User Manual for macOS

Scrivener

Revision 3.0.3

SCRIVENER 3 USER MANUAL
for macOS

COLOPHON
Book design by Ioa Petra'ka, Literature & Latte; concept and cover art by Purpose.

This manual has been typeset using the Calluna font for body text, Effra for headings and interface labels, and Bauer Bodini for chapter numbers and epigraphs.

June, 2018
Revision 3.0.3-01e

Formatted for LaTeX
by MultiMarkdown &
Scrivener

Contents

Part I

Introduction

> The maker of a sentence launches out into the infinite and builds a road into Chaos and old Night, and is followed by those who hear him with something of wild, creative delight.
>
> **Ralph Waldo Emerson**

Philosophy

1

The word processor, that staple of writers' tools, is the digital age's equivalent of the typewriter. And as with the typewriter before it, the word processor is a great tool for typing up a letter, for writing from a plan created somewhere else, or for hammering out words to see where they lead you.

Ultimately, however, a word processor assumes a predominantly linear approach: you write the beginning first, the end last. You enter text, cut and paste, perhaps even work with a basic outline, but the true focus is on presentation, on producing a professional-looking document. The word processor offers little in the way of the foundations on which writing is done—the planning, the research, teasing out an effective structure.

In this sense, Scrivener is the opposite of a word processor, because Scrivener's focus is on how you put together a long document rather than on how it is presented. It is the digital equivalent of a writer's studio, incorporating not only the typewriter, but also the notebook, the index cards and corkboard, the jotted plan, the scrapbook and the folder of research.

Scrivener was built to bring together the various tools of the working writer in ways that only software can. Because this is where software comes into its own—taking tasks that are disparate in the real world and integrating them. To take just one example, in the analogue world, if you move index cards around on a corkboard, you also have to reshuffle the sections those cards represent in your manuscript. In Scrivener, the corkboard and manuscript are integrated, so that moving a card on the corkboard moves the associated section in the manuscript and vice versa.

Every writer sneaks up on the blank page differently; accordingly, Scrivener is designed to be flexible, to adapt to the writers's workflow, not the other way around. It works equally well for writing a novel, a doctoral dissertation, a short story, a screenplay—or anything else. It doesn't prescribe any particular structure or approach. Fiction writers often debate the merits of plunging in and writing versus planning everything before typing a single word, but Scrivener is agnostic and allows either approach—or a mix of both.

The core concepts of Scrivener are these:

— Its editor should be familiar to anyone who has used a word processor.

— You can break the manuscript down into sections as large or small as you like, and work on it in any order.

— You can view sections as discrete chunks or in context with other sections. In a novel, for instance, you could view a scene on its own, in the context of its chapter, or in the context of the whole novel.

— Each section is associated with an optional synopsis, to indicate what the section is or will be about.

— You can step back and see all the synopses in an outline or on a corkboard to get an overview of the whole manuscript, or of a single chapter.

— A writing project is essentially a digital ring-binder in which you can store and view images, web pages, recorded interviews and other research alongside your writing.

— You can view more than one document at the same time: refer to notes alongside a chapter, or bring up an image alongside the description you are writing.

— What looks great in print is not always best for the screen, so you can choose a different format for your exported or printed work without affecting the original text, and you can tailor the formatting for a particular output. An eBook can be formatted one way, a printed manuscript another.

The underlying philosophy of Scrivener was in part inspired by a passage written by the author Hilary Mantel in a collection of essays by writers on the process of writing entitled *The Agony and the Ego*. Hilary Mantel described a process of "growing a book, rather than writing one," which can be summarised thus:

1. During the first stage of writing, you might jot ideas down on index cards—phrases, character names, scene ideas; any insight or glimpse.
2. When you have gathered a few index cards, you might pin them to a corkboard. Other ideas build around them, and you might even write out a few paragraphs and pin them behind the index card with which they are associated. At this stage, the index cards have no definite order.
3. Eventually, you may begin to see an order emerging and re-pin the index cards accordingly.
4. After you have gathered enough material, you might take all of your index cards, sheets of paper and jottings and place them into a ring-binder. You are still free to move everything around, but now you have a good idea of how much work you have done and how much more work you have to do.

Scrivener, then, provides an environment in which you can "grow" your work organically—an environment which in turn grows with your writing, to accommodate the multifarious stages of composition.

About This Manual

2

Scrivener has a wide variety of features to accommodate many different purposes, including novels, screenplays, academic papers and dissertations, general non-fiction, journalism, blogging, and much more. While it strives to present as simple an interface as possible, once you start digging into the application, you will find a degree of flexibility and complexity to suit even the most esoteric needs. To help organise all of these concepts and features, the manual itself has been split into four primary sections including an appendix. In each section the features most useful to you during those phases of your real-world project will be explained in depth:

1. Introduction (Part I): the first section goes over basic installation and updating notes, and then introduces several fundamental concepts regarding Scrivener's interface, how it stores your work into "projects", and how your work is organised within those projects. This will be essential reading for anyone new to the software, and an index into the various elements you'll see on screen as you learn.
2. Preparation (Part II): next, we'll take a deeper look at all of the core organisation and management features of Scrivener. It could easily be said that Scrivener is as much about organising your writings (and the material you use to help write them) as it is about getting the actual typing done. You may not need all of the topics we'll go into in this section, but if you have a question on how the program is meant to work at the higher levels of usage, chances are we'll be going over it in here.
3. Writing (Part III): beyond the obvious aspects of working in the text editor, we'll also go into the many other activities that must be done while writing, such as working with images and inserting them into your manuscript, annotating your work with comments and footnotes, formatting it, handling revisions and so forth.
4. Final Phases (Part IV): getting your work out of the software and into the hands of your agent, publisher, readers or wherever it needs to go next, will be the final practical focus of the manual. We'll also discuss options for exporting files and printing copies.
5. Appendices (Part V): last but not least, we'll go over every single menu command, preference and miscellaneous reference you may need. These sections can also serve as a large topical index, as you can easily look up more advanced or in-depth topics by starting with what you've found while exploring the interface.

