

- To make batch assignments, select all of the types you want to assign in the left list using **Shift** and **Cmd** clicking to add to the select, and then click on the tile these types should all be formatted like.
- If you know the name of the layout you want, you can also right-click on a selection of types and select the layout to be used.
- The special “As-Is” layout at the bottom of the list is a way of saying a type of document shouldn’t use any special formatting. The contents of it will be passed directly through to the compiler, with no additions, embellishments or adjustments (though separators, like page breaks or other markers, may still be inserted around the items, as to be expected).

When you have everything “wired up” the way it should be, click the **OK** button to return to the compile overview screen. You updated preview tiles should now be listed in the centre column. **Cancel** will discard all of your changes.

Keeping Preview Tiles Tidy

In projects that feature a large number of section types and layouts, you might be able to trim down how many preview tiles you have to scroll through in order to get an overview of the document look. The order of items (in the section types project settings tab) can optimise their display in the preview tile area in the compile overview. Adjacent types that use the same layout will be grouped together as one tile instead of each having their own individual preview tiles. For example if we have three section types called “Definition”, “Glossary Entry” and “Figure Reference” all using the same section format called “Hanging Title Block”, then you would see that layout preview tile once in the overview screen.

[Return to chapter](#) ↗

23.4 Compile Settings

Returning to the main compile overview screen, the right-hand column is separated into several different tabs, where you will set up information about the work you are compiling:

- I. The first tab, starting from the left, is the Contents tab ([subsection 23.4.1](#)), where you can adjust which pieces of the binder to include in your compiled document, and as well adjust how they may be formatted by surveying section types. By default this pane pulls from the “Draft” folder with no other adjustments. If you’ve simply written your work into that folder and are expecting to print it in entirety, you can probably ignore this tab.

2. The second tab is where you will set up metadata ([section 13.5](#)), such the title of the work, the author(s) and so on. This panel also provides technical access to extended metadata for eBook and Markdown-based formats.
3. The General Options tab ([subsection 23.4.3](#)) contains all of those fiddly settings you might need—such as whether footnotes should be at the bottom of each page or gathered together as endnotes, if proofer markings should be included or stripped out, and so on.
4. The Project Replacements tab ([subsection 23.4.4](#)) can be thought of as a list of search and replace commands that only modify the compiled document rather than the original.
5. The Cover tab ([subsection 23.4.5](#)) is where you can set up the cover image and catalogue thumbnail used by eBook formats.
6. eBook files also come with an automatically generated table of contents page by default. The Table of Contents tab ([subsection 23.4.6](#)) is where you can adjust whether and how this element should appear in the final book.

All of the options in these tabs are saved into the project automatically whenever you compile, or if you save your settings without compiling. They cannot be transferred to other projects, so if you strike upon a combination of settings you would like to use as a starting point in the future, consider creating a project template ([subsection 5.3.3](#)).

23.4.1 Contents Tab



Figure 23.3: The “Contents” compile settings tab.

The Contents pane is used to establish which parts of the project will be used to create the compiled document. The component with the most immediate impact is the **Compile** dropdown at the very top of the list, marked (a) in [Figure 23.4](#). This sets the compile group, from which all of the other options operate as a basis (this is also used to determine targets and statistics in many cases). By default the setting will have the “Draft” folder selected.

In cases where you want to export only a portion of the book, or are working in a project that includes several editions or articles located in the draft folder, you can use this dropdown to select only a portion of the draft. The selection will include not only the container you select but all of its descendants. For example, if choose the folder called “Part I”, all of the chapter folders within that part would be included in the list below, along with any section or subsection files within those chapters.

