor diantara 1 hingga 5.

--series, -s
Tetapkan siri e-buku ini berasal.

--tags
Tetapkan tag untuk buku. Seharusnya senarai dipisah dengan tanda koma.

--title, -t
Tetapkan tajuk.

--title-sort
Versi tajuk yang digunakan untuk mengisih. Jika tidak dinyatakan, dan tajuk dinyatakan, ia akan auto-jana dari tajuk.

--to-opf
Nyatakan nama bagi fail OPF. Data meta akan ditulis ke fail OPF.

--version
tunjuk nombor versi program dan keluar

13.1.10 ebook-polish

```
ebook-polish [pilihan] fail_input [fail_output]
```

Menggilap buku ialah menjadikan e-buku yang dibina lebih terserlah

Penggilapan cuba meminimumkan perubahan kod dalaman e-buku anda. Tidak seperti pertukaran, ia tidak meratakan CSS, namakan semula fail, ubah saiz fon, laras margins, dll. Setiap tindakan yang dilakukan hanya membuat perubahan minimum yang diperlukan untuk kesan yang dikehendaki.

Anda patut guna alat ini kerana merupakan langkah terakhir dalam penciptaan e-buku anda.

Perhatian, penggilapan hanya berfungsi pada faill dalam format AZW3atauEPUB.

Whenever you pass arguments to **ebook-polish** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--add-soft-hyphens, -H
Add soft hyphens to all words in the book. This allows the book to be rendered better when the text is justified, in readers that do not support hyphenation.

--compress-images, -i
Mampat imej secara tak hilang dalam buku, untuk mengurangkan saiz fail, tanpa mempengaruhi kualiti imej.

--cover, -c
Laluan ke imej kulit buku. Perubahan kulit buku dinyatakan dalam e-buku. Jika tiada kulit buku hadir, atau kulit buku tidak dikenalpasti dengan baik, sisipkan satu kulit buku baharu.

--embed-fonts, -e
Benamkan semua fon yang dirujuk dalam dokumen tetapi sebenarnya tidak dibenam. Ia akan mengimbas fon tersebut dalam komputer anda, dan jika temui, ia akan dibenamkan ke dalam dokumen. Sila pastikan anda mempunyai lesen yang sah untuk membenamkan fon tersebut untuk digunakan dalam dokumen ini.

--help, -h
tunjuk mesej bantuan ini dan keluar

--jacket, -j
Sisip laman "jaket buku" pada permulaan buku yang mengandungi semua data meta buku seperti tajuk, tag, pengarang, siri, ulasan dan, lain-lain. Mana-mana jaket buku terdahulu akan diganti.

--opf, -o
Laluan ke fail OPF. Data meta dalam buku dikemaskini dari fail OPF.

--remove-jacket
Buang halaman jaket buku yang disisip sebelum ini.

--remove-soft-hyphens
Remove soft hyphens from all text in the book.

--remove-unused-css, -u
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 parse them all.

--smarten-punctuation, -p
Tukar sengkang teks, elipsis, petikan, hyphen berbilang, dan lain-lain ke dalam setara yang betul secara tipografi. Perhatian algoritma kadangkala jana keputusan yang salah, terutamanya bila petikan tunggal pada permulaan pemendekan yang terlibat.

--subset-fonts, -f

Subset fon bermaksud mengurangkan fon terbenam yang hanya mengandungi aksara yang perlu sahaja digunakan dalam fon di dalam buku. Ia mengurangkan saiz fail fon (separuh saiz fail fon adalah biasa). Contohnya, jika buku guna fon khusus untuk pengepala maka subset akan mengurangkan fon yang hanya mengandungi aksara yang hadir dalam pengepala di dalam buku. Atau jika buku membenamkan versi tebal dan condong fon, tetapi teks tebal dan condong jarang digunakan, atau tiada langsung digunakan. Oleh itu, fon tebal dan condong boleh dikurangkan atau dibuang sepenuhnya. Kelemahan membuat subset fon ialah jika anda bercadang mahu menambah lagi teks ke dalam buku anda, teks yang baru ditambah mungkin tidak dapat dipapar sepenuhnya kerana fon telah disubset.

--upgrade-book, -U

Tatar struktur dalaman buku, jika boleh. Sebagai contoh, tatar buku EPUB 2 ke buku EPUB 3.

--verbose

Hasilkan lagi output berjela, berguna untuk penyahpepijatan.

--version

tunjuk nombor versi program dan keluar

13.1.11 ebook-viewer

```
ebook-viewer [options] file
```

View an e-book.

Whenever you pass arguments to **ebook-viewer** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--continue

Terus membaca pada buku dibuka sebelum ini

--detach

Tanggalkan dari terminal kawalan, jika ada (Linux sahaja)

--force-reload

Force reload of all opened books

--full-screen, --fullscreen, -f

If specified, the E-book viewer window will try to open full screen when started.

--help, -h

tunjuk mesej bantuan ini dan keluar

--open-at

The position at which to open the specified book. The position is a location or position you can get by using the Go to->Location action in the viewer controls. Alternately, you can use the form toc:something and it will open at the location of the first Table of Contents entry that contains the string "something". The form toc-href:something will match the href (internal link destination) of toc nodes. The matching is exact, If you want to match a substring, use the form toc-href-contains:something. The form ref:something will use Reference mode references.

--raise-window

If specified, the E-book viewer window will try to come to the front when started.

--version

tunjuk nombor versi program dan keluar

13.1.12 `fetch-ebook-metadata`

```
fetch-ebook-metadata [options]
```

Dapatkan data meta buku dari sumber atas-talian. Anda mesti nyatakan sekurang-kurangnya tajuk, pengarang atau ISBN.

Whenever you pass arguments to **`fetch-ebook-metadata`** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

`--allowed-plugin, -p`
Nyatakan nama pemalam muat turun data meta yang digunakan. Secara lalai, semua pemalam data meta akan digunakan. Boleh dinyatakan berbilang kali untuk pemalam berbilang. Semua nama pemalam: Google, Google Images, Amazon.com, Edelweiss, Open Library, Overdrive, Big Book Search

`--authors, -a`
Pengarang buku

`--cover, -c`
Nyatakan satu nama fail. Kulit buku, jika ada, akan disimpan ke dalamnya. Tanpa pilihan ini, tiada kulit buku akan dimuat turun.

`--help, -h`
tunjuk mesej bantuan ini dan keluar

`--identifier, -I`
Identifiers such as ASIN/goodreads id etc. Can be specified multiple times for multiple identifiers. For example: *`--identifier`* (halaman 301) asin:B0082BAJA0

`--isbn, -i`
ISBN buku

`--opf, -o`
Outputkan data meta dalam format OPF selain dari teks boleh dibaca manusia.

`--timeout, -d`
Had masa tamat dalam saat. Lalai ialah 30

`--title, -t`
Tajuk buku

`--verbose, -v`
Cetak log ke konsol (stderr)

`--version`
tunjuk nombor versi program dan keluar

13.1.13 lrf2lrs

```
lrf2lrs book.lrf
```

Tukar fail LRF ke fail LRS (XML terenkod UTF-8)

Whenever you pass arguments to **lrf2lrs** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--dont-output-resources
Tidak simpan imej terbenam dan fail fon ke cakera

--help, -h
tunjuk mesej bantuan ini dan keluar

--output, -o
Fail LRS output

--verbose
Jadi lebih berjela

--version
tunjuk nombor versi program dan keluar

13.1.14 lrfviewer

```
lrfviewer [options] buku.lrf
```

Baca e-buku LRF buku.lrf

Whenever you pass arguments to **lrfviewer** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--disable-hyphenation
Lumpuhkan sengkang. Seharusnya lebih laju ketika penerapan.

--help, -h
tunjuk mesej bantuan ini dan keluar

--profile
Profil penerap LRF

--verbose
Cetak lagi maklumat mengenai proses penerapan

--version
tunjuk nombor versi program dan keluar

--visual-debug
Hidupkan bantuan visual pada penyahpepijatan enjin penerapan

--white-background
Secara lalai latar belakang tidak putih kerana saya rasa ia lebih mudah dibaca. Guna pilihan ini untuk menjadikan latar belakang putih.

13.1.15 lrs2lrf

```
lrs2lrf [options] file.lrs
```

Kompil fail LRS ke dalam fail LRF.

Whenever you pass arguments to **lrs2lrf** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--help, -h
tunjuk mesej bantuan ini dan keluar

--lrs
Tukar LRS ke LRS, berguna untuk penyahpepijatan.

--output, -o
Laluan ke fail output

--verbose
Pemprosesan berjela

--version
tunjuk nombor versi program dan keluar

13.1.16 web2disk

```
web2disk URL
```

Yang mana URL seperti https://google.com

Whenever you pass arguments to **web2disk** that have spaces in them, enclose the arguments in quotation marks. For example: "/some path/with spaces"

[pilihan]

--base-dir, -d
Direktori dasar yang mana URL disimpan. Lalai ialah .

--delay
Sela minimum dalam saat diantara dapatan berturutan. Lalai ialah 0 s

--dont-download-stylesheets
Jangan muat turun lembar gaya CSS.

--encoding
Pengekodan aksara untuk laman sesawang yang anda cuba muat turun. Lalai ialah cuba dan teks pengekodan.

--filter-regexp
Mana-mana pautan yang sepadan dengan ungkapan nalar ini akan diabaikan. Pilihan ini boleh dinyatakan berbilang kali. Secara lalai, tiada pautan diabaikan. Jika kedua-dua ungkapan nalar dan padanan ungkapan nalar dinyatakan, maka ungkapan nalar padanan dilaksana dahulu.

--help, -h
tunjuk mesej bantuan ini dan keluar

--match-regexp
Hanya pautan yang padan dengan ungkapan nalar ini akan diikuti. Pilihan ini boleh dinyatakan berbilang kali. Secara lalai semua pautan diikuti.

--max-files, -n
Bilangan maksimum fail yang dimuat turun. Ia hanya dilaksana pada fail melalui tag <a href>. Lalai ialah 9223372036854775807

--max-recursions, -r
Bilangan aras maksimum untuk rekursif iaitu kedalaman pautan yang diikuti. Lalai 1

--timeout, -t
Had masa tamat dalam saat untuk menunggu respon dari pelayan. Lalai: 10.0 s

--verbose
Tunjuk maklumat output terperinci. Berguna untuk penyahpepijatan

--version
tunjuk nombor versi program dan keluar

13.2 Undocumented commands

- ebook-device
- markdown-calibre

Anda boleh lihat penggunaan perintah yang tidak didokumenkan dengan melakukannya tanpa argumen di dalam terminal.

CHAPTER 14

Menetapkan persekitaran pembangunan calibre

calibre is completely open source, licensed under the GNU GPL v3. 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 with some C/C++ code for speed and system interfacing. Note that calibre requires at least Python 3.8.

Kandungan

- *Design philosophy* (halaman 306)
  - *Bentangan kod* (halaman 306)
- *Mendapatkan kod* (halaman 307)
  - *Penyerahan perubahan yang anda buat untuk disertakan* (halaman 307)
- *Windows development environment* (halaman 308)
- *macOS development environment* (halaman 309)
- *Linux development environment* (halaman 309)
- *Having separate "normal" and "development" calibre installs on the same computer* (halaman 310)
- *Debugging tips* (halaman 310)
  - *Using print statements* (halaman 310)
  - *Using an interactive Python interpreter* (halaman 311)
  - *Using the Python debugger as a remote debugger* (halaman 311)
  - *Using the debugger in your favorite Python IDE* (halaman 311)
  - *Executing arbitrary scripts in the calibre Python environment* (halaman 312)
- *Menggunakan calibre dalam projek anda* (halaman 312)
  - *Pemasangan binari calibre* (halaman 312)

  - *Pemasangan sumber pada Linux* (halaman 312)
- *Dokumentasi API untuk pelbagai bahagian calibre* (halaman 313)

14.1 Design philosophy

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* (halaman 225) 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. Similarly, adding support for new conversion formats involves writing input/output format plugins. Another example of the modular design is the *recipe system* (halaman 27) for fetching news. For more examples of plugins designed to add features to calibre, see the Index of plugins.

14.1.1 Bentangan kod

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 *Pengenalan* (halaman 51). The pipeline consists of an input plugin, various transforms and an output plugin. The code that constructs and drives the pipeline is in `plumber.py`. The pipeline works on a representation of an e-book that is like an unzipped epub, with manifest, spine, toc, guide, html content, etc. The class that manages this representation is OEBBook in `ebooks.oeb.base`. The various transformations that are applied to the book during conversions live in `oeb/transforms/*.py`. And the input and output plugins live in `conversion/plugins/*.py`.
  - E-book editing happens using a different container object. It is documented in *API documentation for the e-book editing tools* (halaman 319).
- db - The database back-end. See *API documentation for the database interface* (halaman 313) 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`.

If you want to locate the entry points for all the various calibre executables, look at the `entry_points` structure in linux.py.

If you need help understanding the code, post in the development forum and you will most likely get help from one of calibre's many developers.

14.2 Mendapatkan kod

Anda boleh dapatkan kod sumber calibre dalam dua cara, iaitu menggunakan sistem kawalan versi atau memuat turun terus tarball.

calibre menggunakan Git, iaiatu satu sistem kawalan versi teragih. Git tersedia dalam semua platforms yang calibre sokong. Selepas memasang Git, anda boleh dapatkan kod sumber calibre dengan perintah:

```
git clone git://github.com/kovidgoyal/calibre.git
```

Pada Windows anda perlukan nama laluan yang lengkap, iaitu seperti ini `C:\Program Files\Git\git.exe`.

calibre merupakan projek yang berskala besar dengan sejarah kawalan sumber yang sangat lama, oleh ia mengambil masa (10 min hingga sejam bergantung pada kelajuan internet anda).

Jika anda mahu dapatkan kod dengan lebih pantas, kod sumber bagi keluaran terkini sentiasa tersedia dalam bentuk arkib.

Untuk mengemaskini cabang ke kod terkini, gunakan perintah:

```
git pull --no-edit
```

Anda juga boleh melayari kod di GitHub.

14.2.1 Penyerahan perubahan yang anda buat untuk disertakan

Jika anda bercadang untuk membuat beberapa perubahan kecil. Anda boleh membuat perubahan tersebut dengan mencipta satu "merge directive" kemudian anda boleh lampirkan tiket ke dalam penjejak pepijat calibre. Untuk membuatnya, buat perubahan tersebut, kemudian jalankan:

```
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 directory, simply attach that to a ticket on the calibre bug tracker. 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
```

Untuk menghantar *n* commit terakhir, gantikan *1* dengan *n*, sebagai contoh, bagi 3 commit terakhir:

```
git format-patch HEAD~3 --stdout > my-changes
```

Be careful to not include merges when using `HEAD~n`.

If you plan to do a lot of development on calibre, then the best method is to create a GitHub 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:

- Persediaan git dalam komputer anda sepertimana yang dijelaskan dalam artikel ini: Setup Git
- Persediaan kunci ssh untuk pengesahihan ke GitHub, sepertimana yang dijelaskan di sini: Generating SSH keys
- Pergi ke https://github.com/kovidgoyal/calibre dan klik butang *Fork*.

- Dalam Terminal taip:

  ```
  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.

- Anda boleh buat perubahan dan commit ia bilamana anda suka. Bila anda sudah sedia menggabungkan kerja anda, lakukan:

  ```
  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. Before making major changes, you should discuss them in the forum or contact Kovid directly (his email address is all over the source code).

14.3 Windows development environment

Note: You must also get the calibre source code separately as described above.

Install calibre normally, using the Windows installer. Then open a Command Prompt and change to the previously checked out calibre code directory. For example:

```
cd C:\Users\kovid\work\calibre
```

calibre is the directory that contains the src and resources sub-directories.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src directory. So, following the example above, it would be `C:\Users\kovid\work\calibre\src`. Here is a short guide to setting environment variables on Windows.