Upgrading from Scrivener 2

If you've been using Scrivener for a while, you'll also want to check out our What's New (Appendix E) appendix, as a few things will have moved around or require your attention when you first upgrade your work to the new version. You can also keeps tabs on new developments in Updates to Version 3 (section E.12).

This manual has been written using Scrivener (yes, we eat our own dog food!) and is available in a few different formats from our web site[1]. It has been written using the MultiMarkdown formatting syntax and so demonstrates that system as well.

Annotating this PDF

If you wish to make notes and annotate the PDF using software such as Adobe Reader, it is recommended you download a separate copy from the above linked page, or open the PDF from the Help menu and drag the icon it the window's title bar into your binder, or to Finder with the Option key held down to create a copy. When Scrivener updates, it will very likely overwrite the existing PDF in the installation (often with revisions to the text), which will destroy any of your notes.

2.1 Terms & Conventions

Some features apply only to the standard version of the software, and others only apply to the Mac App Store version. Various key features will be marked to indicate this. If you have purchased the program directly from our web site, then you have the standard version. If you used Apple's App Store tool to buy Scrivener, then you have the Mac App Store version. Sections applicable to only one or the other will be indicated with the following markers:

— Standard retail version of Scrivener: ⟨Direct-sale only⟩

— Mac App Store version of Scrivener: ⟨MAS only⟩

Similar markings will be used to indicate when specific versions of macOS are required for a particular feature to be available.

2.1.1 Interface & Menus

Whenever the documentation refers to an interface element that can be interacted with, such as a button, the visible name for that element will be formatted

[1] https://www.literatureandlatte.com/learn-and-support/user-guides

in dark red, such as "Click on the OK button to save changes". Button labels, menu items, and keyboard shortcuts will all be displayed in this fashion.

Menus will be displayed in a hierarchy using the " ▸ " character to separate top-level, submenu, and items. Example: To convert a range of selected text to uppercase, invoke the Edit ▸ Transformations ▸ Make Uppercase command. Some menu commands change their name depending on usage. The parts that change will be indicated like so: View ▸ Text Editing ▸ Show|Hide Invisibles.

Difficulty Seeing the Labels?

We have prepared an alternate version of this project[a] that may increase the visibility of hyperlinks and interface labels in this project, for those with red/green colour blindness.

[a] http://www.literatureandlatte.com/downloads/scrivener-mac-user_manual-altcolour.pdf

Some of the names for various elements within Scrivener are customisable on a per project basis, and how you name these will impact much of the interface. A good example is the "Label" setting, which can be used to colour-code your work, and what it is referred to can be changed in settings (to something like "Point of View" or "Rewrite Status"). What you name it will impact the names of menu items and other bits of interface that refer to it. In all cases, this documentation will refer to these malleable elements by their default names.

The names of features will be capitalised as proper nouns if it is necessary to differentiate them from standard nouns. A *Collection* is container for organising loose documents, while the word "collection" can be used to indicate a casual grouping of items and not necessarily a formal Collection.

2.1.2 Keyboard shortcuts

Keyboard shortcuts will use the following symbols:

— ⌘: The Command key, or the Apple key, is the one located directly to the left and right of your spacebar.

— ⌥: The Option key is also labelled the Alt key on some keyboards, depending on which country you purchased your Mac from. Some laptops only have one Option key on the left side.

— ^: Control is usually located to the left and right of the Option keys on their respective side. Some laptops only have one Control key between the Option key and the Fn key on the left side.

— ⇧: The Shift keys are rarely used by themselves in shortcuts but are often used in combination with other modifier keys.

— The arrow keys on your keyboard will often be shortened to the four symbols: ↑ ↓ ← and →.

When a keyboard shortcut requires a combination of modifier keys, they will be printed together. Example: ⇧⌥⌘V, which matches Edit ▸ Paste and Match Style, means you should hold down all of these modifier keys together and then tap the V key on your keyboard.

The Mac distinguishes between the Return key and the Enter key in some contexts. Specifically to Scrivener, we will point out the few places where using one key over another will make a difference. If you are using a compact or laptop keyboard, you may need to use the Fn-Return key combination to press the Enter key.

2.1.3 Preference labels

Scrivener has many preferences, and as a result they are organised into a hierarchy starting with a major category along the top of the window (like "Behaviors"), and sometimes into a secondary tab, either along the top or if there are several sub-categories, along the left in a list. To refer to a specific category then we will say "Behaviors: Navigation". If a subcategory has further tabs within it, then we'll print something like, "Appearance: Main Editor: Fonts".

The labels used in the interface to denote settings will be marked in the following fashion: the **Header Bar** is a font selection tool located within the aforementioned preference pane.

2.1.4 Filenames & Paths

In cases where file paths are printed, the UNIX convention of providing a shorthand to describe your personal home folder will be used. An example might look like:

```
~/Documents
```

The tilde is a shorthand which means: `/Users/yourusername`. In this case, the path refers to the Documents folder in your home folder.