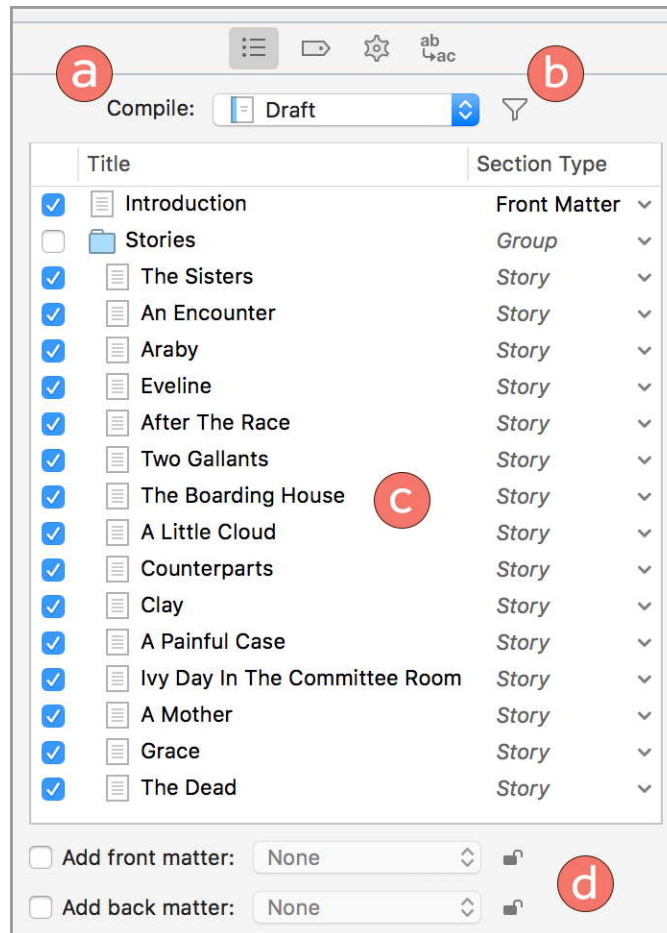


Figure 23.4: The compile contents list is used to establish the material that will comprise the final document.

In addition to selecting a subgroup of the draft folder, there are a few special selections at the bottom of this menu worth notice:

- *Current Selection:* the selection from the active group view you were using prior to entering compile will be used. If two cards are selected on a corkboard and you enter compile, those two cards will be provided as the compile group with this setting.

Your selection can also be used as a filter, discussed in the following section ([section 23.4.1](#)). The distinction between the two is that a filter will act as a search against the selected compile group, while this setting establishes the compile group (and thus can then be further filtered).

- *Search Results & Collections:* the contents of the selected collection or the current search results list will be used to populate the compile group below.

This will always result in a flat list, since these are flat lists themselves. As with selections, collections can also be filtered.

Compile Group Options

Depending upon the type of choice made with the **Compile** dropdown menu, secondary options may appear at the top of the content list.

Two additional settings will be provided whenever an individual container from the “Draft” folder is selected:

1. **Treat compile group as complete manuscript:** ordinarily when a portion of the draft folder is selected for compile, counter numbering will be displayed as though the rest of the manuscript existed; chapter 13 will remain 13. When this option is enabled, the smaller portion you selected will be treated as though it were the entire manuscript. All counters (either in compile settings or in the draft itself) will start at 1.

This is thus useful when hosting several complete and self-contained works in the same draft folder, and using this setting to switch between them. It can also be used to speed up the compiler if you are just proofing and aren’t worried too much about the numbers being accurate.

2. **Include text of containing group:** enabling this adds the selected container to the compile group. This can be useful if the container itself is meant to generate meaningful information, like a chapter heading, page break or introductory text.

When using the “Current Selection” compile group, you will be provided with an option to **Include subdocuments**. With this, adjust whether your selection should automatically include everything *beneath* the selected items, too. This is especially useful if you wish to compile two folders with many subdocuments. You can just select the two containers, open Compile, tick this option and be done with it.

Content Item List

The item list is the large table in the middle, marked (c) in [Figure 23.4](#), displaying the contents of the current compile group. In the provided example the “Draft” folder has been selected, and so the entire Draft contents are revealed in the list below it. The items in this list will be indented just as they are in the binder, hierarchically.

There are three columns within the table. It is usually a good idea to initially scan these columns to make sure everything will act in the manner you expect it to:

Include Unlabelled in the list, this is the first column and contains checkboxes that correlate to the inspector option, **Include in Compile**. When an item

is unchecked it will not be used in the final product unless settings are altered in the filter popup (section 23.4.1). This is generally used to create static exceptions for items which will rarely or never be a part of the compiled product. For quickly filtering or selecting the scope of a compile to for example print out one or two chapters, it will most often be more efficient to use the compile group selector or filters.

Title The visible name of the item in the Binder. This may be used in the compiled output depending upon the format settings for its corresponding section type. For example, it might be that folder names are included as part of a chapter or part break.