Once you have set the environment variable, open a new command prompt and check that it was correctly set by using the command:

```
echo %CALIBRE_DEVELOP_FROM%
```

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.

14.4 macOS development environment

Note: You must also get the calibre source code separately as described above.

Install calibre normally using the provided .dmg. Then open a Terminal and change to the previously checked out calibre code directory, for example:

```
cd /Users/kovid/work/calibre
```

calibre is the directory that contains the src and resources sub-directories. The calibre command line tools are found inside the calibre app bundle, in `/Applications/calibre.app/Contents/MacOS` you should add this directory 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 directory when running calibre in debug mode.

Create a plain text file:

```
#!/bin/sh
export CALIBRE_DEVELOP_FROM="/Users/kovid/work/calibre/src"
calibre-debug -g
```

Save this file as `/usr/bin/calibre-develop`, then set its permissions so that it can be executed:

```
chmod +x /usr/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.

14.5 Linux development environment

Note: 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. Then open a terminal and change to the previously checked out calibre code directory, for example:

```
cd /home/kovid/work/calibre
```

calibre is the directory that contains the src and resources sub-directories.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src directory. 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.

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 Having separate "normal" and "development" calibre installs on the same computer

The calibre source tree is very stable and rarely breaks, but if you feel the need to run from source on a separate test library and run the released calibre version with your everyday library, you can achieve this easily using .bat files or shell scripts to launch calibre. The example below shows how to do this on Windows using .bat files (the instructions for other platforms are the same, just use a shell script instead of a .bat file)

To launch the release version of calibre with your everyday library:

calibre-normal.bat:

```
calibre.exe "--with-library=C:\path\to\everyday\library folder"
```

calibre-dev.bat:

```
set CALIBRE_DEVELOP_FROM=C:\path\to\calibre\checkout\src
calibre.exe "--with-library=C:\path\to\test\library folder"
```

14.7 Debugging tips

Python is a dynamically typed language with excellent facilities for introspection. Kovid wrote the core calibre code without once using a debugger. There are many strategies to debug calibre code:

14.7.1 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.

Note: 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)`.

Note: 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 directory 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` subdirectory 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 Menggunakan calibre dalam projek anda

Boleh menggunakan kod/fungsi calibre terus ke dalam projek Python anda. Terdapat dua cara yang boleh dilakukan:

14.8.1 Pemasangan binari calibre

Jika anda pemasangan binari calibre, anda boleh gunakan pentafsir Python yang diserta sekali di dalam calibre, seperti berikut:

```
calibre-debug /path/to/your/python/script.py -- arguments to your script
```

14.8.2 Pemasangan sumber pada Linux

Selain dari teknik di atas, jika anda mempunyai pemasangan sumber dalam Linux, anda juga boleh mengimport terus calibre, seperti berikut:

```
import init_calibre
import calibre

print calibre.__version__
```

Perlu ingat anda perlu mengimport modul init_calibre sebelum lain-lain modul/pakej calibre menetapkan pentafsir untuk jalankan kod calibre.

14.9 Dokumentasi API untuk pelbagai bahagian calibre

14.9.1 API documentation for the database interface

This API is thread safe (it uses a multiple reader, single writer locking scheme). You can access this API like this:

```
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

If you are in a calibre plugin that is part of the main calibre GUI, you get access to it like this instead:

```
db = self.gui.current_db.new_api
```

class `calibre.db.cache.`**`Cache`**(*backend*)

An in-memory cache of the metadata.db file from a calibre library. This class also serves as a threadsafe API for accessing the database. The in-memory cache is maintained in normal form for maximum performance.

SQLITE is simply used as a way to read and write from metadata.db robustly. All table reading/sorting/searching/caching logic is re-implemented. This was necessary for maximum performance and flexibility.

`add_books`(*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)

Add the specified books to the library. Books should be an iterable of 2-tuples, each 2-tuple of the form `(mi, format_map)` where mi is a Metadata object and format_map is a dictionary of the form `{fmt : path_or_stream}`, for example: `{'EPUB': '/path/to/file.epub'}`.

Returns a pair of lists: `ids, duplicates`. `ids` contains the book ids for all newly created books in the database. `duplicates` contains the `(mi, format_map)` for all books that already exist in the database as per the simple duplicate detection heuristic used by *`has_book()`* (halaman 316).

`add_custom_book_data`(*name*, *val_map*, *delete_first=False*)

Add data for name where val_map is a map of book_ids to values. If delete_first is True, all previously stored data for name will be removed.

`add_format`(*book_id*, *fmt*, *stream_or_path*, *replace=True*, *run_hooks=True*, *dbapi=None*)

Add a format to the specified book. Return True if the format was added successfully.

Parameters

- **`replace`** -- If True replace existing format, otherwise if the format already exists, return False.
- **`run_hooks`** -- If True, file type plugins are run on the format before and after being added.
- **`dbapi`** -- Internal use only.

`all_book_ids`(*type=<class 'frozenset'>*)

Frozen set of all known book ids.

`all_field_for`(*field*, *book_ids*, *default_value=None*)

Same as field_for, except that it operates on multiple books at once

`all_field_ids`(*name*)

Frozen set of ids for all values in the field `name`.

`all_field_names`(*field*)

Frozen set of all fields names (should only be used for many-one and many-many fields)

author_data(*author_ids=None*)

Return author data as a dictionary with keys: name, sort, link

If no authors with the specified ids are found an empty dictionary is returned. If author_ids is None, data for all authors is returned.

author_sort_from_authors(*authors*, *key_func=<function lower>*)

Given a list of authors, return the author_sort string for the authors, preferring the author sort associated with the author over the computed string.

books_for_field(*name*, *item_id*)

Return all the books associated with the item identified by `item_id`, where the item belongs to the field `name`.

Returned value is a set of book ids, or the empty set if the item or the field does not exist.

books_in_virtual_library(*vl*, *search_restriction=None*)

Return the set of books in the specified virtual library

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*)

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.

Parameters

- **as_file** -- If True return the image as an open file object (a SpooledTemporaryFile)
- **as_image** -- If True return the image as a QImage object
- **as_path** -- If True return the image as a path pointing to a temporary file

data_for_find_identical_books()

Return data that can be used to implement *find_identical_books()* (halaman 315) 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()* (halaman 316). This can be used for an implementation of *has_book()* (halaman 316) in a worker process without access to the db.

delete_custom_book_data(*name*, *book_ids=()*)

Delete data for name. By default deletes all data, if you only want to delete data for some book ids, pass in a list of book ids.

embed_metadata(*book_ids*, *only_fmts=None*, *report_error=None*, *report_progress=None*)

Update metadata in all formats of the specified book_ids to current metadata in the database.

fast_field_for(*field_obj*, *book_id*, *default_value=None*)

Same as field_for, except that it avoids the extra lookup to get the field object

field_for (*name*, *book_id*, *default_value=None*)

Return the value of the field `name` for the book identified by `book_id`. If no such book exists or it has no defined value for the field `name` or no such field exists, then `default_value` is returned.

`default_value` is not used for title, title_sort, authors, author_sort and series_index. This is because these always have values in the db. `default_value` is used for all custom columns.

The returned value for is_multiple fields are always tuples, even when no values are found (in other words, default_value is ignored). The exception is identifiers for which the returned value is always a dict. 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()`* (halaman 314).

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.

Parameters

- **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()`* (halaman 314) 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()`* (halaman 314).

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()`* (halaman 314) method.

Parameters

- **allow_cache** -- If `True` cached values are used, otherwise a slow filesystem access is done. The cache values could be out of date if access was performed to the filesystem outside of this API.
- **update_db** -- If `True` The max_size field of the database is updated for this book.

formats (*book_id*, *verify_formats=True*)
Return tuple of all formats for the specified book. If verify_formats is True, verifies that the files exist on disk.

get_categories (*sort='name'*, *book_ids=None*, *already_fixed=None*, *first_letter_sort=False*)
Used internally to implement the Tag Browser

get_custom_book_data (*name*, *book_ids=()*, *default=None*)
Get data for name. By default returns data for all book_ids, pass in a list of book ids if you only want some data. Returns a map of book_id to values. If a particular value could not be decoded, uses default for it.

get_id_map (*field*)
Return a mapping of id numbers to values for the specified field. The field must be a many-one or many-many field, otherwise a ValueError is raised.

get_ids_for_custom_book_data (*name*)
Return the set of book ids for which name has data.

get_item_id (*field*, *item_name*)
Return the item id for item_name (case-insensitive)

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()` (halaman 316).

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` (halaman 180) 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.

Parameters

- **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()` (halaman 316) 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()` (halaman 314).

has_format (*book_id*, *fmt*)
Return True iff the format exists on disk

has_id(*book_id*)

Return True iff the specified book_id exists in the db

init()

Initialize this cache with data from the backend.

multisort(*fields*, *ids_to_sort=None*, *virtual_fields=None*)

Return a list of sorted book ids. If ids_to_sort is None, all book ids are returned.

fields must be a list of 2-tuples of the form (field_name, ascending=True or False). The most significant field is the first 2-tuple.

pref(*name*, *default=None*, *namespace=None*)

Return the value for the specified preference or the value specified as `default` if the preference is not set.

read_backup(*book_id*)

Return the OPF metadata backup for the book as a bytestring or None if no such backup exists.

remove_books(*book_ids*, *permanent=False*)

Remove 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 recycle bin.

remove_formats(*formats_map*, *db_only=False*)

Remove the specified formats from the specified books.

Parameters

- **formats_map** -- A mapping of book_id to a list of formats to be removed from the book.
- **db_only** -- If True, only remove the record for the format from the db, do not delete the actual format file from the filesystem.

remove_items(*field*, *item_ids*, *restrict_to_book_ids=None*)

Delete all items in the specified field with the specified ids. Returns the set of affected book ids. `restrict_to_book_ids` is an optional set of books ids. If specified the items will only be removed from those books.

rename_items(*field*, *item_id_to_new_name_map*, *change_index=True*, *restrict_to_book_ids=None*)

Rename items from a many-one or many-many field such as tags or series.

Parameters

- **change_index** -- When renaming in a series-like field also change the series_index values.
- **restrict_to_book_ids** -- An optional set of book ids for which the rename is to be performed, defaults to all books.

restore_book(*book_id*, *mi*, *last_modified*, *path*, *formats*, *annotations=()*)

Restore the book entry in the database for a book that already exists on the filesystem

restore_original_format(*book_id*, *original_fmt*)

Restore the specified format from the previously saved ORIGINAL_FORMAT, if any. Return True on success. The ORIGINAL_FORMAT is deleted after a successful restore.

property safe_read_lock

A safe read lock is a lock that does nothing if the thread already has a write lock, otherwise it acquires a read lock. This is necessary to prevent DowngradeLockErrors, which can happen when updating the search cache in the presence of composite columns. Updating the search cache holds an exclusive lock, but searching a composite column involves reading field values via ProxyMetadata which tries to get a shared lock. There may be other scenarios that trigger this as well.

This property returns a new lock object on every access. This lock object is not recursive (for performance) and must only be used in a with statement as `with cache.safe_read_lock:` otherwise bad things will happen.

`save_original_format`(*book_id*, *fmt*)
Save a copy of the specified format as ORIGINAL_FORMAT, overwriting any existing ORIGINAL_FORMAT.

`search`(*query*, *restriction=''*, *virtual_fields=None*, *book_ids=None*)
Search the database for the specified query, returning a set of matched book ids.

Parameters

- **`restriction`** -- A restriction that is ANDed to the specified query. Note that restrictions are cached, therefore the search for a AND b will be slower than a with restriction b.
- **`virtual_fields`** -- Used internally (virtual fields such as on_device to search over).
- **`book_ids`** -- If not None, a set of book ids for which books will be searched instead of searching all books.

`set_conversion_options`(*options*, *fmt='PIPE'*)
options must be a map of the form {book_id:conversion_options}

`set_cover`(*book_id_data_map*)
Set the cover for this book. data can be either a QImage, QPixmap, file object or bytestring. It can also be None, in which case any existing cover is removed.

`set_field`(*name*, *book_id_to_val_map*, *allow_case_change=True*, *do_path_update=True*)
Set the values of the field specified by `name`. Returns the set of all book ids that were affected by the change.

Parameters

- **`book_id_to_val_map`** -- Mapping of book_ids to values that should be applied.
- **`allow_case_change`** -- If True, the case of many-one or many-many fields will be changed. For example, if a book has the tag `tag1` and you set the tag for another book to `Tag1` then the both books will have the tag `Tag1` if allow_case_change is True, otherwise they will both have the tag `tag1`.
- **`do_path_update`** -- Used internally, you should never change it.

`set_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()` (halaman 317).

`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.

Parameters

- **`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 API documentation for the e-book editing tools

The e-book editing tools consist of a *calibre.ebooks.oeb.polish.container.Container* (halaman 319) object that represents a book as a collection of HTML + resource files, and various tools that can be used to perform operations on the container. All the tools are in the form of module level functions in the various `calibre.ebooks.oeb.polish.*` modules.

You obtain a container object for a book at a path like this:

```
from calibre.ebooks.oeb.polish.container import get_container
container = get_container('Path to book file', tweak_mode=True)
```

If you are writing a plugin for the E-book editor, you get the current container for the book being edited like this:

```
from calibre.gui2.tweak_book import current_container
container = current_container()
if container is None:
    report_error # No book has been opened yet
```

The Container object

class calibre.ebooks.oeb.polish.container.**Container**(*rootpath*, *opfpath*, *log*, *clone_data=None*)

A container represents an Open E-Book as a directory full of files and an opf file. There are two important concepts:

- The root directory. This is the base of the e-book. All the e-books files are inside this directory or in its sub-directories.
- Names: These are paths to the books' files relative to the root directory. 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()* (halaman 322) or *open()* (halaman 321) 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`

Parameters root -- The base directory. 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'

The type of book (epub for EPUB files and azw3 for AZW3 files)

commit(*outpath=None*, *keep_parsed=False*)

Commit all dirtied parsed objects to the filesystem and write out the e-book file at outpath.

Parameters

- **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()` (halaman 322)

dirty(*name*)

Mark the parsed object corresponding to name as dirty. See also: `parsed()` (halaman 322).

exists(*name*)

True iff a file/directory 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()` (halaman 320)

filesize(*name*)

Return the size in bytes of the file represented by the specified canonical name. Automatically handles dirtied parsed objects. See also: `parsed()` (halaman 322)

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()` (halaman 320) this method is always case-sensitive.

href_to_name(*href*, *base=None*)

Convert an href (relative to base) to a name. base must be a name or None, in which case self.root is used.

insert_into_xml(*parent*, *item*, *index=None*)
Insert item into parent (or append if index is None), fixing indentation. Only works with self closing items.

is_dir = False
If this container represents an unzipped book (a directory)

iterlinks(*name*, *get_line_numbers=True*)
Iterate over all links in name. If get_line_numbers is True the yields results of the form (link, line_number, offset). Where line_number is the line_number at which the link occurs and offset is the number of characters from the start of the line. Note that offset could actually encompass several lines if not zero.

make_name_unique(*name*)
Ensure that *name* does not already exist in this book. If it does, return a modified version that does not exist.

manifest_has_name(*name*)
Return True if the manifest has an entry corresponding to name

property manifest_id_map
Mapping of manifest id to canonical names

manifest_items_of_type(*predicate*)
The names of all manifest items whose media-type matches predicate. *predicate* can be a set, a list, a string or a function taking a single argument, which will be called with the media-type.

manifest_items_with_property(*property_name*)
All manifest items that have the specified property

property manifest_type_map
Mapping of manifest media-type to list of canonical names of that media-type

property mi
The metadata of this book as a Metadata object. Note that this object is constructed on the fly every time this property is requested, so use it sparingly.

name_to_abspath(*name*)
Convert a canonical name to an absolute OS dependant 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
The parsed OPF file

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()` (halaman 320) so that the container knows to update the cache. See also `replace()` (halaman 322).

raw_data(*name*, *decode=True*, *normalize_to_nfc=True*)
Return the raw data corresponding to the file specified by name

Parameters

- **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()` (halaman 319) 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 directory 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()` (halaman 322).