2.2 Finding Things in this Manual

Since this PDF has been birthed within the age of digital documentation a proper index has never been compiled for it. Despite this, in practice you should have little difficulty in locating the topic you are interested in. Modern PDF reader software features excellent searching capabilities; most things can be discovered merely by searching for the proper names of things as labelled in menus, buttons or dialogue boxes.

Alternatively, the appendices have been written to be used as a sort of topical index. If you have a question about a particular menu command, for instance, you can find it in the appendix, Menus & Keyboard Shortcuts (Appendix A), or Preferences (Appendix B). Often, if the feature merits it, there will be a cross-reference to a more thorough description of the feature in the main manual. When all else fails, the table of contents has been provided at the beginning of the PDF to get you roughly where you need to go, and where chapters themselves go into depth on several topics, they will feature a secondary table of contents after their title page. Additionally, if you are using a PDF reader with a contents sidebar feature you will find that to be a much more detailed table of contents than the one in the front of the book.

2.3 Spot a Problem?

A user manual is a living document. It evolves constantly as the software itself evolves, and as a result it can oftentimes be difficult to keep edited to the same calibre we'd expect of a finished and published book. If you spot a problem, or have a suggestion that would make this a better resource for you, we'd love to hear about it. Drop us a line at contact@literatureandlatte.com.

Installation & Upgrading

3

3.1 Installation

⟨Direct-sale only⟩ If you purchased Scrivener through the Mac App Store (MAS), then this section will not be relevant to you, as it pertains to the registration, installation, and maintenance of the standard retail version. When purchasing software through the MAS, installation and maintenance of the software is generally handled for you. If you require assistance installing or updating Scrivener through the Store, consult Apple's documentation on the matter.

1. If you have not already downloaded the trial version from the Literature & Latte web site[1], do so now. The trial version can be unlocked at any time with a matching registration serial number that will be sent to you upon purchase of the software.
2. When the installation DMG finishes downloading, double-click on it in the Finder (if it hasn't opened for you automatically), and drag the Scrivener icon to your Applications folder.[2]
3. Eject the installation DMG by use the File ▸ Eject command in Finder on the installation window.
4. Open the Applications folder in Finder and double-click the Scrivener icon to launch the software. **If you drag the icon to your Dock, always do so from the Applications folder, not the installation DMG.**

If you attempt to run Scrivener from a place that resembles the installation DMG, you may get a message offering the chance to have it installed for you.[3] If you intend to run Scrivener from outside of an Applications folder, then you should check the box that will inhibit this warning from appearing again in the future.

3.1.1 The Trial Version

You can try out all of the features of Scrivener for 30 non-consecutive days without having to pay or register. During that period, Scrivener will be **fully functional**. After 30 days of use (and one grace session so you can export your work), you will no longer be able to access Scrivener at all unless you register. If you are getting close to the end of your trial and have decided to not purchase Scrivener, please skip forward to the sections on Compiling the Draft (chapter 23) and Exporting (chapter 25), for details on getting your data out of the program.

---

[1] http://www.literatureandlatte.com/download?product=Scrivener

[2] If you lack permissions to install software into the main system Applications folder, you can create an Applications folder in your home folder and drag Scrivener (and other software) there.

[3] In technical terms, that means a removable volume or a disk that is read–only. If either of those conditions exist the warning message will be triggered.

During the trial period, whenever you launch Scrivener, you will be reminded of how many trial days you have left and given the option of buying and registering.

3.1.2 Purchasing Scrivener

Should you choose to purchase the software, registration will be a seamless process of unlocking the copy you've already downloaded, meaning all of your settings and work will be right where you left them. Use the Help ▸ Purchase Scrivener... menu command to navigate to the Literature & Latte online store.

When you buy a licence for Scrivener online, you will be emailed your unique serial number, which will have the following format[4]. Make sure that you keep this serial number, along with the exact name under which you registered Scrivener (the "Serial Number Name"), in a safe place, as you will need both to register Scrivener again in the future.

```
SCRIV3MAC001-XXXX-XXXX-XXXX-XXXX-XXXX
```

If you have lost your serial number, clicking the Retrieve Lost Serial... button in the registration window will take you to our vendor's self-service support site, where you can request to have the information sent to your email address again. If for some reason you cannot get that to work, please contact us on our support page[5].

3.1.3 Registering Scrivener

After purchasing a licence and receiving your unique serial number, you can register Scrivener by clicking on the Enter License... button in the demo window that appears whenever you launch Scrivener, or by using the Scrivener ▸ Registration... menu command.

1. In the "Serial Number Name" box of the registration window (Figure 3.1), enter the exact name used to register Scrivener into the first field.
2. Copy and paste the unique serial number in the "Serial Number" box, making sure that the hyphens are included as indicated, and that there are no unwanted spaces at the beginning or end of the pasted text.
3. Click the Register button.

You must make sure that both the user name and serial number are exactly as they appear in the registration email you received, or registration will fail. If you receive a message stating that the name or serial number is invalid:

[4] If the initial block of letters is different you have an old serial number and will need to upgrade from our web store, or find a newer invoice with the appropriate number.