Section Type Displays the type of document the associated item is, and allows you to change that association right in the contents list—as described in Applying Section Types Manually (subsection 7.6.1). This often determines how the item will be formatted, as previewed in the centre column of the compile overview screen.

If you find yourself changing many of these by rote, it might be a good idea to adjust how these are automatically assigned in the Project Settings: Section Types tab (section C.2).

Bulk changes can be made within this list:

- Hold down the **Opt** key when clicking on an “include” checkbox to toggle all boxes in the same manner (on or off). When done within a selection, only the items within the selection will be toggled.
- Individually or with multiple items selected, right-click to change either the inclusion or section type assignment.

Upgrading from Scrivener 2

Looking for the old **Page Break Before** or **As-Is** checkboxes? These options have been removed from the software entirely, as what they provided is now served by assigning types of documents to section formats that generate page breaks on their own, or display content as-is, as part of their design. This is a logical change in that if something should be generating a page break, it probably ought to have a section type appropriately set for it, like “Chapter” or “Front Matter”. Meanwhile whether a section of text should print as-is, like an interlude between chapters, should perhaps be referred to as an “Interlude” and be set to format “as-is” in the centre section layouts column.

Filtering

Filtering makes it possible to supply certain criteria by which the compile contents list will be modified. For example you can have it only include those items which have a “Red” label set to them, or conversely, remove all items from the list marked with “Red”. Start by clicking the filter button, marked (b) in [Figure 23.4](#), alongside the main **Compile** group dropdown.

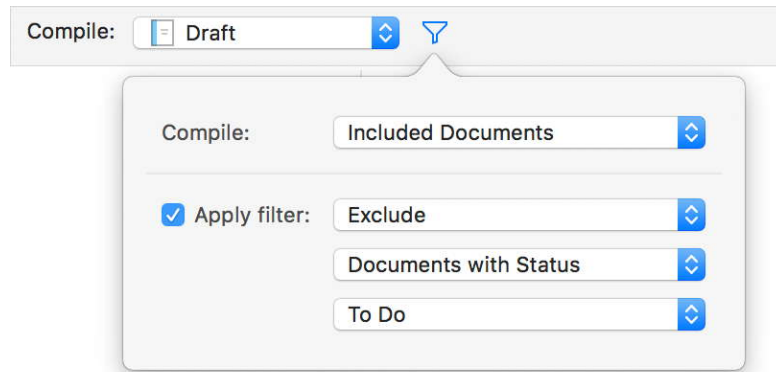


Figure 23.5: The “Filter” button reveals options for further refining the compile contents list.

If when you first load up compile and are puzzled by an empty list or one that appears to be missing chunks of text, check the Filter options and make sure nothing has been left set there from a prior session.

At the top of the popover you will be provided with a choice for how to treat the **Include in Compile** checkboxes, controlled either by checking items off in the Contents list right here, or in the inspector and outliner in the main project window. There are three choices available:

- *Included Documents*: this is the default and logical behaviour. If something is checked it will be included in the compiled document, if not it will be excluded.
- *Excluded Documents*: all documents ordinarily set to be included will be left out, and only those excluded documents will be compiled. If you use this feature to keep notes alongside your main text this can be a useful way to only export those notes.
- *All*: the checkbox will be disregarded outright and all documents shown in the compile contents list will be included.

To enable further filtering, first tick the **Apply filter** checkbox. Filters can be defined as either “including” or “excluding” matches, set with the first dropdown menu, the effects of which can be seen illustrated in [Figure 23.6](#), with the leftmost frame showing the unfiltered contents list:

- *Include*: Everything matching the filter settings will be included in compile, non-matching items will be removed. This is the default setting. Refer to the middle frame in the figure to see what this setting looks like, when choosing the “Red” label to filter by.
- *Exclude*: Everything matching will be *removed* from the compile. The right frame in the figure demonstrates a list with all “Red” items excluded.