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* (halaman 323) and *spine_items* (halaman 323).

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* (halaman 323) and *spine_items* (halaman 323).

property spine_names
An iterator yielding name and is_linear for every item in the books' spine. See also: *spine_iter* (halaman 323) and *spine_items* (halaman 323).

Managing component files in a container

calibre.ebooks.oeb.polish.replace.**replace_links**(*container*, *link_map*, *frag_map=<function <lambda>>*, *replace_in_opf=False*)
Replace links to files in the container. Will iterate over all files in the container and change the specified links in them.

Parameters

- **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.

Parameters **file_map** -- A mapping of old canonical name to new canonical name, for example: `{'text/chapter1.html': 'chapter1.html'}`.

calibre.ebooks.oeb.polish.replace.**get_recommended_folders**(*container*, *names*)
Return the folders that are recommended for the given filenames. The recommendation is based on where the majority of files of the same type are located in the container. If no files of a particular type are present, the recommended folder is assumed to be the folder containing the OPF file.

Pretty printing and auto fixing parse errors

calibre.ebooks.oeb.polish.pretty.**fix_html**(*container*, *raw*)
Fix any parsing errors in the HTML represented as a string in raw. Fixing is done using the HTML5 parsing algorithm.

calibre.ebooks.oeb.polish.pretty.**fix_all_html**(*container*)
Fix any parsing errors in all HTML files in the container. Fixing is done using the HTML5 parsing algorithm.

calibre.ebooks.oeb.polish.pretty.**pretty_html**(*container*, *name*, *raw*)
Pretty print the HTML represented as a string in raw

calibre.ebooks.oeb.polish.pretty.**pretty_css**(*container*, *name*, *raw*)

Pretty print the CSS represented as a string in raw

calibre.ebooks.oeb.polish.pretty.**pretty_xml**(*container*, *name*, *raw*)

Pretty print the XML represented as a string in raw. If `name` is the name of the OPF, extra OPF-specific prettying is performed.

calibre.ebooks.oeb.polish.pretty.**pretty_all**(*container*)

Pretty print all HTML/CSS/XML files in the container

Managing book jackets

calibre.ebooks.oeb.polish.jacket.**remove_jacket**(*container*)

Remove an existing jacket, if any. Returns False if no existing jacket was found.

calibre.ebooks.oeb.polish.jacket.**add_or_replace_jacket**(*container*)

Either create a new jacket from the book's metadata or replace an existing jacket. Returns True if an existing jacket was replaced.

Splitting and merging of files

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.

Parameters

- **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** -- Used internally

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()* (halaman 324). Splitting automatically migrates all links and references to the affected files.

Parameters **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.

Parameters

- **category** -- Must be either `'text'` for HTML files or `'styles'` for CSS files
- **names** -- The list of files to be merged
- **master** -- Which of the merged files is the *master* file, that is, the file that will remain after merging.

Managing covers

`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.

Parameters

- **`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.

Parameters **`move_to_start`** -- If True the HTML file is moved to the start of the spine

Working with 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 all unused CSS rules from the book. An unused CSS rule is one that does not match any actual content.

Parameters

- **`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.

`calibre.ebooks.oeb.polish.css.`**`filter_css`**`(`*container*, *properties*, *names=()*`)`

Remove the specified CSS properties from all CSS rules in the book.

Parameters

- **`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.

Working with the Table of Contents

`calibre.ebooks.oeb.polish.toc.`**`from_xpaths`**(*container*, *xpaths*)

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.

Parameters **`title`** -- The title for this table of contents.

Sunting alat buku

`class` `calibre.gui2.tweak_book.plugin.`**`Tool`**

Bases: `object`

The base class for individual tools in an Edit Book plugin. Useful members include:

- `self.plugin`: A reference to the *`calibre.customize.Plugin`* (halaman 226) object to which this tool belongs.
- self. *`boss`* (halaman 326)
- self. *`gui`* (halaman 326)

Methods that must be overridden in sub classes:

- *`create_action()`* (halaman 327)
- *`register_shortcut()`* (halaman 326)

`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`* (halaman 327) object. Used to control the user interface.

`property gui`

The main window of the user interface

`property current_container`

Return the current *`calibre.ebooks.oeb.polish.container.Container`* (halaman 319) 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 section of the editor preferences.

Parameters

- **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:

```python
def create_action(self, for_toolbar=True):
    ac = QAction(get_icons('myicon.png'), 'Do something')
    if for_toolbar:
        # We want the toolbar button to have a popup menu
        menu = QMenu()
        ac.setMenu(menu)
        menu.addAction('Do something else')
        subaction = menu.addAction('And another')

        # Register a keyboard shortcut for this toolbar action be
        # careful to do this for only one of the toolbar action or
        # the menu action, not both.
        self.register_shortcut(ac, 'some-unique-name', default_keys=('Ctrl+K',
↪))
    return ac
```

See also:

Method *register_shortcut()* (halaman 326).

Controlling the editor's user interface

The e-book editor's user interface is controlled by a single global *Boss* object. This has many useful methods that can be used in plugin code to perform common tasks.

class calibre.gui2.tweak_book.boss.**Boss** (*parent*, *notify=None*)

add_savepoint (*msg*)

Create a restore checkpoint with the name specified as `msg`

apply_container_update_to_gui(*mark_as_modified=True*)
Update all the components of the user interface to reflect the latest data in the current book container.

Parameters **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

Parameters

- **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*)
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.

Parameters **edit_file** -- The name of a file inside the newly opened book to start editing. Can also be a list of names.

rewind_savepoint()
Undo the previous creation of a restore checkpoint, useful if you create a checkpoint, then abort the operation with no changes

save_book()
Save the book. Saving is performed in the background

set_modified()
Mark the book as having been modified

show_current_diff(*allow_revert=True*, *to_container=None*)
Show the changes to the book from its last checkpointed state

Parameters

- **allow_revert** -- If True the diff dialog will have a button to allow the user to revert all changes
- **to_container** -- A container object to compare the current container to. If None, the previously checkpointed container is used

show_editor(*name*)
Show the editor that is editing the file specified by `name`

sync_preview_to_editor()
Sync the position of the preview panel to the current cursor position in the current editor

CHAPTER 15

Digital Rights Management (DRM)

Digital rights management (DRM) is a generic term for access control technologies that can be used by hardware manufacturers, publishers, copyright holders and individuals to try to impose limitations on the usage of digital content and devices. It is also, sometimes, disparagingly described as Digital Restrictions Management. The term is used to describe any technology which inhibits uses (legitimate or otherwise) of digital content that were not desired or foreseen by the content provider. The term generally doesn't refer to other forms of copy protection which can be circumvented without modifying the file or device, such as serial numbers or key-files. It can also refer to restrictions associated with specific instances of digital works or devices. DRM technologies attempt to control use of digital media by preventing access, copying or conversion to other formats by end users. See wikipedia.

15.1 What does DRM imply for me personally?

When you buy an e-book with DRM you don't really own it but have purchased the permission to use it in a manner dictated to you by the seller. DRM limits what you can do with e-books you have "bought". Often people who buy books with DRM are unaware of the extent of these restrictions. These restrictions prevent you from reformatting the e-book to your liking, including making stylistic changes like adjusting the font sizes, although there is software that empowers you to do such things for non DRM books. People are often surprised that an e-book they have bought in a particular format cannot be converted to another format if the e-book has DRM. So if you have an Amazon Kindle and buy a book sold by Barnes and Nobles, you should know that if that e-book has DRM you will not be able to read it on your Kindle. Notice that I am talking about a book you buy, not steal or pirate but BUY.

15.2 What does DRM do for authors?

Publishers of DRMed e-books argue that the DRM is all for the sake of authors and to protect their artistic integrity and prevent piracy. But DRM does NOT prevent piracy. People who want to pirate content or use pirated content still do it and succeed. The three major DRM schemes for e-books today are run by Amazon, Adobe and Barnes and Noble and all three DRM schemes have been cracked. All DRM does is inconvenience legitimate users. It can be argued that it actually harms authors as people who would have bought the book choose to find a pirated version as they are not willing to put up with DRM. Those that would pirate in the absence of DRM do so in its presence as well. To reiterate, the key point is that DRM *does not prevent piracy*. So DRM is not only pointless and harmful to buyers of e-books but also a waste of money.

15.3 DRM and freedom

Although digital content can be used to make information as well as creative works easily available to everyone and empower humanity, this is not in the interests of some publishers who want to steer people away from this possibility of freedom simply to maintain their relevance in world developing so fast that they cant keep up.

15.4 Why does calibre not support DRM?

calibre is open source software while DRM by its very nature is closed. If calibre were to support opening or viewing DRM files it could be trivially modified to be used as a tool for DRM removal which is illegal under today's laws. Open source software and DRM are a clash of principles. While DRM is all about controlling the user open source software is about empowering the user. The two simply can not coexist.

15.5 What is calibre's view on content providers?

We firmly believe that authors and other content providers should be compensated for their efforts, but DRM is not the way to go about it. We are developing this database of DRM-free e-books from various sources to help you find DRM-free alternatives and to help independent authors and publishers of DRM-free e-books publicize their content. We hope you will find this useful and we request that you do not pirate the content made available to you here.

15.6 How can I help fight DRM?

As somebody who reads and buys e-books you can help fight DRM. Do not buy e-books with DRM. There are some publishers who publish DRM-free e-books. Make an effort to see if they carry the e-book you are looking for. If you like books by certain independent authors that sell DRM-free e-books and you can afford it make donations to them. This is money well spent as their e-books tend to be cheaper (there may be exceptions) than the ones you would buy from publishers of DRMed books and would probably work on all devices you own in the future saving you the cost of buying the e-book again. Do not discourage publishers and authors of DRM-free e-books by pirating their content. Content providers deserve compensation for their efforts. Do not punish them for trying to make your reading experience better by making available DRM-free e-books. In the long run this is detrimental to you. If you have bought books from sellers that carry both DRMed as well as DRM-free books, not knowing if they carry DRM or not make it a point to leave a comment or review on the website informing future buyers of its DRM status. Many sellers do not think it important to clearly indicate to their buyers if an e-book carries DRM or not. *Here <https://www.defectivebydesign.org/guide/ebooks>* you will find a Guide to DRM-free living.

CHAPTER 16

Glosari

RSS **RSS** *(Really Simple Syndication)* iaitu Pensindiketan Sangat Ringkas merupakan format suapan sesawang yang digunakan untuk menerbitkan kandungan dikemaskini secara kerap, seperti artikel berita, pos blog dan lain-lain. Ia adalah format yang biasanya sesuai dibaca menerusi komputers, dan cara terbaik mendapatkan kandungan daripada sesawang ke dalam e-buku. Terdapat pelbagai format suapan yang digunakan di Internet, dan calibre memahaminya. Dengan cara ini, ia mempunyai sokongan yang baik dengan format *ATOM*, yang biasanya digunakan untuk blog.

resepi Resepi adalah satu set arahan yang mengajar calibre bagaimana hendak tukarkan sumber berita atas talina, seperti majalah atau blog, menjadi e-buku. Resepi adalah dari kod Python. Oleh itu, ia berupaya menukar sumber berita arbitari yang kompleks kepada sebuah ebuku. Pada tahap paling mudah, ia hanyalah satu set pembolehubah, seperti URL, yang memberikan calibre maklumat yang mencukupi untuk pergi ke Internet dan muat turun berita yang dikehendaki.

HTML **HTML** *(Hyper Text Mark-Up Language)*, atau Bahasa Penanda Teks Hiper iaitu subset bagi Bahasa Penanda Am Piawai (SGML) untuk penerbitan elektronik, merupakan piawaian khusus yang digunakan untuk Dunia Internet iaitu World Wide Web.

CSS **CSS** *(Cascading Style Sheets)* atau Lembaran Gaya Lata merupakan bahasa yang digunakan untuk menjelaskan bagaimana dokumen *HTML* harus diterapkan (penggayaan visual).

API **API** *(Application Programming Interface)* iaitu Antaramuka Pengaturcaraan Aplikasi merupakan antaramuka kod sumber yang mana pustaka sediakan untuk menyokong permintaan perkhidmatan yang dibuat padanya oleh program komputer.

LRF **LRF** Format e-buku yang boleh dibaca oleh pembaca e-buku SONY.

URL **URL** *(Uniform Resource Locator)* atau Pelokasi Sumber Seragam contohnya: `http://contoh.com`

regexp **Regular expressions** provide a concise and flexible means for identifying strings of text of interest, such as particular characters, words, or patterns of characters. See regexp syntax for the syntax of regular expressions used in Python.

Python Module Index

c

calibre.customize, 225
calibre.customize.conversion, 234
calibre.db.cache, 313
calibre.devices.interface, 236
calibre.ebooks.metadata.book.base, 180
calibre.ebooks.metadata.sources.base, 230
calibre.ebooks.oeb.polish.container, 319
calibre.ebooks.oeb.polish.cover, 325
calibre.ebooks.oeb.polish.css, 325
calibre.ebooks.oeb.polish.jacket, 324
calibre.ebooks.oeb.polish.pretty, 323
calibre.ebooks.oeb.polish.replace, 323
calibre.ebooks.oeb.polish.split, 324
calibre.ebooks.oeb.polish.toc, 326
calibre.gui2.tweak_book.boss, 327
calibre.utils.formatter_functions, 165
calibre.web.feeds.news, 36

Index

Symbols

-1
  calibredb-add command line option, 279

-H
  ebook-polish command line option, 299

-I
  calibredb-add command line option, 278
  fetch-ebook-metadata command line option, 301

-S
  calibredb-add command line option, 278

-T
  calibredb-add command line option, 278

-U
  ebook-polish command line option, 300

--access-log
  calibre-server command line option, 271

--add
  calibredb-add command line option, 279

--add-plugin
  calibre-customize command line option, 268

--add-simple-plugin
  calibre-debug command line option, 269

--add-soft-hyphens
  ebook-polish command line option, 299

--ajax-timeout
  calibre-server command line option, 271

--all
  calibredb-backup_metadata command line option, 287
  calibredb-export command line option, 281

--allowed-plugin
  fetch-ebook-metadata command line option, 301

--append
  calibredb-set_custom command line option, 285

--as-opf
  calibredb-show_metadata command line option, 280

--ascending
  calibredb-list command line option, 277

--asciiize
  ebook-convert command line option, 290

--attachment
  calibre-smtp command line option, 275

--auth-mode
  calibre-server command line option, 271

--author-sort
  ebook-convert command line option, 295
  ebook-meta command line option, 297

--authors
  calibredb-add command line option, 278
  ebook-convert command line option, 295
  ebook-meta command line option, 297
  fetch-ebook-metadata command line option, 301