[5] https://www.literatureandlatte.com/learn-and-support

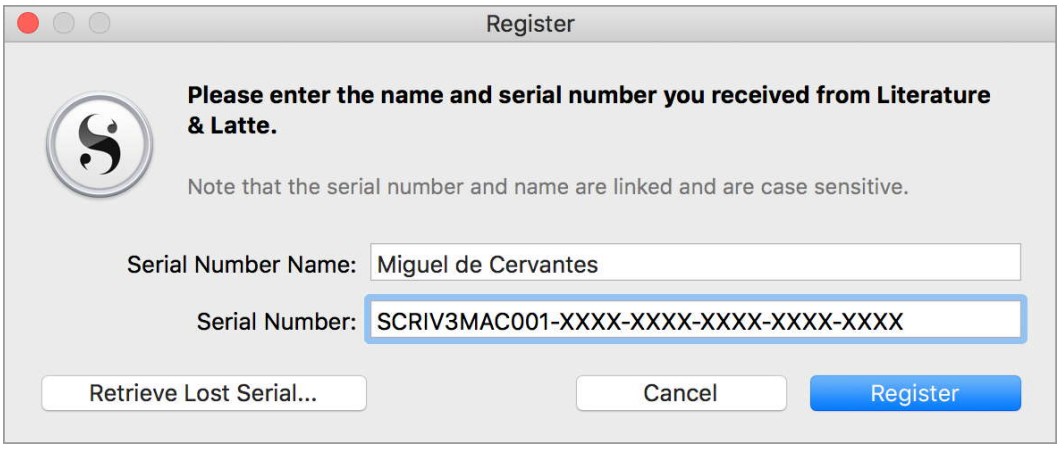


Figure 3.1: The registration panel for unlocking a copy of Scrivener

— Check and make sure they are in the right order. The name of the software owner should be in the top field; the serial number in the second field.

— If copy and paste doesn't bring over the hyphens you could either try copying from the original email, which won't be in PDF format but plain-text, or if all else fails type it in by hand.

After clicking the registration button, Scrivener will attempt to authenticate your copy over the Internet. No private information will be relayed to the vendor's server. If for some reason the Internet cannot be accessed from your computer, you will receive a warning message with instructions on how to activate the software manually. Follow the provided instructions to complete the activation either on another device, or using a browser on your machine. You can use copy and paste into the forms to make this easy.

Once Scrivener has been registered, you can begin using it immediately. There will no longer be a time limit on its use and you will no longer see the nag box at startup.

3.1.4 Application Updates

The second time Scrivener is launched you will be presented with a panel asking whether or not Scrivener should automatically check for updates. To use this feature, you must have a connection to the Internet.

— Click the Check Automatically button to have Scrivener perform a daily check to see if there is a newer version available and will prompt you to update if one is found.

— Click the Don't Check button to disable automatic checks. You can always manually have the software check for an update by using the Scrivener ▸ Check for Updates... menu command.

You can change the automatic update settings in preferences, under the General: Startup tab, **Automatically check for updates**. Once enabled, adjust how frequently the software checks in with our server to see if there is a newer version.

When an update is detected, a window will appear with information about the update and some buttons to install or defer the update. It is recommended you brief yourself with the update notes, as sometimes changes in operation occur.[6]

— Install Update: download the newer version of Scrivener and have it installed for you automatically. Note that you can work while it is downloading, but the software will need to restart at some point to implement the upgrade.

— Remind Me Later: dismiss the update panel and return to your work. You will be notified again the next time an update check is performed.

— Skip This Version: dismiss the update panel and return to your work. You will never be notified about this specific version number again, but you will be alerted of future updates beyond the current one.

Automatically download and install updates in the future If you leave this option checked when clicking the Install Update button, then Scrivener will never bother you about updates in the future. Instead it will start downloading them as soon as they are available, and will inform you of when the download is complete so that you can have it install and restart the software.

Software vs Your Work

Unlike mobile systems, standard operating systems don't conflate your work with the software icon and there is no connection between the two. Computers are considerably more robust and well designed for content creators. Like all programs, Scrivener saves your work into files separate from the software. Upgrading versions, registering your trial, or switching to a beta build will have no impact on your data (save that you of course will not be able to open it so long as Scrivener is not installed, and that it will be accessible once the software is available again).

3.1.5 Portable Installations

We do not recommend installing Scrivener on a portable drive that you routinely remove from your computer. It is best to install and register Scrivener on *each* machine independently and keep your data portable. In most cases this can be

[6] You can view the full list of changes at any time on our web site[7].

done with a single user licence, but you should check with our licensing details to verify[8].

3.2 Setting Up Your System

For most uses, Scrivener will be ready to go as soon as you install it. However if you're interested in some of the optional extras it provides, this section will describe how to set them up.

3.2.1 Setting Up System Services

Scrivener comes with a number of system-wide Services that you may use to more easily capture information from other software. As with all Services, they can be configured in System Preferences: Keyboard: Shortcuts, within the "Services" pane. You will find all of our services located within the "Text" subsection of this pane. Read more about Scrivener Services (section 9.3).

3.2.2 Use of System Contacts

When you first create a new project (such as the tutorial) you will be asked by the operating system if it is okay for Scrivener to access your Contacts. This is a standard request for permission that is generated by all programs that use your contact list. Scrivener only uses this access to generate your name into page headers, for "signing" comments as well as generating cover sheets and the like when creating templates. If you would rather not grant Scrivener the ability to do this, you will need to fill out these details yourself.

Using a pseudonym

By default, Scrivener will look for your Mac's default contact card, found in Contacts.app, under the Card ▸ Go To My Card menu command. If for any reason you require the use of another name than what your Mac is set up to, you can have Scrivener use different contact information entirely, in the General: Author Information preference tab (subsection B.2.3).