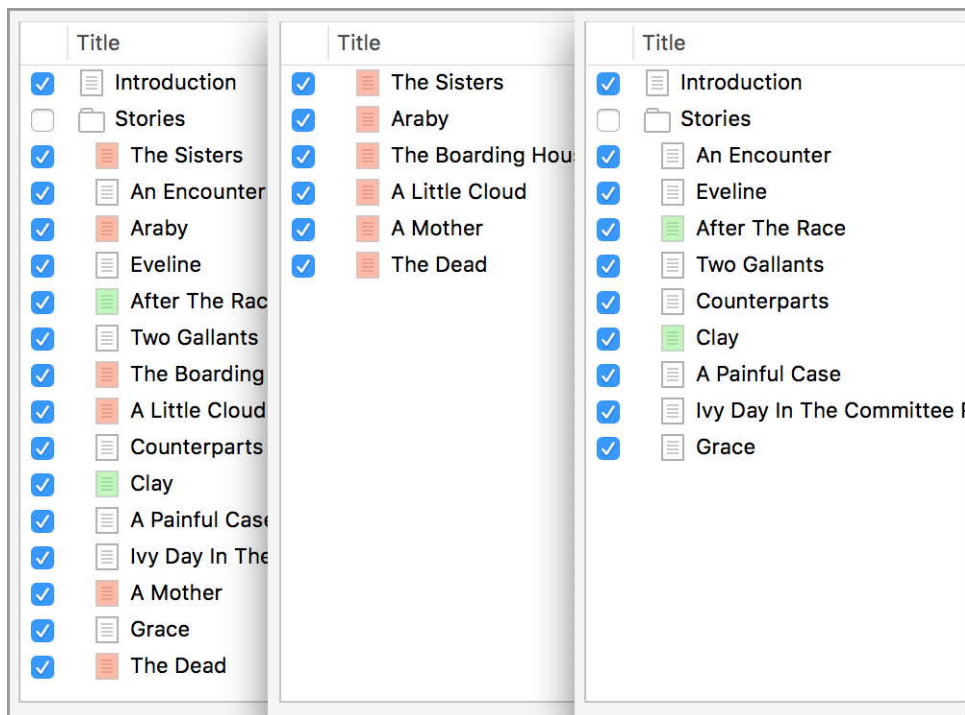


Figure 23.6: Filtering by “Red” produces either a list of only red marked items (centre) or a list with no red marked items (right).

The second dropdown specifies which type of attribute or metadata to filter by. There are four core options always available:

- *Documents with Label*: documents matching the specified label will be matched and handled according to the logic in the first dropdown menu. This is the default setting.
- *Documents with Status*: as above, only using the status metadata field.
- *Documents in Collection*: items found in the specified collection will be included or excluded.
- *Documents in Current Selection*: as with the content selection item, this will use the current selection that has been made in the sidebar, prior to open-

ing the compile interface. Since this method has no optional behaviour, the third selection dropdown will be removed.

- Below these core choices, any custom metadata found within the project that is of either a List of Checkbox type will be provided for you.

Lastly, a third dropdown menu will appear at the bottom of the filter popup if relevant. The contents of this will vary depending on the choice made above:

- Label, status and list-style custom metadata will provide you with all of the possible assignments that can be made with these fields. For example, with default settings the “Label” choice would include selections like “Red”, “Orange” or “No Label”. The “Status” choice would provide options such as “First Draft” and so on.
- Checkbox-style custom metadata will give you a “Yes” or “No” choice.
- When filtering by collection, a list of every collection in the project, whether they be standard or generated by search results. The basic Search Results list is also available from this menu. When using these dynamic settings, one could see a different result every time they compile.

Front & Back Matter

The main matter (or body matter) is a publishing term for the core content of a book, from the first chapter to the last. Front and back matter are defined as the material preceding and following the main content of a book. Front matter (also known as preliminaries or just prelims) will typically includes everything from the title page and copyright page to the preface and introductions. In print publishing, this often includes a different header and footer style, such as Roman numerals for the page numbering. Back matter (also known as end matter) traditionally includes such things as epilogues, glossaries, appendices, bibliographies, author bios and advertisements.

As it becomes increasingly important to be able to deliver material in multiple formats, swapping out these peripheral materials dynamically depending on your compile format may be necessary. You could need, from one single project, a PDF delivered to a Print on Demand service, an eBook published to Amazon or Kobo or even a proofing instruction page for your editing and proofing team.

The specifics of these can vary in that with print formats, you might want a table of contents that uses page numbers, but in the eBook you won't have page numbers, or you might not even want a table of contents at all. With back matter, different sets of links to store pages for buying additional novels from the author might be needed depending on whether the book is being uploaded to Amazon or Nook.

The front/back matter features will be only work when the following conditions are set in the Contents tab of compile overview. In all other cases the out-