--auto-reload
  calibre-server command line option,

271
--automerge
calibredb-add command line option, 278
--ban-after
calibre-server command line option, 271
--ban-for
calibre-server command line option, 271
--base-dir
web2disk command line option, 303
--base-font-size
ebook-convert command line option, 290
--book-list-mode
calibre-server command line option, 271
--book-producer
ebook-convert command line option, 295
ebook-meta command line option, 297
--build-plugin
calibre-customize command line option, 268
--cafile
calibre-smtp command line option, 275
--catalog-title
calibredb-catalog command line option, 282
--categories
calibredb-list_categories command line option, 286
--category
ebook-meta command line option, 297
--change-justification
ebook-convert command line option, 290
--chapter
ebook-convert command line option, 294
--chapter-mark
ebook-convert command line option, 294
--command
calibre-debug command line option, 269
--comments
ebook-convert command line option, 295
ebook-meta command line option, 297
--compress-images
ebook-polish command line option, 299
--compress-min-size
calibre-server command line option, 271
--continue
ebook-viewer command line option, 300
--cover
calibredb-add command line option, 278
ebook-convert command line option, 295
ebook-meta command line option, 298
ebook-polish command line option, 299
fetch-ebook-metadata command line option, 301
--cross-reference-authors
calibredb-catalog command line option, 282
--csv
calibredb-check_library command line option, 286
calibredb-list_categories command line option, 286
--custom-list-template
calibre-server command line option, 271
--customize-plugin
calibre-customize command line option, 268
--daemonize
calibre-server command line option, 271
--date
ebook-meta command line option, 298
--debug-device-driver
calibre-debug command line option, 269
--debug-pipeline
calibredb-catalog command line option, 282
ebook-convert command line option, 296
--default-programs
calibre-debug command line option, 269
--delay
web2disk command line option, 303
--detach
calibre command line option, 268
ebook-edit command line option, 297
ebook-viewer command line option, 300

--details
    calibredb-custom_columns command line option, 284
--dialect
    calibredb-list_categories command line option, 286
--diff
    calibre-debug command line option, 269
--disable-allow-socket-preallocation
    calibre-server command line option, 271
--disable-auth
    calibre-server command line option, 272
--disable-dehyphenate
    ebook-convert command line option, 293
--disable-delete-blank-paragraphs
    ebook-convert command line option, 293
--disable-fallback-to-detected-interface
    calibre-server command line option, 272
--disable-fix-indents
    ebook-convert command line option, 293
--disable-font-rescaling
    ebook-convert command line option, 290
--disable-format-scene-breaks
    ebook-convert command line option, 293
--disable-hyphenation
    lrfviewer command line option, 302
--disable-italicize-common-cases
    ebook-convert command line option, 293
--disable-local-write
    calibre-server command line option, 272
--disable-log-not-found
    calibre-server command line option, 272
--disable-markup-chapter-headings
    ebook-convert command line option, 293
--disable-plugin
    calibre-customize command line option, 268
--disable-remove-fake-margins
    ebook-convert command line option, 294
--disable-renumber-headings
    ebook-convert command line option, 293
--disable-unwrap-lines
    ebook-convert command line option, 293
--disable-use-bonjour
    calibre-server command line option, 272
--disable-use-sendfile
    calibre-server command line option, 272
--display
    calibredb-add_custom_column command line option, 284
--displayed-fields
    calibre-server command line option, 271
--dont-asciiize
    calibredb-export command line option, 281
--dont-download-stylesheets
    web2disk command line option, 303
--dont-output-resources
    lrf2lrs command line option, 302
--dont-replace
    calibredb-add_format command line option, 279
--dont-save-cover
    calibredb-export command line option, 281
--dont-split-on-page-breaks
    ebook-convert command line option, 289
--dont-update-metadata
    calibredb-export command line option, 281
--dont-verify-server-certificate
    calibre-smtp command line option, 275
--dont-write-opf
    calibredb-export command line option, 281
--duplicate-links-in-toc
    ebook-convert command line option, 295
--duplicates
    calibredb-add command line option, 278
--edit-book
    calibre-debug command line option, 269
--embed-all-fonts
    ebook-convert command line option, 291

--embed-font-family
ebook-convert command line option, 291
--embed-fonts
ebook-polish command line option, 299
--empty
calibredb-add command line option, 278
--enable-allow-socket-preallocation
calibre-server command line option, 271
--enable-auth
calibre-server command line option, 272
--enable-fallback-to-detected-interface
calibre-server command line option, 272
--enable-heuristics
ebook-convert command line option, 293
--enable-local-write
calibre-server command line option, 272
--enable-log-not-found
calibre-server command line option, 272
--enable-plugin
calibre-customize command line option, 268
--enable-use-bonjour
calibre-server command line option, 272
--enable-use-sendfile
calibre-server command line option, 272
--encoding
web2disk command line option, 303
--encryption-method
calibre-smtp command line option, 275
--epub-flatten
ebook-convert command line option, 289
--epub-inline-toc
ebook-convert command line option, 289
--epub-toc-at-end
ebook-convert command line option, 289
--epub-version
ebook-convert command line option, 290
--exclude-genre
calibredb-catalog command line option, 282
--exclusion-rules
calibredb-catalog command line option, 282
--exec-file
calibre-debug command line option, 269
--expand-css
ebook-convert command line option, 291
--explode-book
calibre-debug command line option, 269
--export-all-calibre-data
calibre-debug command line option, 269
--extra-css
ebook-convert command line option, 291
--extract-to
ebook-convert command line option, 290
--field
calibredb-set_metadata command line option, 280
--fields
calibredb-list command line option, 277
--filter-css
ebook-convert command line option, 291
--filter-regexp
web2disk command line option, 303
--fix-multiprocessing
calibre-debug command line option, 270
--flow-size
ebook-convert command line option, 290
--font-size-mapping
ebook-convert command line option, 291
--for-machine
calibredb-list command line option, 277
--force
calibredb-remove_custom_column command line option, 285
--force-reload
ebook-viewer command line option, 300
--fork
calibre-smtp command line option,

274
--formats
calibredb-export command line option, 281
--from-opf
ebook-convert command line option, 296
ebook-meta command line option, 298
--full-screen
ebook-viewer command line option, 300
--fullscreen
ebook-viewer command line option, 300
--generate-authors
calibredb-catalog command line option, 282
--generate-descriptions
calibredb-catalog command line option, 282
--generate-genres
calibredb-catalog command line option, 282
--generate-recently-added
calibredb-catalog command line option, 283
--generate-series
calibredb-catalog command line option, 283
--generate-titles
calibredb-catalog command line option, 283
--genre-source-field
calibredb-catalog command line option, 283
--get-cover
ebook-meta command line option, 298
--gui
calibre-debug command line option, 270
--gui-debug
calibre-debug command line option, 270
--header-note-source-field
calibredb-catalog command line option, 283
--help
calibre command line option, 268
calibre-customize command line option, 268
calibre-debug command line option, 270
calibre-server command line option, 272
calibre-smtp command line option, 274
command line option, 277
ebook-convert command line option, 289
ebook-edit command line option, 297
ebook-meta command line option, 298
ebook-polish command line option, 299
ebook-viewer command line option, 300
fetch-ebook-metadata command line option, 301
lrf2lrs command line option, 302
lrfviewer command line option, 302
lrs2lrf command line option, 303
web2disk command line option, 304
--html-unwrap-factor
ebook-convert command line option, 293
--identifier
calibredb-add command line option, 278
ebook-meta command line option, 298
fetch-ebook-metadata command line option, 301
--ids
calibredb-catalog command line option, 282
--ignore
calibredb-add command line option, 279
--ignore_extensions
calibredb-check_library command line option, 286
--ignore_names
calibredb-check_library command line option, 286
--ignore-plugins
calibre command line option, 268
--ignored-fields
calibre-server command line option, 272
--implode-book
calibre-debug command line option, 270
--import-calibre-data
calibre-debug command line option, 270
--index
ebook-meta command line option, 298
--input-encoding
ebook-convert command line option, 289

--input-profile
ebook-convert command line option, 289
--insert-blank-line
ebook-convert command line option, 291
--insert-blank-line-size
ebook-convert command line option, 291
--insert-metadata
ebook-convert command line option, 294
--inspect-mobi
calibre-debug command line option, 270
--is-multiple
calibredb-add_custom_column command line option, 284
--isbn
calibredb-add command line option, 278
ebook-convert command line option, 295
ebook-meta command line option, 298
fetch-ebook-metadata command line option, 301
--item_count
calibredb-list_categories command line option, 286
--jacket
ebook-polish command line option, 299
--keep-ligatures
ebook-convert command line option, 291
--language
ebook-convert command line option, 296
ebook-meta command line option, 298
--languages
calibredb-add command line option, 278
--level1-toc
ebook-convert command line option, 295
--level2-toc
ebook-convert command line option, 295
--level3-toc
ebook-convert command line option, 295
--library-path
command line option, 277
--limit
calibredb-list command line option, 277
calibredb-search command line option, 288
--line-height
ebook-convert command line option, 291
--line-width
calibredb-list command line option, 277
--linearize-tables
ebook-convert command line option, 291
--list-fields
calibredb-set_metadata command line option, 280
--list-plugins
calibre-customize command line option, 268
--list-recipes
ebook-convert command line option, 289
--listen-on
calibre-server command line option, 272
--localhost
calibre-smtp command line option, 274
--log
calibre-server command line option, 272
--lrf-bookid
ebook-meta command line option, 298
--lrs
lrs2lrf command line option, 303
--manage-users
calibre-server command line option, 272
--margin-bottom
ebook-convert command line option, 291
--margin-left
ebook-convert command line option, 292
--margin-right
ebook-convert command line option, 292
--margin-top
ebook-convert command line option, 292
--match-regexp
web2disk command line option, 304
--max-files
web2disk command line option, 304

--max-header-line-size
calibre-server command line option, 272
--max-job-time
calibre-server command line option, 272
--max-jobs
calibre-server command line option, 273
--max-log-size
calibre-server command line option, 273
--max-opds-items
calibre-server command line option, 273
--max-opds-ungrouped-items
calibre-server command line option, 273
--max-recursions
web2disk command line option, 304
--max-request-body-size
calibre-server command line option, 273
--max-toc-links
ebook-convert command line option, 295
--merge-comments-rule
calibredb-catalog command line option, 283
--minimum-line-height
ebook-convert command line option, 292
--no-chapters-in-toc
ebook-convert command line option, 295
--no-default-epub-cover
ebook-convert command line option, 290
--no-svg-cover
ebook-convert command line option, 290
--no-update-check
calibre command line option, 268
--num-per-page
calibre-server command line option, 273
--one-book-per-directory
calibredb-add command line option, 279
--only-formats
calibredb-embed_metadata command line option, 287
--open-at
ebook-viewer command line option, 300
--opf
ebook-polish command line option, 299
fetch-ebook-metadata command line option, 301
--outbox
calibre-smtp command line option, 274
--output
lrf2lrs command line option, 302
lrs2lrf command line option, 303
--output-profile
calibredb-catalog command line option, 283
ebook-convert command line option, 289
--page-breaks-before
ebook-convert command line option, 294
--password
calibre-smtp command line option, 275
command line option, 277
--paths
calibre-debug command line option, 270
--permanent
calibredb-remove command line option, 279
--pidfile
calibre-server command line option, 273
--port
calibre-server command line option, 273
calibre-smtp command line option, 275
--prefer-metadata-cover
ebook-convert command line option, 294
--prefix
calibredb-list command line option, 277
--prefix-rules
calibredb-catalog command line option, 283
--preserve-cover-aspect-ratio
ebook-convert command line option, 290
--preset
calibredb-catalog command line option, 283
--pretty-print

ebook-convert command line option, 290
--profile
lrfviewer command line option, 302
--progress
calibredb-export command line option, 281
--pubdate
ebook-convert command line option, 296
--publisher
ebook-convert command line option, 296
ebook-meta command line option, 298
--raise-window
ebook-viewer command line option, 300
--rating
ebook-convert command line option, 296
ebook-meta command line option, 298
--read-metadata-from-opf
ebook-convert command line option, 296
--really-do-it
calibredb-restore_database command line option, 285
--recurse
calibredb-add command line option, 279
--reinitialize-db
calibre-debug command line option, 270
--relay
calibre-smtp command line option, 275
--remove-first-image
ebook-convert command line option, 294
--remove-jacket
ebook-polish command line option, 299
--remove-paragraph-spacing
ebook-convert command line option, 292
--remove-paragraph-spacing-indent-size
ebook-convert command line option, 292
--remove-plugin
calibre-customize command line option, 269
--remove-soft-hyphens
ebook-polish command line option, 299
--remove-unused-css
ebook-polish command line option, 299
--replace-scene-breaks
ebook-convert command line option, 293
--replace-whitespace
calibredb-export command line option, 281
--report
calibredb-check_library command line option, 286
--run-plugin
calibre-debug command line option, 270
--search
calibredb-catalog command line option, 282
calibredb-list command line option, 277
--search-replace
ebook-convert command line option, 293
--search-the-net-urls
calibre-server command line option, 273
--separator
calibredb-list command line option, 278
--series
calibredb-add command line option, 278
ebook-convert command line option, 296
ebook-meta command line option, 298
--series-index
calibredb-add command line option, 278
ebook-convert command line option, 296
--shutdown-running-calibre
calibre command line option, 268
calibre-debug command line option, 270
--shutdown-timeout
calibre-server command line option, 273
--single-dir
calibredb-export command line option, 281
--smarten-punctuation
ebook-convert command line option, 292
ebook-polish command line option,

299
--sort-by
calibredb-list command line option, 278
--sr1-replace
ebook-convert command line option, 293
--sr1-search
ebook-convert command line option, 293
--sr2-replace
ebook-convert command line option, 293
--sr2-search
ebook-convert command line option, 294
--sr3-replace
ebook-convert command line option, 294
--sr3-search
ebook-convert command line option, 294
--ssl-certfile
calibre-server command line option, 273
--ssl-keyfile
calibre-server command line option, 273
--start-in-tray
calibre command line option, 268
--start-reading-at
ebook-convert command line option, 294
--subject
calibre-smtp command line option, 275
--subset-embedded-fonts
ebook-convert command line option, 292
--subset-font
calibre-debug command line option, 270
--subset-fonts
ebook-polish command line option, 299
--tags
calibredb-add command line option, 278
ebook-convert command line option, 296
ebook-meta command line option, 298
--template
calibredb-export command line option, 281
--test-build
calibre-debug command line option, 270
--thumb-width
calibredb-catalog command line option, 283
--timefmt
calibredb-export command line option, 281
--timeout
calibre-server command line option, 273
calibre-smtp command line option, 274
fetch-ebook-metadata command line option, 301
web2disk command line option, 304
--timestamp
ebook-convert command line option, 296
--title
calibredb-add command line option, 278
ebook-convert command line option, 296
ebook-meta command line option, 298
fetch-ebook-metadata command line option, 301
--title-sort
ebook-convert command line option, 296
ebook-meta command line option, 298
--to-dir
calibredb-export command line option, 281
--to-lowercase
calibredb-export command line option, 281
--to-opf
ebook-meta command line option, 298
--toc-filter
ebook-convert command line option, 295
--toc-threshold
ebook-convert command line option, 295
--toc-title
ebook-convert command line option, 290
--transform-css-rules
ebook-convert command line option, 292
--trusted-ips
calibre-server command line option,

273

--unsmarten-punctuation
- ebook-convert command line option, 292

--upgrade-book
- ebook-polish command line option, 300

--url-prefix
- calibre-server command line option, 273

--use-auto-toc
- ebook-convert command line option, 295

--use-existing-cover
- calibredb-catalog command line option, 283

--userdb
- calibre-server command line option, 274

--username
- calibre-smtp command line option, 275
- command line option, 277

--verbose
- calibre command line option, 268
- calibre-smtp command line option, 274
- calibredb-catalog command line option, 282
- ebook-convert command line option, 296
- ebook-polish command line option, 300
- fetch-ebook-metadata command line option, 301
- lrf2lrs command line option, 302
- lrfviewer command line option, 302
- lrs2lrf command line option, 303
- web2disk command line option, 304