3.2.3 Installing Extras

⟨MAS only⟩ If you are using the MAS version and wish to use the "Print PDF to Scrivener" feature from print panels Mac-wide, you will need to install the alias yourself:

[8] http://www.literatureandlatte.com/licence

1. Open two Finder windows, one pointing to the Applications folder where Scrivener is installed, and the second to the "PDF Services" folder in your user Library folder.[9]
2. Drag the Scrivener icon from the Applications window to the PDF Services window. This will create an alias (the icon should have a curved arrow in the corner of it where you dropped it).
3. Rename the alias in the PDF Services folder to something like "Save PDF to Scrivener".

It is also not possible to install the custom Scrivener colour swatch group with the MAS version. The Scrivener group will still show up in the palette, but any changes you make to it will be lost after you restart the program. To make permanent changes, you'll need to create your own group called "Scrivener Custom". The best way to do this is to rename the Scrivener set to "Scrivener Custom", using the gear menu to the right of the palette selection menu. This will create a persistent colour swatch group for you, preloaded with all of Scrivener's built-in colours. Refer to Naming Text Highlights (subsection 18.5.1) for more information.

3.2.4 Setting up MultiMarkdown or Pandoc

⟨Direct-sale only⟩ Scrivener comes with MultiMarkdown embedded within it, and all of the extra files necessary to produce documents of any type, directly out of Scrivener. However you may want to update MMD to a newer version if the one we provide is older. Additionally, those using the LaTeX document typesetting system will get cleaner compile results by installing the diverse LaTeX support files that MultiMarkdown uses to build different document types. You will find instructions for downloading installing MultiMarkdown on its website[10]. Scrivener will check for and make use of any version of MultiMarkdown you've installed yourself, so long as it is installed in the standard location (`/usr/local/bin`).

Scrivener also supports a few Pandoc export options. However given the size of Pandoc, we are unable to embed a copy of it within Scrivener, and you will not see any compile options for it, unless you install it yourself on the system. You will find download and installation instructions on the Pandoc website[11]. Once installed and Scrivener is restarted, you should see the Pandoc entries at the bottom of the compile file type list.

---

[9] Hold down the Option key when using the "Go" menu in Finder to get there.

[10] http://fletcherpenney.net/multimarkdown/

[11] http://johnmacfarlane.net/pandoc/

3.3 Staying Informed

If you would like to keep up to date on the latest developments and releases of Scrivener, you can sign up for our low-volume newsletter using Help ▸ Keep Up to Date.... Once you submit the form, a confirmation email will be sent to the address you provided. You will need to click a link within this email before you will be officially added to the list. If you cannot find the confirmation email after 24 hours, check your spam folders, and consider adding "literatureandlatte.com" to your white-list.

The "Keep Up to Date..." window also has links to our Twitter feed and Facebook page. We frequently publish small tricks and tips through these channels.

3.4 Upgrading to Scrivener 3

Major paid upgrades to the software represent large overhauls to the software. You may be alerted to this in the standard update panel, leading to more information on our site, rather than offering to replace your current copy of Scrivener 2. If you wish to merely audition the new version, it might be a better idea to download the demo from our website directly, and run it alongside the older version.

It is our hope that you find major upgrades to be familiar to use in most regards, but if you are a veteran of the software, there will always be new or changed areas of the software worth investigating. The built-in interactive tutorial, found in the Help menu has a guided tour of what has changed, and if you require more detail, please refer to the appendix, where you will find a full guide on modified and added features in Scrivener 3 (Appendix E).

Close to a deadline?

The main thing you may wish to be aware of, prior to upgrading Scrivener, is that you will need to reconfigure the compile settings of your projects. We have made every effort to make the transition as painless as possible, but given the extent of the overhaul that has been done it is not possible to copy all of the settings across for you. So if you are on a tight deadline and have a lot of complicated compile settings in play, it might be best to hold off on upgrading that project until such a time as you have the leisure to review the new options and see how they can implement the effects you were looking to achieve with the older settings.

3.4.1 Running Multiple Versions

While it is possible (and safe) to run more than one version of Scrivener at once on your computer, it is advisable to first delete older versions prior to installing the new version. To do this, make sure Scrivener is closed, then drag its icon

from the Applications folder to the trash in your Dock before following the above installation instructions.

In some cases, you may need to keep an older copy around. When running multiple versions of a program, you'll want to keep the following in mind:

— To open older projects in Scrivener 1 or 2, you will need to drag these items onto their icons, or use the File ▸ Open... menu command from within these older versions. When double-clicking on projects in the Finder they will open in the new version.

— So long as you have both versions installed, the Mac might get confused over the Services, which can be used to clip information from other programs into Scrivener. Once you have fully uninstalled the old version of Scrivener, these clipping services should work just fine after a reboot.

— It is safe to run both copies at once, as they use different preference files, though they will share common resources like project templates and various presets. If it is important to keep some of these distinct, we advise using naming schemes to do so.