--version
- calibre command line option, 268
- calibre-customize command line option, 269
- calibre-debug command line option, 270
- calibre-server command line option, 274
- calibre-smtp command line option, 274
- command line option, 277
- ebook-convert command line option, 289
- ebook-edit command line option, 297
- ebook-meta command line option, 298
- ebook-polish command line option, 300
- ebook-viewer command line option, 300
- fetch-ebook-metadata command line option, 301
- lrf2lrs command line option, 302
- lrfviewer command line option, 302
- lrs2lrf command line option, 303
- web2disk command line option, 304

--viewer
- calibre-debug command line option, 270

--visual-debug
- lrfviewer command line option, 302

--white-background
- lrfviewer command line option, 302

--width
- calibredb-list_categories command line option, 286

--with-library
- calibre command line option, 268
- command line option, 277

--worker-count
- calibre-server command line option, 274

-a
- calibre-customize command line option, 268
- calibre-smtp command line option, 275
- calibredb-add command line option, 278
- calibredb-set_custom command line option, 285
- ebook-meta command line option, 297
- fetch-ebook-metadata command line option, 301

-b
- calibre-customize command line option, 268

-c
- calibre-debug command line option, 269
- calibredb-add command line option, 278
- calibredb-check_library command line option, 286
- calibredb-list_categories command line option, 286
- ebook-meta command line option, 297
- ebook-polish command line option, 299
- fetch-ebook-metadata command line

option, 301
-d
calibre-debug command line option, 269
calibredb-add command line option, 278
calibredb-custom_columns command line option, 284
ebook-convert command line option, 296
ebook-meta command line option, 298
fetch-ebook-metadata command line option, 301
web2disk command line option, 303
-e
calibre-debug command line option, 269
calibre-smtp command line option, 275
calibredb-add command line option, 278
calibredb-check_library command line option, 286
ebook-polish command line option, 299
-f
calibre-debug command line option, 270
calibre-smtp command line option, 274
calibredb-embed_metadata command line option, 287
calibredb-list command line option, 277
calibredb-remove_custom_column command line option, 285
calibredb-set_metadata command line option, 280
ebook-polish command line option, 299
ebook-viewer command line option, 300
-g
calibre-debug command line option, 270
-h
calibre command line option, 268
calibre-customize command line option, 268
calibre-debug command line option, 270
calibre-server command line option, 272
calibre-smtp command line option, 274
command line option, 277
ebook-convert command line option, 289
ebook-edit command line option, 297
ebook-meta command line option, 298
ebook-polish command line option, 299
ebook-viewer command line option, 300
fetch-ebook-metadata command line option, 301
lrf2lrs command line option, 302
lrfviewer command line option, 302
lrs2lrf command line option, 303
web2disk command line option, 304
-i
calibre-debug command line option, 270
calibredb-add command line option, 278
calibredb-catalog command line option, 282
calibredb-list_categories command line option, 286
ebook-meta command line option, 298
ebook-polish command line option, 299
fetch-ebook-metadata command line option, 301
-j
ebook-polish command line option, 299
-k
ebook-meta command line option, 297
-l
calibre-customize command line option, 268
calibre-smtp command line option, 274
calibredb-add command line option, 278
calibredb-search command line option, 288
calibredb-set_metadata command line option, 280
ebook-meta command line option, 298
-m
calibre-debug command line option, 270
calibredb-add command line option, 278
ebook-convert command line option, 296

-n
calibredb-check_library command line option, 286
web2disk command line option, 304
-o
calibre-smtp command line option, 274
ebook-polish command line option, 299
fetch-ebook-metadata command line option, 301
lrf2lrs command line option, 302
lrs2lrf command line option, 303
-p
calibre-smtp command line option, 275
ebook-meta command line option, 298
ebook-polish command line option, 299
fetch-ebook-metadata command line option, 301
-r
calibre-customize command line option, 269
calibre-debug command line option, 270
calibre-smtp command line option, 275
calibredb-add command line option, 279
calibredb-check_library command line option, 286
calibredb-list_categories command line option, 286
calibredb-restore_database command line option, 285
ebook-meta command line option, 298
web2disk command line option, 304
-s
calibre command line option, 268
calibre-debug command line option, 270
calibre-smtp command line option, 275
calibredb-add command line option, 278
calibredb-catalog command line option, 282
calibredb-list command line option, 277
ebook-meta command line option, 298
-t
calibre-debug command line option, 269
calibre-smtp command line option, 274
calibredb-add command line option, 278
ebook-meta command line option, 298
fetch-ebook-metadata command line option, 301
web2disk command line option, 304
-u
calibre-smtp command line option, 275
ebook-polish command line option, 299
-v
calibre command line option, 268
calibre-smtp command line option, 274
calibredb-catalog command line option, 282
ebook-convert command line option, 296
fetch-ebook-metadata command line option, 301
-w
calibre-debug command line option, 270
calibredb-list command line option, 277
calibredb-list_categories command line option, 286
-x
calibre-debug command line option, 269

A

abort_article() (*calibre.web.feeds.news.BasicNewsRecipe method*), 36
abort_recipe_processing() (*calibre.web.feeds.news.BasicNewsRecipe method*), 36
abspath_to_name() (*calibre.ebooks.oeb.polish.container.Container method*), 319
accept_drag_move_event() (*calibre.gui2.actions.InterfaceAction method*), 249
accept_enter_event() (*calibre.gui2.actions.InterfaceAction method*), 249
accepts_drops (*calibre.gui2.actions.InterfaceAction attribute*), 249
action_add_menu (*calibre.gui2.actions.InterfaceAction attribute*), 249
action_menu_clone_qaction

(calibre.gui2.actions.InterfaceAction attribute), 249

action_spec (calibre.gui2.actions.InterfaceAction attribute), 249

action_type (calibre.gui2.actions.InterfaceAction attribute), 249

add_annotation_to_library() (calibre.devices.usbms.device.Device method), 246

add_book() (calibre.devices.interface.BookList method), 243

add_books() (calibre.db.cache.Cache method), 313

add_books_to_metadata() (calibre.devices.interface.DevicePlugin class method), 240

add_books_to_metadata() (calibre.devices.usbms.driver.USBMS method), 247

add_custom_book_data() (calibre.db.cache.Cache method), 313

add_file() (calibre.ebooks.oeb.polish.container.Container method), 319

add_format() (calibre.db.cache.Cache method), 313

add_name_to_manifest() (calibre.ebooks.oeb.polish.container.Container method), 320

add_or_replace_jacket() (in module calibre.ebooks.oeb.polish.jacket), 324

add_properties() (calibre.ebooks.oeb.polish.container.Container method), 320

add_savepoint() (calibre.gui2.tweak_book.boss.Boss method), 327

add_toc_thumbnail() (calibre.web.feeds.news.BasicNewsRecipe method), 36

adeify_images() (calibre.web.feeds.news.BasicNewsRecipe class method), 36

all_book_ids() (calibre.db.cache.Cache method), 313

all_field_for() (calibre.db.cache.Cache method), 313

all_field_ids() (calibre.db.cache.Cache method), 313

all_field_keys() (calibre.ebooks.metadata.book.base.Metadata method), 180

all_field_names() (calibre.db.cache.Cache method), 313

all_non_none_fields() (calibre.ebooks.metadata.book.base.Metadata method), 180

allowed_in_menu (calibre.gui2.tweak_book.plugin.Tool attribute), 326

allowed_in_toolbar (calibre.gui2.tweak_book.plugin.Tool attribute), 326

API, **331**

apply_container_update_to_gui() (calibre.gui2.tweak_book.boss.Boss method), 327

apply_unique_properties() (calibre.ebooks.oeb.polish.container.Container method), 320

articles_are_obfuscated (calibre.web.feeds.news.BasicNewsRecipe attribute), 40

ASK_TO_ALLOW_CONNECT (calibre.devices.interface.DevicePlugin attribute), 237

author (calibre.customize.Plugin attribute), 226

author_data() (calibre.db.cache.Cache method), 313

author_sort_from_authors() (calibre.db.cache.Cache method), 314

auto_cleanup (calibre.web.feeds.news.BasicNewsRecipe attribute), 40

auto_cleanup_keep (calibre.web.feeds.news.BasicNewsRecipe attribute), 40

auto_repeat (calibre.gui2.actions.InterfaceAction attribute), 249

auto_trim_covers (calibre.ebooks.metadata.sources.base.Source attribute), 231

B

BasicNewsRecipe (class in calibre.web.feeds.news), 36

BCD (calibre.devices.interface.DevicePlugin attribute), 236

book_class (calibre.devices.usbms.driver.USBMS attribute), 246

book_type (calibre.ebooks.oeb.polish.container.Container attribute), 320

BookList (class in calibre.devices.interface), 243

booklist_class (calibre.devices.usbms.driver.USBMS attribute), 246

books() (calibre.devices.interface.DevicePlugin method), 240

books() (calibre.devices.usbms.driver.USBMS method), 247

books_for_field() (calibre.db.cache.Cache method), 314

books_in_virtual_library() (calibre.db.cache.Cache method), 314

Boss (class in calibre.gui2.tweak_book.boss), 327

boss() (calibre.gui2.tweak_book.plugin.Tool property), 326

BuiltinAdd (class in calibre.utils.formatter_functions), 165

BuiltinAnd (*class in calibre.utils.formatter_functions*), 166
BuiltinAnnotationCount (*class in calibre.utils.formatter_functions*), 168
BuiltinApproximateFormats (*class in calibre.utils.formatter_functions*), 168
BuiltinAssign (*class in calibre.utils.formatter_functions*), 176
BuiltinAuthorLinks (*class in calibre.utils.formatter_functions*), 168
BuiltinAuthorSorts (*class in calibre.utils.formatter_functions*), 169
BuiltinBooksize (*class in calibre.utils.formatter_functions*), 169
BuiltinCapitalize (*class in calibre.utils.formatter_functions*), 177
BuiltinCeiling (*class in calibre.utils.formatter_functions*), 165
BuiltinCheckYesNo (*class in calibre.utils.formatter_functions*), 172
BuiltinCmp (*class in calibre.utils.formatter_functions*), 177
BuiltinConnectedDeviceName (*class in calibre.utils.formatter_functions*), 169
BuiltinContains (*class in calibre.utils.formatter_functions*), 172
BuiltinCount (*class in calibre.utils.formatter_functions*), 174
BuiltinCurrentLibraryName (*class in calibre.utils.formatter_functions*), 169
BuiltinCurrentLibraryPath (*class in calibre.utils.formatter_functions*), 169
BuiltinDaysBetween (*class in calibre.utils.formatter_functions*), 167
BuiltinDivide (*class in calibre.utils.formatter_functions*), 165
BuiltinEval (*class in calibre.utils.formatter_functions*), 176
BuiltinField (*class in calibre.utils.formatter_functions*), 169
BuiltinFinishFormatting (*class in calibre.utils.formatter_functions*), 167
BuiltinFirstMatchingCmp (*class in calibre.utils.formatter_functions*), 177
BuiltinFirstNonEmpty (*class in calibre.utils.formatter_functions*), 172
BuiltinFloor (*class in calibre.utils.formatter_functions*), 165
BuiltinFormatDate (*class in calibre.utils.formatter_functions*), 167
BuiltinFormatNumber (*class in calibre.utils.formatter_functions*), 168
BuiltinFormatsModtimes (*class in calibre.utils.formatter_functions*), 170
BuiltinFormatsPaths (*class in calibre.utils.formatter_functions*), 170
BuiltinFormatsSizes (*class in calibre.utils.formatter_functions*), 170
BuiltinFractionalPart (*class in calibre.utils.formatter_functions*), 166
BuiltinHasCover (*class in calibre.utils.formatter_functions*), 170
BuiltinHumanReadable (*class in calibre.utils.formatter_functions*), 168
BuiltinIdentifierInList (*class in calibre.utils.formatter_functions*), 173
BuiltinIfempty (*class in calibre.utils.formatter_functions*), 172
BuiltinInList (*class in calibre.utils.formatter_functions*), 173
BuiltinLanguageCodes (*class in calibre.utils.formatter_functions*), 170
BuiltinLanguageStrings (*class in calibre.utils.formatter_functions*), 170
BuiltinListDifference (*class in calibre.utils.formatter_functions*), 174
BuiltinListEquals (*class in calibre.utils.formatter_functions*), 174
BuiltinListIntersection (*class in calibre.utils.formatter_functions*), 174
BuiltinListitem (*class in calibre.utils.formatter_functions*), 173
BuiltinListRe (*class in calibre.utils.formatter_functions*), 175
BuiltinListReGroup (*class in calibre.utils.formatter_functions*), 175
BuiltinListSort (*class in calibre.utils.formatter_functions*), 175
BuiltinListUnion (*class in calibre.utils.formatter_functions*), 175
BuiltinLookup (*class in calibre.utils.formatter_functions*), 172
BuiltinLowercase (*class in calibre.utils.formatter_functions*), 177
BuiltinMod (*class in calibre.utils.formatter_functions*), 166
BuiltinMultiply (*class in calibre.utils.formatter_functions*), 166
BuiltinNot (*class in calibre.utils.formatter_functions*), 166
BuiltinOndevice (*class in calibre.utils.formatter_functions*), 171
BuiltinOr (*class in calibre.utils.formatter_functions*), 167
BuiltinPrint (*class in calibre.utils.formatter_functions*), 176
BuiltinRatingToStars (*class in calibre.utils.formatter_functions*), 168

BuiltinRawField (*class in calibre.utils.formatter_functions*), 171
BuiltinRawList (*class in calibre.utils.formatter_functions*), 171
BuiltinRe (*class in calibre.utils.formatter_functions*), 178
BuiltinReGroup (*class in calibre.utils.formatter_functions*), 178
BuiltinRound (*class in calibre.utils.formatter_functions*), 166
BuiltinSelect (*class in calibre.utils.formatter_functions*), 173
BuiltinSeriesSort (*class in calibre.utils.formatter_functions*), 171
BuiltinShorten (*class in calibre.utils.formatter_functions*), 178
BuiltinStrcat (*class in calibre.utils.formatter_functions*), 178
BuiltinStrcatMax (*class in calibre.utils.formatter_functions*), 179
BuiltinStrcmp (*class in calibre.utils.formatter_functions*), 177
BuiltinStrInList (*class in calibre.utils.formatter_functions*), 174
BuiltinStrlen (*class in calibre.utils.formatter_functions*), 179
BuiltinSubitems (*class in calibre.utils.formatter_functions*), 175
BuiltinSublist (*class in calibre.utils.formatter_functions*), 176
BuiltinSubstr (*class in calibre.utils.formatter_functions*), 179
BuiltinSubtract (*class in calibre.utils.formatter_functions*), 166
BuiltinSwapAroundArticles (*class in calibre.utils.formatter_functions*), 179
BuiltinSwapAroundComma (*class in calibre.utils.formatter_functions*), 179
BuiltinSwitch (*class in calibre.utils.formatter_functions*), 173
BuiltinTemplate (*class in calibre.utils.formatter_functions*), 176
BuiltinTest (*class in calibre.utils.formatter_functions*), 172
BuiltinTitlecase (*class in calibre.utils.formatter_functions*), 177
BuiltinToday (*class in calibre.utils.formatter_functions*), 167
BuiltinTransliterate (*class in calibre.utils.formatter_functions*), 179
BuiltinUppercase (*class in calibre.utils.formatter_functions*), 178
BuiltinUserCategories (*class in calibre.utils.formatter_functions*), 171
BuiltinVirtualLibraries (*class in calibre.utils.formatter_functions*), 171

C

Cache (*class in calibre.db.cache*), 313
cached_cover_url_is_reliable (*calibre.ebooks.metadata.sources.base.Source attribute*), 230
calibre command line option
- --detach, 268
- --help, 268
- --ignore-plugins, 268
- --no-update-check, 268
- --shutdown-running-calibre, 268
- --start-in-tray, 268
- --verbose, 268
- --version, 268
- --with-library, 268
- -h, 268
- -s, 268
- -v, 268

calibre.customize
- module, 225

calibre.customize.conversion
- module, 234

calibre.db.cache
- module, 313

calibre.devices.interface
- module, 236

calibre.ebooks.metadata.book.base
- module, 180

calibre.ebooks.metadata.sources.base
- module, 230

calibre.ebooks.oeb.polish.container
- module, 319

calibre.ebooks.oeb.polish.cover
- module, 325

calibre.ebooks.oeb.polish.css
- module, 325

calibre.ebooks.oeb.polish.jacket
- module, 324

calibre.ebooks.oeb.polish.pretty
- module, 323

calibre.ebooks.oeb.polish.replace
- module, 323

calibre.ebooks.oeb.polish.split
- module, 324

calibre.ebooks.oeb.polish.toc
- module, 326

calibre.gui2.tweak_book.boss
- module, 327

calibre.utils.formatter_functions
- module, 165

calibre.web.feeds.news

module, 36
calibre-customize command line option
--add-plugin, 268
--build-plugin, 268
--customize-plugin, 268
--disable-plugin, 268
--enable-plugin, 268
--help, 268
--list-plugins, 268
--remove-plugin, 269
--version, 269
-a, 268
-b, 268
-h, 268
-l, 268
-r, 269
calibre-debug command line option
--add-simple-plugin, 269
--command, 269
--debug-device-driver, 269
--default-programs, 269
--diff, 269
--edit-book, 269
--exec-file, 269
--explode-book, 269
--export-all-calibre-data, 269
--fix-multiprocessing, 270
--gui, 270
--gui-debug, 270
--help, 270
--implode-book, 270
--import-calibre-data, 270
--inspect-mobi, 270
--paths, 270
--reinitialize-db, 270
--run-plugin, 270
--shutdown-running-calibre, 270
--subset-font, 270
--test-build, 270
--version, 270
--viewer, 270
-c, 269
-d, 269
-e, 269
-f, 270
-g, 270
-h, 270
-i, 270
-m, 270
-r, 270
-s, 270
-t, 269
-w, 270
-x, 269
calibre-server command line option
--access-log, 271
--ajax-timeout, 271
--auth-mode, 271
--auto-reload, 271
--ban-after, 271
--ban-for, 271
--book-list-mode, 271
--compress-min-size, 271
--custom-list-template, 271
--daemonize, 271
--disable-allow-socket-preallocation, 271
--disable-auth, 272
--disable-fallback-to-detected-interface, 272
--disable-local-write, 272
--disable-log-not-found, 272
--disable-use-bonjour, 272
--disable-use-sendfile, 272
--displayed-fields, 271
--enable-allow-socket-preallocation, 271
--enable-auth, 272
--enable-fallback-to-detected-interface, 272
--enable-local-write, 272
--enable-log-not-found, 272
--enable-use-bonjour, 272
--enable-use-sendfile, 272
--help, 272
--ignored-fields, 272
--listen-on, 272
--log, 272
--manage-users, 272
--max-header-line-size, 272
--max-job-time, 272
--max-jobs, 273
--max-log-size, 273
--max-opds-items, 273
--max-opds-ungrouped-items, 273
--max-request-body-size, 273
--num-per-page, 273
--pidfile, 273
--port, 273
--search-the-net-urls, 273
--shutdown-timeout, 273
--ssl-certfile, 273
--ssl-keyfile, 273
--timeout, 273
--trusted-ips, 273
--url-prefix, 273
--userdb, 274
--version, 274

--worker-count, 274
-h, 272
calibre-smtp command line option
--attachment, 275
--cafile, 275
--dont-verify-server-certificate, 275
--encryption-method, 275
--fork, 274
--help, 274
--localhost, 274
--outbox, 274
--password, 275
--port, 275
--relay, 275
--subject, 275
--timeout, 274
--username, 275
--verbose, 274
--version, 274
-a, 275
-e, 275
-f, 274
-h, 274
-l, 274
-o, 274
-p, 275
-r, 275
-s, 275
-t, 274
-u, 275
-v, 274
calibredb-add command line option
-1, 279
-I, 278
-S, 278
-T, 278
--add, 279
--authors, 278
--automerge, 278
--cover, 278
--duplicates, 278
--empty, 278
--identifier, 278
--ignore, 279
--isbn, 278
--languages, 278
--one-book-per-directory, 279
--recurse, 279
--series, 278
--series-index, 278
--tags, 278
--title, 278
-a, 278
-c, 278
-d, 278
-e, 278
-i, 278
-l, 278
-m, 278
-r, 279
-s, 278
-t, 278
calibredb-add_custom_column command line option
--display, 284
--is-multiple, 284
calibredb-add_format command line option
--dont-replace, 279
calibredb-backup_metadata command line option
--all, 287
calibredb-catalog command line option
--catalog-title, 282
--cross-reference-authors, 282
--debug-pipeline, 282
--exclude-genre, 282
--exclusion-rules, 282
--generate-authors, 282
--generate-descriptions, 282
--generate-genres, 282
--generate-recently-added, 283
--generate-series, 283
--generate-titles, 283
--genre-source-field, 283
--header-note-source-field, 283
--ids, 282
--merge-comments-rule, 283
--output-profile, 283
--prefix-rules, 283
--preset, 283
--search, 282
--thumb-width, 283
--use-existing-cover, 283
--verbose, 282
-i, 282
-s, 282
-v, 282
calibredb-check_library command line option
--csv, 286
--ignore_extensions, 286
--ignore_names, 286
--report, 286
-c, 286
-e, 286
-n, 286

-r, 286
calibredb-custom_columns command line option
--details, 284
-d, 284
calibredb-embed_metadata command line option
--only-formats, 287
-f, 287
calibredb-export command line option
--all, 281
--dont-asciiize, 281
--dont-save-cover, 281
--dont-update-metadata, 281
--dont-write-opf, 281
--formats, 281
--progress, 281
--replace-whitespace, 281
--single-dir, 281
--template, 281
--timefmt, 281
--to-dir, 281
--to-lowercase, 281
calibredb-list command line option
--ascending, 277
--fields, 277
--for-machine, 277
--limit, 277
--line-width, 277
--prefix, 277
--search, 277
--separator, 278
--sort-by, 278
-f, 277
-s, 277
-w, 277
calibredb-list_categories command line option
--categories, 286
--csv, 286
--dialect, 286
--item_count, 286
--width, 286
-c, 286
-i, 286
-r, 286
-w, 286
calibredb-remove command line option
--permanent, 279
calibredb-remove_custom_column command line option
--force, 285
-f, 285
calibredb-restore_database command line option
--really-do-it, 285
-r, 285
calibredb-search command line option
--limit, 288
-l, 288
calibredb-set_custom command line option
--append, 285
-a, 285
calibredb-set_metadata command line option
--field, 280
--list-fields, 280
-f, 280
-l, 280
calibredb-show_metadata command line option
--as-opf, 280
can_be_disabled (*calibre.customize.Plugin attribute*), 226
CAN_DO_DEVICE_DB_PLUGBOARD (*calibre.devices.interface.DevicePlugin attribute*), 236
can_get_multiple_covers (*calibre.ebooks.metadata.sources.base.Source attribute*), 231
can_handle() (*calibre.devices.interface.DevicePlugin method*), 238
can_handle_windows() (*calibre.devices.interface.DevicePlugin method*), 238
can_handle_windows() (*calibre.devices.usbms.device.Device method*), 245
CAN_SET_METADATA (*calibre.devices.interface.DevicePlugin attribute*), 236
canonicalize_internal_url() (*calibre.web.feeds.news.BasicNewsRecipe method*), 36
capabilities (*calibre.ebooks.metadata.sources.base.Source attribute*), 230
card_prefix() (*calibre.devices.interface.DevicePlugin method*), 239
card_prefix() (*calibre.devices.usbms.device.Device method*), 245
CatalogPlugin (*class in calibre.customize*), 230
category (*calibre.customize.PreferencesPlugin attribute*), 251
category_order (*calibre.customize.PreferencesPlugin attribute*), 251
center_navbar (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 40

change_font() (in module calibre.ebooks.oeb.polish.fonts), 325
changed_signal (calibre.gui2.preferences.ConfigWidgetInterface attribute), 251
clean_downloaded_metadata() (calibre.ebooks.metadata.sources.base.Source method), 231
cleanup() (calibre.web.feeds.news.BasicNewsRecipe method), 36
CLI (class in calibre.devices.usbms.cli), 246
cli_main() (calibre.customize.Plugin method), 228
cli_options (calibre.customize.CatalogPlugin attribute), 230
clone_browser() (calibre.web.feeds.news.BasicNewsRecipe method), 36
close_editor() (calibre.gui2.tweak_book.boss.Boss method), 328
command line option
  --help, 277
  --library-path, 277
  --password, 277
  --username, 277
  --version, 277
  --with-library, 277
  -h, 277
commit() (calibre.ebooks.oeb.polish.container.Container method), 320
commit() (calibre.gui2.preferences.ConfigWidgetBase method), 253
commit() (calibre.gui2.preferences.ConfigWidgetInterface method), 252
commit_all_editors_to_container() (calibre.gui2.tweak_book.boss.Boss method), 328
commit_item() (calibre.ebooks.oeb.polish.container.Container method), 320
common_options (calibre.customize.conversion.InputFormatPlugin attribute), 234
common_options (calibre.customize.conversion.OutputFormatPlugin attribute), 235
compress_news_images (calibre.web.feeds.news.BasicNewsRecipe attribute), 40
compress_news_images_auto_size (calibre.web.feeds.news.BasicNewsRecipe attribute), 40
compress_news_images_max_size (calibre.web.feeds.news.BasicNewsRecipe attribute), 40
config_help_message (calibre.ebooks.metadata.sources.base.Source attribute), 231
config_widget (calibre.customize.PreferencesPlugin attribute), 251
config_widget() (calibre.customize.Plugin method), 227
config_widget() (calibre.devices.interface.DevicePlugin class method), 241
config_widget() (calibre.ebooks.metadata.sources.base.Source method), 231
ConfigWidgetBase (class in calibre.gui2.preferences), 252
ConfigWidgetInterface (class in calibre.gui2.preferences), 251
Container (class in calibre.ebooks.oeb.polish.container), 319
contains(), **147**
conversion_options (calibre.web.feeds.news.BasicNewsRecipe attribute), 41
convert() (calibre.customize.conversion.InputFormatPlugin method), 234
convert() (calibre.customize.conversion.OutputFormatPlugin method), 235
copy_cover_to() (calibre.db.cache.Cache method), 314
copy_format_to() (calibre.db.cache.Cache method), 314
core_usage (calibre.customize.conversion.InputFormatPlugin attribute), 234
cover() (calibre.db.cache.Cache method), 314
cover_margins (calibre.web.feeds.news.BasicNewsRecipe attribute), 41
create_action() (calibre.gui2.tweak_book.plugin.Tool method), 327
create_inline_toc() (in module calibre.ebooks.oeb.polish.toc), 326
create_menu_action() (calibre.gui2.actions.InterfaceAction method), 249
create_widget() (calibre.customize.PreferencesPlugin method), 251
css, **351**
current_container() (calibre.gui2.tweak_book.plugin.Tool property), 326
currently_editing() (calibre.gui2.tweak_book.boss.Boss property), 328
custom_field_keys() (calibre.ebooks.metadata.book.base.Metadata method), 180
customization_help() (calibre.customize.Plugin method), 227
customization_help() (calibre.ebooks.metadata.sources.base.Source method), 231

D

data_for_find_identical_books() (*calibre.db.cache.Cache method*), 314

data_for_has_book() (*calibre.db.cache.Cache method*), 314

debug_managed_device_detection() (*calibre.devices.interface.DevicePlugin method*), 238

deepcopy() (*calibre.ebooks.metadata.book.base.Metadata method*), 180

default_cover() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37

delay (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 41

delete_books() (*calibre.devices.interface.DevicePlugin method*), 240

delete_books() (*calibre.devices.usbms.driver.USBMS method*), 248

delete_custom_book_data() (*calibre.db.cache.Cache method*), 314

description (*calibre.customize.Plugin attribute*), 226

description (*calibre.customize.PreferencesPlugin attribute*), 251

description (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 41

description() (*calibre.customize.conversion.OutputFormatPlugin property*), 235

detect_managed_devices() (*calibre.devices.interface.DevicePlugin method*), 237

Device (*class in calibre.devices.usbms.device*), 244

DevicePlugin (*class in calibre.devices.interface*), 236

dirty() (*calibre.ebooks.oeb.polish.container.Container method*), 320

do_user_config() (*calibre.customize.Plugin method*), 227

dont_add_to (*calibre.gui2.actions.InterfaceAction attribute*), 249

dont_remove_from (*calibre.gui2.actions.InterfaceAction attribute*), 249

download() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37

download_cover() (*calibre.ebooks.metadata.sources.base.Source method*), 233

drop_event() (*calibre.gui2.actions.InterfaceAction method*), 249

E

ebook-convert command line option
- --asciiize, 290
- --author-sort, 295
- --authors, 295
- --base-font-size, 290
- --book-producer, 295
- --change-justification, 290
- --chapter, 294
- --chapter-mark, 294
- --comments, 295
- --cover, 295
- --debug-pipeline, 296
- --disable-dehyphenate, 293
- --disable-delete-blank-paragraphs, 293
- --disable-fix-indents, 293
- --disable-font-rescaling, 290
- --disable-format-scene-breaks, 293
- --disable-italicize-common-cases, 293
- --disable-markup-chapter-headings, 293
- --disable-remove-fake-margins, 294
- --disable-renumber-headings, 293
- --disable-unwrap-lines, 293
- --dont-split-on-page-breaks, 289
- --duplicate-links-in-toc, 295
- --embed-all-fonts, 291
- --embed-font-family, 291
- --enable-heuristics, 293
- --epub-flatten, 289
- --epub-inline-toc, 289
- --epub-toc-at-end, 289
- --epub-version, 290
- --expand-css, 291
- --extra-css, 291
- --extract-to, 290
- --filter-css, 291
- --flow-size, 290
- --font-size-mapping, 291
- --from-opf, 296
- --help, 289
- --html-unwrap-factor, 293
- --input-encoding, 289
- --input-profile, 289
- --insert-blank-line, 291
- --insert-blank-line-size, 291
- --insert-metadata, 294
- --isbn, 295
- --keep-ligatures, 291
- --language, 296
- --level1-toc, 295
- --level2-toc, 295
- --level3-toc, 295
- --line-height, 291
- --linearize-tables, 291
- --list-recipes, 289
- --margin-bottom, 291
- --margin-left, 292
- --margin-right, 292