For best results, you should download the latest copy of Scrivener 2.9 from the legacy downloads page[12], and use the following instructions. If you have already registered Scrivener 2.9 with your new 3.0 serial number, then you may skip these instructions; likewise if you are running into no issues with activation when launching either version, you may have no need of the following procedure. This is only necessary if you are continually running into activation issues, as both versions do share the same libraries for registration. You will need to be online to do this:

1. Open the Scrivener 2.9 DMG. From the window that will appear, drag the Scrivener icon *not* to Applications, but to a temporary location like your desktop or the Downloads folder.
2. Using Finder, rename the copy you dragged in step 1 to "Scrivener-2".
3. Now drag this copy to your Applications folder. We use a different name so that you can keep Scrivener 3 as simply "Scrivener", in the same Applications folder.
4. Launch Scrivener 2.9 and from the main Scrivener application menu, select "Registration...". You may, if you have already launched and registered Scrivener 3, see some warning messages about activation—we will be fixing these problems with this checklist, so for now just dismiss them.
5. If Scrivener is registered you will see a "Deactivate..." button. Click this, and confirm deactivation.

[12] https://www.literatureandlatte.com/legacy-downloads

6. If Scrivener 2.9 was not registered, or if you've just deregistered it, either way make sure you are at the Scrivener ▸ Registration... window, and click the Enter License button.
7. Enter your serial name and number into the appropriate fields using your new 3.0 information. Scrivener 2.9 (and no lower) will accept this as a valid serial number and unlock.
8. You may need to activate online at this point, but you might not if you've already activated your copy of 3.0.

With both versions using the same serial information, you should run into no problems switching between the two.

3.4.2 Upgrading Projects

The project file format has been updated considerably, and new or upgraded projects cannot be opened in older versions of the software. The first time you load each of your older projects you will be presented with a dialogue box asking if you wish to upgrade the project format. Review the instructions and advice provided in Project Format Upgrades (subsection 5.1.6) for further information on that process.

3.4.3 Recovered Files

Older versions of Scrivener's project format were designed before the concept of synchronisation, or "cloud" servers, became popular. A common method of handling conflicts with these systems is to duplicate files that have been edited in two different places at once. In the past, these duplicate files were left unremarked, and left to accumulate within Scrivener's project format.

Scrivener 3 includes a much more thorough project repair system that runs whenever you open a project, and this will include the first time the project is updated from an older version. The result for some people might be years worth of old conflict files being rooted out, and presented to you as "recovered files" in the binder. You might also be asked to choose between several different binders.

This may be alarming at first, but bear in mind Scrivener is merely going through old problems it is just now detecting. Going forward you will be made more immediately aware of them as they occur.

3.4.4 Saving Your Projects for Older Versions

As noted, once you have upgraded your project to the new format, it will no longer be directly accessible to Scrivener 2.x for macOS, or 1.x for Windows. However it is possible to save a copy of your project in a format that can be read with these older versions, and even later load the modified project back into v3 with minimal loss of new settings and features.

The procedure is simple, but it's important to understand that you won't be able to share the project *directly*. If you've been working through a medium like Dropbox, with all devices and collaborators sharing one single project, you will need to adjust your working habits slightly.

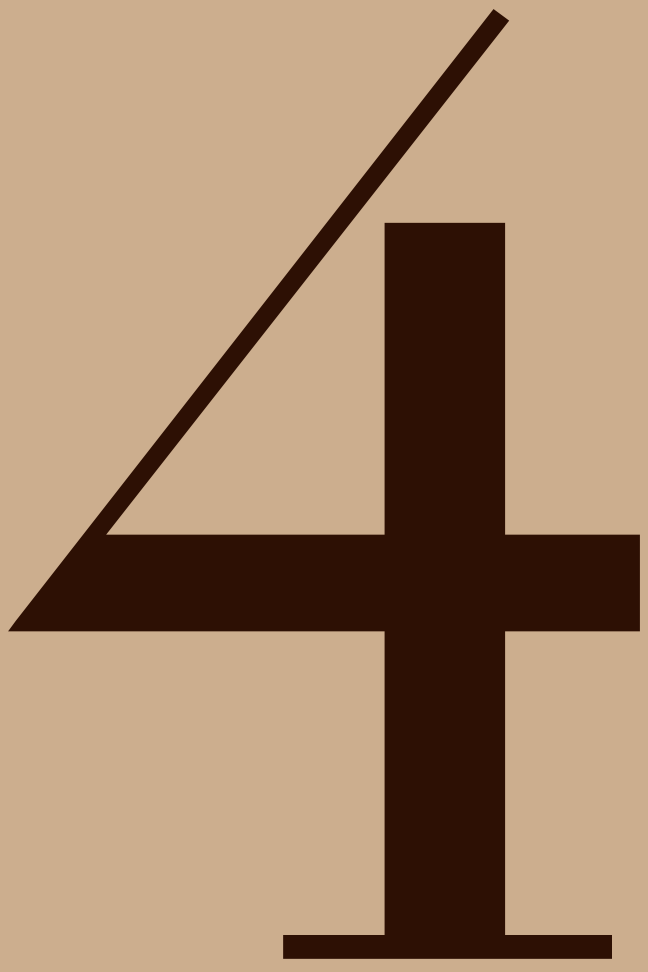
To save a copy of your project in legacy format:

1. Open the project you need to save, and use the File ▸ Export ▸ as Scrivener 2 Project... menu command.
2. Select a location to save the copy (if using the cloud, choosing your shared folder would be a good option here) and select a name for the project.
3. Click Save

This will create a completely new copy of the project in a format that can be edited freely by older versions of Scrivener. *However*, when you wish to load the project back into Scrivener 3.x, it will need to be upgraded once again.

This method will therefore require a little management of project files which will accumulate every time you switch versions—and it also means that with larger projects that take considerable time to fully save and upgrade, it may not be a feasible way of working at all. In such cases, it may be easier to keep a separate copy of the older version of Scrivener (which you are entitled to use and unlock with your version 3 serial number, if purchased directly from us) and simply defer upgrading the project to v3 until such a time that all machines that need to work with the project can read and work in that format.