--margin-top, 292
--max-toc-links, 295
--minimum-line-height, 292
--no-chapters-in-toc, 295
--no-default-epub-cover, 290
--no-svg-cover, 290
--output-profile, 289
--page-breaks-before, 294
--prefer-metadata-cover, 294
--preserve-cover-aspect-ratio, 290
--pretty-print, 290
--pubdate, 296
--publisher, 296
--rating, 296
--read-metadata-from-opf, 296
--remove-first-image, 294
--remove-paragraph-spacing, 292
--remove-paragraph-spacing-indent-size, 292
--replace-scene-breaks, 293
--search-replace, 293
--series, 296
--series-index, 296
--smarten-punctuation, 292
--sr1-replace, 293
--sr1-search, 293
--sr2-replace, 293
--sr2-search, 294
--sr3-replace, 294
--sr3-search, 294
--start-reading-at, 294
--subset-embedded-fonts, 292
--tags, 296
--timestamp, 296
--title, 296
--title-sort, 296
--toc-filter, 295
--toc-threshold, 295
--toc-title, 290
--transform-css-rules, 292
--unsmarten-punctuation, 292
--use-auto-toc, 295
--verbose, 296
--version, 289
-d, 296
-h, 289
-m, 296
-v, 296

ebook-edit command line option
--detach, 297
--help, 297
--version, 297
-h, 297

ebook-meta command line option
--author-sort, 297
--authors, 297
--book-producer, 297
--category, 297
--comments, 297
--cover, 298
--date, 298
--from-opf, 298
--get-cover, 298
--help, 298
--identifier, 298
--index, 298
--isbn, 298
--language, 298
--lrf-bookid, 298
--publisher, 298
--rating, 298
--series, 298
--tags, 298
--title, 298
--title-sort, 298
--to-opf, 298
--version, 298
-a, 297
-c, 297
-d, 298
-h, 298
-i, 298
-k, 297
-l, 298
-p, 298
-r, 298
-s, 298
-t, 298

ebook-polish command line option
-H, 299
-U, 300
--add-soft-hyphens, 299
--compress-images, 299
--cover, 299
--embed-fonts, 299
--help, 299
--jacket, 299
--opf, 299
--remove-jacket, 299
--remove-soft-hyphens, 299
--remove-unused-css, 299
--smarten-punctuation, 299
--subset-fonts, 299
--upgrade-book, 300
--verbose, 300
--version, 300
-c, 299
-e, 299

-f, 299
-h, 299
-i, 299
-j, 299
-o, 299
-p, 299
-u, 299
ebook-viewer command line option
--continue, 300
--detach, 300
--force-reload, 300
--full-screen, 300
--fullscreen, 300
--help, 300
--open-at, 300
--raise-window, 300
--version, 300
-f, 300
-h, 300
edit_file() (*calibre.gui2.tweak_book.boss.Boss method*), 328
eject() (*calibre.devices.interface.DevicePlugin method*), 239
eject() (*calibre.devices.usbms.device.Device method*), 246
embed_metadata() (*calibre.db.cache.Cache method*), 314
encoding (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 41
exists() (*calibre.ebooks.oeb.polish.container.Container method*), 320
extra_css (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 41
extract_readable_article() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37

F

fast_field_for() (*calibre.db.cache.Cache method*), 314
feeds (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 41
fetch-ebook-metadata command line option
-I, 301
--allowed-plugin, 301
--authors, 301
--cover, 301
--help, 301
--identifier, 301
--isbn, 301
--opf, 301
--timeout, 301
--title, 301
--verbose, 301
--version, 301
-a, 301
-c, 301
-d, 301
-h, 301
-i, 301
-o, 301
-p, 301
-t, 301
-v, 301
field_for() (*calibre.db.cache.Cache method*), 314
field_ids_for() (*calibre.db.cache.Cache method*), 315
file_type (*calibre.customize.conversion.OutputFormatPlugin attribute*), 235
file_types (*calibre.customize.CatalogPlugin attribute*), 230
file_types (*calibre.customize.conversion.InputFormatPlugin attribute*), 234
file_types (*calibre.customize.FileTypePlugin attribute*), 228
file_types (*calibre.customize.MetadataReaderPlugin attribute*), 229
file_types (*calibre.customize.MetadataWriterPlugin attribute*), 229
filename_callback() (*calibre.devices.usbms.device.Device method*), 246
filesize() (*calibre.ebooks.oeb.polish.container.Container method*), 320
FileTypePlugin (*class in calibre.customize*), 228
filter_css() (*in module calibre.ebooks.oeb.polish.css*), 325
filter_regexps (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 41
find_identical_books() (*calibre.db.cache.Cache method*), 315
fix_all_html() (*in module calibre.ebooks.oeb.polish.pretty*), 323
fix_html() (*in module calibre.ebooks.oeb.polish.pretty*), 323
for_viewer (*calibre.customize.conversion.InputFormatPlugin attribute*), 234
format() (*calibre.db.cache.Cache method*), 315
format_abspath() (*calibre.db.cache.Cache method*), 315
format_field() (*calibre.ebooks.metadata.book.base.Metadata method*), 181
format_hash() (*calibre.db.cache.Cache method*), 315
format_metadata() (*calibre.db.cache.Cache method*), 315
FORMATS (*calibre.devices.interface.DevicePlugin*

attribute), 236
formats() (*calibre.db.cache.Cache method*), 315
free_space() (*calibre.devices.interface.DevicePlugin method*), 239
free_space() (*calibre.devices.usbms.device.Device method*), 245
from_files() (*in module calibre.ebooks.oeb.polish.toc*), 326
from_links() (*in module calibre.ebooks.oeb.polish.toc*), 326
from_xpaths() (*in module calibre.ebooks.oeb.polish.toc*), 326

G

generate_item() (*calibre.ebooks.oeb.polish.container.Container method*), 320
genesis() (*calibre.gui2.actions.InterfaceAction method*), 250
genesis() (*calibre.gui2.preferences.ConfigWidgetInterface method*), 252
get_all_standard_metadata() (*calibre.ebooks.metadata.book.base.Metadata method*), 180
get_all_user_metadata() (*calibre.ebooks.metadata.book.base.Metadata method*), 181
get_annotations() (*calibre.devices.usbms.device.Device method*), 246
get_article_url() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37
get_author_tokens() (*calibre.ebooks.metadata.sources.base.Source method*), 231
get_book_url() (*calibre.ebooks.metadata.sources.base.Source method*), 232
get_book_url_name() (*calibre.ebooks.metadata.sources.base.Source method*), 232
get_book_urls() (*calibre.ebooks.metadata.sources.base.Source method*), 232
get_browser() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37
get_cached_cover_url() (*calibre.ebooks.metadata.sources.base.Source method*), 232
get_categories() (*calibre.db.cache.Cache method*), 316
get_collections() (*calibre.devices.interface.BookList method*), 243
get_cover_url() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37
get_custom_book_data() (*calibre.db.cache.Cache method*), 316
get_device_information() (*calibre.devices.interface.DevicePlugin method*), 239
get_device_information() (*calibre.devices.usbms.driver.USBMS method*), 246
get_device_uid() (*calibre.devices.interface.DevicePlugin method*), 241
get_driveinfo() (*calibre.devices.interface.DevicePlugin method*), 239
get_extra_css() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37
get_feeds() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37
get_file() (*calibre.devices.interface.DevicePlugin method*), 241
get_file_path_for_processing() (*calibre.ebooks.oeb.polish.container.Container method*), 320
get_id_map() (*calibre.db.cache.Cache method*), 316
get_identifiers() (*calibre.ebooks.metadata.book.base.Metadata method*), 180
get_ids_for_custom_book_data() (*calibre.db.cache.Cache method*), 316
get_images() (*calibre.customize.conversion.InputFormatPlugin method*), 234
get_item_id() (*calibre.db.cache.Cache method*), 316
get_item_ids() (*calibre.db.cache.Cache method*), 316
get_item_name() (*calibre.db.cache.Cache method*), 316
get_masthead_title() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37
get_masthead_url() (*calibre.web.feeds.news.BasicNewsRecipe method*), 37
get_metadata() (*calibre.customize.MetadataReaderPlugin method*), 229
get_metadata() (*calibre.db.cache.Cache method*), 316
get_next_series_num_for() (*calibre.db.cache.Cache method*), 316
get_obfuscated_article() (*calibre.web.feeds.news.BasicNewsRecipe method*), 38
get_option() (*calibre.devices.interface.DevicePlugin method*), 242
get_proxy_metadata() (*calibre.db.cache.Cache method*), 316

get_recommended_folders() (*in module calibre.ebooks.oeb.polish.replace*), 323
get_standard_metadata() (*calibre.ebooks.metadata.book.base.Metadata method*), 180
get_title_tokens() (*calibre.ebooks.metadata.sources.base.Source method*), 231
get_usage_count_by_id() (*calibre.db.cache.Cache method*), 316
get_user_blacklisted_devices() (*calibre.devices.interface.DevicePlugin method*), 242
get_user_metadata() (*calibre.ebooks.metadata.book.base.Metadata method*), 181
gui() (*calibre.gui2.tweak_book.plugin.Tool property*), 326
gui_category (*calibre.customize.PreferencesPlugin attribute*), 251
gui_configuration_widget() (*calibre.customize.conversion.InputFormatPlugin method*), 235
gui_configuration_widget() (*calibre.customize.conversion.OutputFormatPlugin method*), 236
gui_layout_complete() (*calibre.gui2.actions.InterfaceAction method*), 251
gui_name (*calibre.customize.PreferencesPlugin attribute*), 251
guide_type_map() (*calibre.ebooks.oeb.polish.container.Container property*), 320

H

handle_gzip (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 41
has_book() (*calibre.db.cache.Cache method*), 316
has_format() (*calibre.db.cache.Cache method*), 316
has_html_comments (*calibre.ebooks.metadata.sources.base.Source attribute*), 230
has_id() (*calibre.db.cache.Cache method*), 316
has_name() (*calibre.ebooks.oeb.polish.container.Container method*), 320
href_to_name() (*calibre.ebooks.oeb.polish.container.Container method*), 320
HTML, **331**

I

icon (*calibre.customize.PreferencesPlugin attribute*), 251
icon (*calibre.devices.interface.DevicePlugin attribute*), 237
id_from_url() (*calibre.ebooks.metadata.sources.base.Source method*), 232
identify() (*calibre.ebooks.metadata.sources.base.Source method*), 232
identify_results_keygen() (*calibre.ebooks.metadata.sources.base.Source method*), 232
ignore_connected_device() (*calibre.devices.interface.DevicePlugin method*), 241
ignore_duplicate_articles (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
ignore_ssl_errors (*calibre.ebooks.metadata.sources.base.Source attribute*), 230
image_url_processor() (*calibre.web.feeds.news.BasicNewsRecipe class method*), 38
index_to_soup() (*calibre.web.feeds.news.BasicNewsRecipe method*), 38
init() (*calibre.db.cache.Cache method*), 317
initialization_complete() (*calibre.gui2.actions.InterfaceAction method*), 251
initialize() (*calibre.customize.CatalogPlugin method*), 230
initialize() (*calibre.customize.Plugin method*), 226
initialize() (*calibre.gui2.preferences.ConfigWidgetBase method*), 252
initialize() (*calibre.gui2.preferences.ConfigWidgetInterface method*), 252
InputFormatPlugin (*class in calibre.customize.conversion*), 234
insert_into_xml() (*calibre.ebooks.oeb.polish.container.Container method*), 320
InterfaceAction (*class in calibre.gui2.actions*), 248
InterfaceActionBase (*class in calibre.customize*), 251
InternalMetadataCompareKeyGen (*class in calibre.ebooks.metadata.sources.base*), 233
is_configured() (*calibre.ebooks.metadata.sources.base.Source method*), 231
is_dir (*calibre.ebooks.oeb.polish.container.Container attribute*), 321
is_dynamically_controllable() (*calibre.devices.interface.DevicePlugin method*), 242
is_image_collection (*calibre.customize.conversion.InputFormatPlugin attribute*), 234
is_link_wanted() (*calibre.web.feeds.news.BasicNewsRecipe

method), 38
is_null() (*calibre.ebooks.metadata.book.base.Metadata method*), 180
is_running() (*calibre.devices.interface.DevicePlugin method*), 242
is_usb_connected() (*calibre.devices.interface.DevicePlugin method*), 237
iterlinks() (*calibre.ebooks.oeb.polish.container.Container method*), 321

K

keep_only_tags (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42

L

language (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
library_changed() (*calibre.gui2.actions.InterfaceAction method*), 250
load_actual_plugin() (*calibre.customize.InterfaceActionBase method*), 251
load_resources() (*calibre.customize.Plugin method*), 227
load_resources() (*calibre.gui2.actions.InterfaceAction method*), 250
location_selected() (*calibre.gui2.actions.InterfaceAction method*), 250
LRF, **331**
lrf2lrs command line option
- --dont-output-resources, 302
- --help, 302
- --output, 302
- --verbose, 302
- --version, 302
- -h, 302
- -o, 302

lrfviewer command line option
- --disable-hyphenation, 302
- --help, 302
- --profile, 302
- --verbose, 302
- --version, 302
- --visual-debug, 302
- --white-background, 302
- -h, 302

lrs2lrf command line option
- --help, 303
- --lrs, 303
- --output, 303
- --verbose, 303
- --version, 303
- -h, 303
- -o, 303

M

make_name_unique() (*calibre.ebooks.oeb.polish.container.Container method*), 321
MANAGES_DEVICE_PRESENCE (*calibre.devices.interface.DevicePlugin attribute*), 237
manifest_has_name() (*calibre.ebooks.oeb.polish.container.Container method*), 321
manifest_id_map() (*calibre.ebooks.oeb.polish.container.Container property*), 321
manifest_items_of_type() (*calibre.ebooks.oeb.polish.container.Container method*), 321
manifest_items_with_property() (*calibre.ebooks.oeb.polish.container.Container method*), 321
manifest_type_map() (*calibre.ebooks.oeb.polish.container.Container property*), 321
mark_as_cover() (*in module calibre.ebooks.oeb.polish.cover*), 325
mark_as_titlepage() (*in module calibre.ebooks.oeb.polish.cover*), 325
masthead_url (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
match_regexps (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
max_articles_per_feed (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
MAX_PATH_LEN (*calibre.devices.usbms.device.Device attribute*), 244
merge() (*in module calibre.ebooks.oeb.polish.split*), 324
Metadata (*class in calibre.ebooks.metadata.book.base*), 180
metadata_for_field() (*calibre.ebooks.metadata.book.base.Metadata method*), 180
MetadataReaderPlugin (*class in calibre.customize*), 229
MetadataWriterPlugin (*class in calibre.customize*), 229
mi() (*calibre.ebooks.oeb.polish.container.Container property*), 321
minimum_calibre_version (*calibre.customize.Plugin attribute*), 226

module
calibre.customize, 225
calibre.customize.conversion, 234
calibre.db.cache, 313
calibre.devices.interface, 236
calibre.ebooks.metadata.book.base, 180
calibre.ebooks.metadata.sources.base, 230
calibre.ebooks.oeb.polish.container, 319
calibre.ebooks.oeb.polish.cover, 325
calibre.ebooks.oeb.polish.css, 325
calibre.ebooks.oeb.polish.jacket, 324
calibre.ebooks.oeb.polish.pretty, 323
calibre.ebooks.oeb.polish.replace, 323
calibre.ebooks.oeb.polish.split, 324
calibre.ebooks.oeb.polish.toc, 326
calibre.gui2.tweak_book.boss, 327
calibre.utils.formatter_functions, 165
calibre.web.feeds.news, 36
multisort() (*calibre.db.cache.Cache method*), 317
multisplit() (*in module calibre.ebooks.oeb.polish.split*), 324

N

name (*calibre.customize.Plugin attribute*), 226
name (*calibre.gui2.actions.InterfaceAction attribute*), 249
name (*calibre.gui2.tweak_book.plugin.Tool attribute*), 326
name(), **147**
name_order (*calibre.customize.PreferencesPlugin attribute*), 251
name_to_abspath() (*calibre.ebooks.oeb.polish.container.Container method*), 321
name_to_href() (*calibre.ebooks.oeb.polish.container.Container method*), 321
names_that_must_not_be_changed() (*calibre.ebooks.oeb.polish.container.Container property*), 321
names_that_must_not_be_removed() (*calibre.ebooks.oeb.polish.container.Container property*), 321
names_that_need_not_be_manifested() (*calibre.ebooks.oeb.polish.container.Container property*), 321
needs_subscription (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
NEWS_IN_FOLDER (*calibre.devices.usbms.device.Device attribute*), 244
no_stylesheets (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
normalize_path() (*calibre.devices.usbms.driver.USBMS class method*), 248
NUKE_COMMENTS (*calibre.devices.interface.DevicePlugin attribute*), 237