Interface in Overview



In This Section...

Scrivener's interface has been carefully designed to scale across a wide range of uses. At its most minimal, the program may look no more complicated than a basic text editor like TextEdit; at its most complex, it can fill a multi-monitor workstation with expansive detail into a major project of thousands of components. In this section, we will go over some of the basic interface elements that will be present in nearly every project you work in, as well as the way we'll be talking and thinking about the software. Advanced features will be gradually introduced in their own sections later on as they pertain to specific areas of the writing process. We'll try to point you to those areas as we go along, making this chapter a sort of "index" of the things you can see and make use of in the software.

4.1 The Project Window

When you initially run Scrivener, you will be presented with a Getting Started and project template selection window (Figure 5.1). This will be the launching pad for all of your future projects, and can be used to load existing ones as well. Whenever you are ready to start a new project, whether it be a new novel or

paper for a journal, you'll use this interface to select a starting template, or just the blank start to go with a clean slate (subsection 5.1.1).

You may want to click on the "Blank" category in the left list, and then use the "Blank" starter to create a temporary test project to play around with, while reading this chapter.

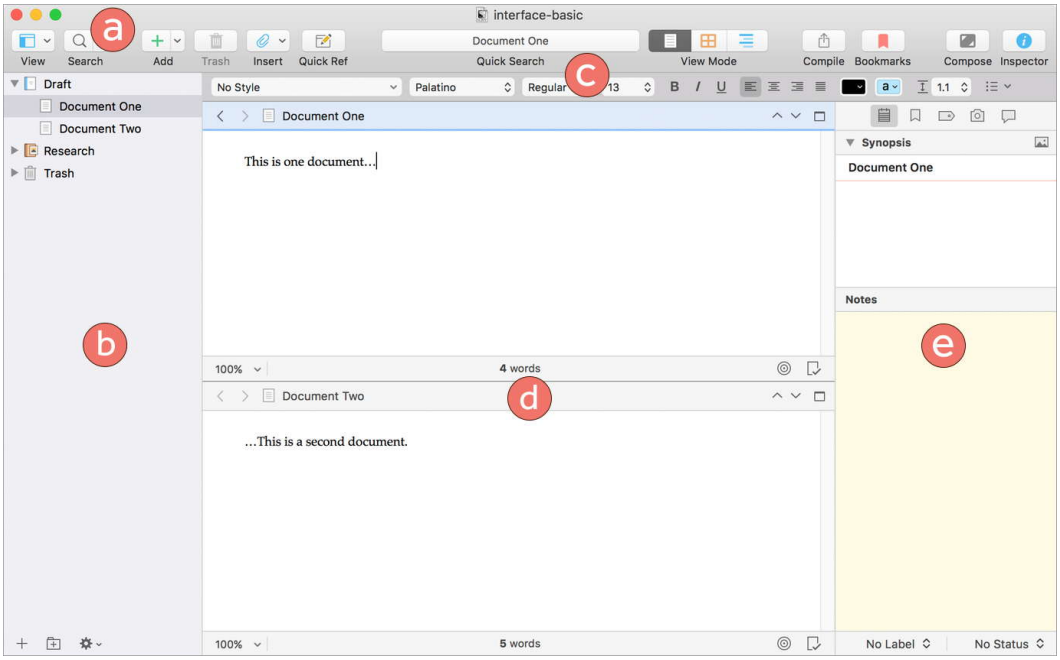


Figure 4.1: The five main areas of the project window.

The main project interface comprises five main elements (Figure 4.1), not all of which are visible when a new project is created. Each of these elements will be explained in greater detail in the following sections.

a) *Toolbar*: frequently used tools displayed in a standard fashion, familiar to most macOS applications, that can be customised as you see fit.

b) *Binder*: the master list of all the documents and resources in your project, including the thing you're writing in the draft folder; a general collection area for research materials, notes, and so forth; a trash can for collecting deleted items. You can add your own folders and organise material freely within them.

c) *Format Bar*: frequently used text formatting tools, available to both editor splits and a few other text editing contexts as well.

d) *Editors*: this multi-use area is where all viewing and text editing is done, from PDF files to audio files to the contents of folders. The editor can be split into two panes (as shown here in horizontal orientation). These splits can show two parts of the same document, or two different items altogether.

e) *Inspector*: a feature-packed tool providing information about the currently active split. The inspector has up to five separate tabs, which address diverse aspects of the viewed resources, depending on their type.

The main interface sections (binder, editors, and inspector) can be resized by dragging the divider line between them, and many of the smaller elements within the sidebars can be resized as well in the same fashion.

4.1.1 Selection & Focus

Selection, and the areas of the project window have focus, are often important in Scrivener. There are features that only work when certain types of things are selected in the interface, or when the cursor is in a particular area of the window—what we refer to as "focus". A good example of this is the Edit ▸ Transformations ▸ Make Uppercase menu command. It has no meaning when a selection of index cards is currently active, as it works upon selected text. If there is a selection restriction on a feature, this manual will often indicate as such using the following terminology:

— *Active*: this relates to "focus", or where the application is currently accepting keyboard input. In other words, if you tapped the 'h' key, that is where the 'h' would be sent. In some cases this might not do much or anything at all; in an active editor it would insert an 'h' letter, and in the binder or outliner it would select an item starting with 'h'.