O

oldest_article (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
on_import (*calibre.customize.FileTypePlugin attribute*), 228
on_postimport (*calibre.customize.FileTypePlugin attribute*), 228
on_postprocess (*calibre.customize.FileTypePlugin attribute*), 228
on_preprocess (*calibre.customize.FileTypePlugin attribute*), 228
open() (*calibre.devices.interface.DevicePlugin method*), 238
open() (*calibre.devices.usbms.device.Device method*), 246
open() (*calibre.ebooks.oeb.polish.container.Container method*), 321
open_book() (*calibre.gui2.tweak_book.boss.Boss method*), 328
OPEN_FEEDBACK_MESSAGE (*calibre.devices.interface.DevicePlugin attribute*), 237
opf() (*calibre.ebooks.oeb.polish.container.Container property*), 321
opf_get_or_create() (*calibre.ebooks.oeb.polish.container.Container method*), 321
opf_version() (*calibre.ebooks.oeb.polish.container.Container property*), 321
opf_version_parsed() (*calibre.ebooks.oeb.polish.container.Container property*), 322
opf_xpath() (*calibre.ebooks.oeb.polish.container.Container method*), 322
options (*calibre.customize.conversion.InputFormatPlugin attribute*), 234
options (*calibre.customize.conversion.OutputFormatPlugin attribute*), 235
options (*calibre.ebooks.metadata.sources.base.Source attribute*), 230
OSX_MAIN_MEM_VOL_PAT (*calibre.devices.usbms.device.Device attribute*), 244

output_encoding (*calibre.customize.conversion.InputFormatPlugin attribute*), 234
OutputFormatPlugin (*class in calibre.customize.conversion*), 235

P

parse_feeds() (*calibre.web.feeds.news.BasicNewsRecipe method*), 38
parse_index() (*calibre.web.feeds.news.BasicNewsRecipe method*), 38
parsed() (*calibre.ebooks.oeb.polish.container.Container method*), 322
path_sep (*calibre.devices.interface.DevicePlugin attribute*), 237
Plugin (*class in calibre.customize*), 226
populate_article_metadata() (*calibre.web.feeds.news.BasicNewsRecipe method*), 38
popup_type (*calibre.gui2.actions.InterfaceAction attribute*), 249
post_yank_cleanup() (*calibre.devices.interface.DevicePlugin method*), 239
post_yank_cleanup() (*calibre.devices.usbms.device.Device method*), 246
postadd() (*calibre.customize.FileTypePlugin method*), 228
postimport() (*calibre.customize.FileTypePlugin method*), 228
postprocess_book() (*calibre.customize.conversion.InputFormatPlugin method*), 235
postprocess_book() (*calibre.web.feeds.news.BasicNewsRecipe method*), 39
postprocess_html() (*calibre.web.feeds.news.BasicNewsRecipe method*), 39
pref() (*calibre.db.cache.Cache method*), 317
prefer_results_with_isbn (*calibre.ebooks.metadata.sources.base.Source attribute*), 231
PreferencesPlugin (*class in calibre.customize*), 251
prepare_addable_books() (*calibre.devices.interface.DevicePlugin method*), 241
preprocess_html() (*calibre.web.feeds.news.BasicNewsRecipe method*), 39
preprocess_image() (*calibre.web.feeds.news.BasicNewsRecipe method*), 39
preprocess_raw_html() (*calibre.web.feeds.news.BasicNewsRecipe method*), 39
preprocess_regexps (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 42
pretty_all() (*in module calibre.ebooks.oeb.polish.pretty*), 324
pretty_css() (*in module calibre.ebooks.oeb.polish.pretty*), 323
pretty_html() (*in module calibre.ebooks.oeb.polish.pretty*), 323
pretty_xml() (*in module calibre.ebooks.oeb.polish.pretty*), 324
print_version() (*calibre.web.feeds.news.BasicNewsRecipe class method*), 39
priority (*calibre.customize.Plugin attribute*), 226
priority (*calibre.gui2.actions.InterfaceAction attribute*), 249
PRODUCT_ID (*calibre.devices.interface.DevicePlugin attribute*), 236
publication_type (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43

R

raw_data() (*calibre.ebooks.oeb.polish.container.Container method*), 322
re:test(), **147**
read_backup() (*calibre.db.cache.Cache method*), 317
recipe_disabled (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
recommendations (*calibre.customize.conversion.InputFormatPlugin attribute*), 234
recommendations (*calibre.customize.conversion.OutputFormatPlugin attribute*), 235
recursions (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
refresh_gui() (*calibre.gui2.preferences.ConfigWidgetInterface method*), 252
regexp, **331**
register() (*calibre.gui2.preferences.ConfigWidgetBase method*), 252
register_shortcut() (*calibre.gui2.tweak_book.plugin.Tool method*), 326
relpath() (*calibre.ebooks.oeb.polish.container.Container method*), 322
remove_attributes (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
remove_book() (*calibre.devices.interface.BookList method*), 243

remove_books() (*calibre.db.cache.Cache method*), 317
remove_books_from_metadata() (*calibre.devices.interface.DevicePlugin class method*), 240
remove_books_from_metadata() (*calibre.devices.usbms.driver.USBMS method*), 248
remove_empty_feeds (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
remove_formats() (*calibre.db.cache.Cache method*), 317
remove_from_spine() (*calibre.ebooks.oeb.polish.container.Container method*), 322
remove_from_xml() (*calibre.ebooks.oeb.polish.container.Container method*), 322
remove_item() (*calibre.ebooks.oeb.polish.container.Container method*), 322
remove_items() (*calibre.db.cache.Cache method*), 317
remove_jacket() (*in module calibre.ebooks.oeb.polish.jacket*), 324
remove_javascript (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
remove_stale_user_metadata() (*calibre.ebooks.metadata.book.base.Metadata method*), 181
remove_tags (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
remove_tags_after (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
remove_tags_before (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 43
remove_unused_css() (*in module calibre.ebooks.oeb.polish.css*), 325
rename() (*calibre.ebooks.oeb.polish.container.Container method*), 322
rename_files() (*in module calibre.ebooks.oeb.polish.replace*), 323
rename_items() (*calibre.db.cache.Cache method*), 317
replace() (*calibre.ebooks.oeb.polish.container.Container method*), 322
replace_links() (*calibre.ebooks.oeb.polish.container.Container method*), 322
replace_links() (*in module calibre.ebooks.oeb.polish.replace*), 323
requires_version (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
resepi, **331**
reset() (*calibre.devices.interface.DevicePlugin method*), 238
reset() (*calibre.devices.usbms.device.Device method*), 244
resolve_internal_links (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
restart_critical (*calibre.gui2.preferences.ConfigWidgetInterface attribute*), 252
restore_book() (*calibre.db.cache.Cache method*), 317
restore_defaults() (*calibre.gui2.preferences.ConfigWidgetBase method*), 253
restore_defaults() (*calibre.gui2.preferences.ConfigWidgetInterface method*), 252
restore_defaults_desc (*calibre.gui2.preferences.ConfigWidgetInterface attribute*), 252
restore_original_format() (*calibre.db.cache.Cache method*), 317
reverse_article_order (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
rewind_savepoint() (*calibre.gui2.tweak_book.boss.Boss method*), 328
RSS, **331**
run() (*calibre.customize.CatalogPlugin method*), 230
run() (*calibre.customize.FileTypePlugin method*), 228

S

safe_read_lock() (*calibre.db.cache.Cache property*), 317
sanitize_callback() (*calibre.devices.usbms.device.Device method*), 246
sanitize_path_components() (*calibre.devices.usbms.device.Device method*), 246
save_book() (*calibre.gui2.tweak_book.boss.Boss method*), 328
save_original_format() (*calibre.db.cache.Cache method*), 318
save_settings() (*calibre.customize.Plugin method*), 227
save_settings() (*calibre.devices.interface.DevicePlugin class method*), 241
save_settings() (*calibre.ebooks.metadata.sources.base.Source method*), 231

scale_news_images (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44

scale_news_images_to_device (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44

search() (*calibre.db.cache.Cache method*), 318

serialize_item() (*calibre.ebooks.oeb.polish.container.Container method*), 322

set_all_user_metadata() (*calibre.ebooks.metadata.book.base.Metadata method*), 181

set_conversion_options() (*calibre.db.cache.Cache method*), 318

set_cover() (*calibre.db.cache.Cache method*), 318

set_cover() (*in module calibre.ebooks.oeb.polish.cover*), 325

set_driveinfo_name() (*calibre.devices.interface.DevicePlugin method*), 241

set_driveinfo_name() (*calibre.devices.usbms.driver.USBMS method*), 247

set_field() (*calibre.db.cache.Cache method*), 318

set_identifier() (*calibre.ebooks.metadata.book.base.Metadata method*), 180

set_identifiers() (*calibre.ebooks.metadata.book.base.Metadata method*), 180

set_library_info() (*calibre.devices.interface.DevicePlugin method*), 242

set_metadata() (*calibre.customize.MetadataWriterPlugin method*), 229

set_metadata() (*calibre.db.cache.Cache method*), 318

set_modified() (*calibre.gui2.tweak_book.boss.Boss method*), 328

set_option() (*calibre.devices.interface.DevicePlugin method*), 242

set_plugboards() (*calibre.devices.interface.DevicePlugin method*), 241

set_pref() (*calibre.db.cache.Cache method*), 318

set_progress_reporter() (*calibre.devices.interface.DevicePlugin method*), 239

set_progress_reporter() (*calibre.devices.usbms.device.Device method*), 245

set_spine() (*calibre.ebooks.oeb.polish.container.Container method*), 322

set_user_blacklisted_devices() (*calibre.devices.interface.DevicePlugin method*), 242

set_user_metadata() (*calibre.ebooks.metadata.book.base.Metadata method*), 181

settings() (*calibre.devices.interface.DevicePlugin class method*), 241

show_current_diff() (*calibre.gui2.tweak_book.boss.Boss method*), 328

show_editor() (*calibre.gui2.tweak_book.boss.Boss method*), 328

shutdown() (*calibre.devices.interface.DevicePlugin method*), 241

shutting_down() (*calibre.gui2.actions.InterfaceAction method*), 251

simultaneous_downloads (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44

skip_ad_pages() (*calibre.web.feeds.news.BasicNewsRecipe method*), 39

SLOW_DRIVEINFO (*calibre.devices.interface.DevicePlugin attribute*), 237

smart_update() (*calibre.ebooks.metadata.book.base.Metadata method*), 181

sort_index_by() (*calibre.web.feeds.news.BasicNewsRecipe method*), 40

Source (*class in calibre.ebooks.metadata.sources.base*), 230

specialize() (*calibre.customize.conversion.InputFormatPlugin method*), 235

specialize_css_for_output() (*calibre.customize.conversion.OutputFormatPlugin method*), 236

specialize_global_preferences() (*calibre.devices.interface.DevicePlugin method*), 242

specialize_options() (*calibre.customize.conversion.OutputFormatPlugin method*), 235

spine_items() (*calibre.ebooks.oeb.polish.container.Container property*), 323

spine_iter() (*calibre.ebooks.oeb.polish.container.Container property*), 323

spine_names() (*calibre.ebooks.oeb.polish.container.Container property*), 323

split() (*in module calibre.ebooks.oeb.polish.split*), 324

split_jobs() (*calibre.ebooks.metadata.sources.base.Source method*), 231

standard_field_keys() (*calibre.ebooks.metadata.book.base.Metadata method*), 180

STANDARD_METADATA_FIELDS (*in module calibre.ebooks.metadata.book.base*), 181

start_plugin() (*calibre.devices.interface.DevicePlugin*

method), 242
startup() (*calibre.devices.interface.DevicePlugin method*), 241
stop_plugin() (*calibre.devices.interface.DevicePlugin method*), 242
summary_length (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
supported_platforms (*calibre.customize.Plugin attribute*), 226
supports_collections() (*calibre.devices.interface.BookList method*), 243
supports_gzip_transfer_encoding (*calibre.ebooks.metadata.sources.base.Source attribute*), 230
supports_restoring_to_defaults (*calibre.gui2.preferences.ConfigWidgetInterface attribute*), 252
sync_booklists() (*calibre.devices.interface.DevicePlugin method*), 240
sync_booklists() (*calibre.devices.usbms.driver.USBMS method*), 248
sync_preview_to_editor() (*calibre.gui2.tweak_book.boss.Boss method*), 328
synchronize_with_db() (*calibre.devices.interface.DevicePlugin method*), 242

T

tag_to_string() (*calibre.web.feeds.news.BasicNewsRecipe class method*), 40
tags_older_than() (*calibre.db.cache.Cache method*), 318
template_css (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
template_to_attribute() (*calibre.ebooks.metadata.book.base.Metadata method*), 181
temporary_file() (*calibre.customize.Plugin method*), 227
test_fields() (*calibre.ebooks.metadata.sources.base.Source method*), 231
THUMBNAIL_COMPRESSION_QUALITY (*calibre.devices.interface.DevicePlugin attribute*), 236
THUMBNAIL_HEIGHT (*calibre.devices.interface.DevicePlugin attribute*), 236
timefmt (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
timeout (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
title (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
to_html() (*calibre.ebooks.metadata.book.base.Metadata method*), 181
Tool (*class in calibre.gui2.tweak_book.plugin*), 326
toolbar_button_popup_mode (*calibre.gui2.tweak_book.plugin.Tool attribute*), 326
total_space() (*calibre.devices.interface.DevicePlugin method*), 239
total_space() (*calibre.devices.usbms.device.Device method*), 245
touched_fields (*calibre.ebooks.metadata.sources.base.Source attribute*), 230
type (*calibre.customize.Plugin attribute*), 226

U

upload_books() (*calibre.devices.interface.DevicePlugin method*), 240
upload_books() (*calibre.devices.usbms.driver.USBMS method*), 247
upload_cover() (*calibre.devices.usbms.driver.USBMS method*), 247
URL, **331**
USBMS (*class in calibre.devices.usbms.driver*), 246
use_embedded_content (*calibre.web.feeds.news.BasicNewsRecipe attribute*), 44
user_categories_for_books() (*calibre.db.cache.Cache method*), 319
user_feedback_after_callback (*calibre.devices.interface.DevicePlugin attribute*), 237
UserAnnotation (*calibre.devices.interface.DevicePlugin attribute*), 237

V

VENDOR_ID (*calibre.devices.interface.DevicePlugin attribute*), 236
version (*calibre.customize.Plugin attribute*), 226
VIRTUAL_BOOK_EXTENSION_MESSAGE (*calibre.devices.interface.DevicePlugin attribute*), 237
VIRTUAL_BOOK_EXTENSIONS (*calibre.devices.interface.DevicePlugin attribute*), 237

W

WANTS_UPDATED_THUMBNAILS (*calibre.devices.interface.DevicePlugin attribute*), 236
web2disk command line option
  --base-dir, 303
  --delay, 303
  --dont-download-stylesheets, 303
  --encoding, 303

- --filter-regexp, 303
- --help, 304
- --match-regexp, 304
- --max-files, 304
- --max-recursions, 304
- --timeout, 304
- --verbose, 304
- --version, 304
- -d, 303
- -h, 304
- -n, 304
- -r, 304
- -t, 304

WINDOWS_CARD_A_MEM (*calibre.devices.usbms.device.Device attribute*), 244

WINDOWS_CARD_B_MEM (*calibre.devices.usbms.device.Device attribute*), 244

WINDOWS_MAIN_MEM (*calibre.devices.usbms.device.Device attribute*), 244

windows_sort_drives() (*calibre.devices.usbms.device.Device method*), 245