— *Active selection*: expanding on the above, an active selection is when something is *selected* within the active area of the application. An example might be some selected text in the editor, or three selected items in the binder or corkboard. Active selections are often used as the target for various commands in the menus. An example would be the Documents ▸ Convert ▸ Text to Default Formatting..., which would reset the text formatting for any selected documents in the active selection. On the other hand that command will reformat the contents of the editor window you are working in, if it is active.

— *Inactive selection*: A selection can be inactive. If you select a folder in your binder and then click in the editor, the selection will become dimmer. Inactive selections are seldom used, but there are a few cases, especially when exporting, and as targets for when dropping things onto an inactive selection—such as assigning a keyword by dragging and dropping it onto several dozen selected items at once.

— *Active editor*: the main editor must have focus. In cases where the interface is split into two views, this not only means any editor must be selected, but specifically the editor (or copyholder) you wish to perform the function on. Some commands do not require text to be selected, only an editor, others

will perform universal actions if no text is selected, still others will refuse to work unless you have a selection.

— *Item Selection*: or "selected items", refers to the selection of documents (or "items", more generally) in the binder sidebar, corkboard or outline views in the editor. Item selections are often used in conjunction with commands such as those found in the Documents menu, like "Move To", which would move a group of selected items to another place in the project.

4.1.2 Toolbar

Spanning the width along the top of the project window is a customisable toolbar, with convenient access to some of Scrivener's most used features. To change the order, add, or remove icons, use the View ▸ Customize Toolbar... menu item, or right-click on the toolbar to switch between text, icon and text + icon display modes.

There are four styles of buttons used on the toolbar:

1. Multifunction: these have a downward facing arrow on the right side of the button, and no dividing line between the arrow and the rest of the button—such as the **View** button on the left side. These buttons work just like the main menu in that you can click once to show the respective options and click again to choose one, or click and hold, move the mouse and then release the button on the desired option to select it.
2. Multifunction with a default action: when there is a dividing line between the icon and the downward arrow—such as the **Add** button. The left side of the button can be clicked once to activate the button's default behaviour. If you click on the arrow side of the button, it will act exactly as described above.
3. Simple: the button only has one function and clicking it performs that function.
4. Toggle button (or "segmented control"): you can think of these as being a bit like one single button with more than one setting. You select from one of the provided options by clicking on the associated icon within it. The **Group Mode** button is the only sort of this kind.

It is also possible to hide the toolbar by using the View ▸ Hide Toolbar menu item. All functions present on the toolbar have alternate access throughout the rest of the interface or through menus.

The Default Button Set

We've curated a selection of buttons out of the box for quick access to commonly used functions, and a few tools suited to this prominent placement in the window. We'll go over these buttons in brief, pointing you to additional reading

material if you are curious to read more on the feature itself. You aren't stuck with any of our choices, any of the buttons can be removed and there is a wide selection of alternative buttons available to be added.

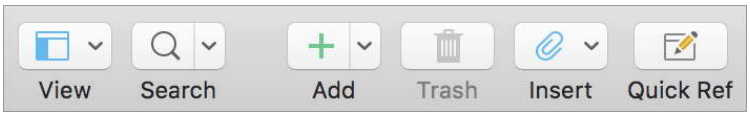


Figure 4.2: Default toolbar (left): view settings, search and item editing.

The left side of the toolbar contains the following functions in two primary groups:

View Quickly toggle a few of the primary organisational tools in the project window and formatting tools in the editor. Also provides access to any saved project window layouts (section 12.3).

All of the options found in this multifunction button can be individually placed onto the toolbar as an additional button. The functions themselves can otherwise be accessed via the View menu, View ▸ Text Editing submenu and Window ▸ Layouts ▸.

Search The default action of this button toggles the display of the project search (section 11.1) field that appears above the binder sidebar when active. The multifunction button provides additional access to Scrivener's numerous search capabilities (sans Quick Search, which has its own button on the toolbar).

Project search and project replace can all be added as optional separate toolbar buttons, and all of these and additional search tools are located within the Edit ▸ Find ▸ submenu.

Add Offers a convenient way to add new items to your binder, or the view you are currently working within. After generic files and folders, available document template files from the project will be listed, followed by any shared document templates. Refer to Document Templates (section 7.5) for more information on these features.

There are many ways to add new items in Scrivener, refer to Adding New Items (subsection 6.3.1) for further detail, and for importing existing material into the project, File Import (section 9.1).

Trash Moves the selected items to the Trash folder (subsection 6.2.3). Also available as the command, Documents ▸ Move to Trash (⌘ Delete).

Insert This multifunction button is only available when working within a text editor. It provides for the insertion of images, inspector comments and footnotes, links and tables into text.

Footnotes, Comments and Tables can be added as separate buttons, and otherwise all insertion commands (and much more) are located within the Insert menu, as well as a number being available from the Touch Bar.

Quick Ref Opens the selected items, or the currently edited text, into separate windows, or Quick Reference panels. You can also drag any icon from the project window or search panels onto this button to load the associated item in a window. Also available for the currently selected item or active text editor as Navigate ▸ Open ▸ as Quick Reference.

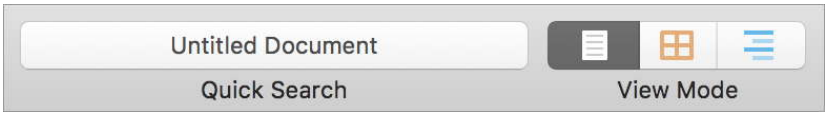


Figure 4.3: Default toolbar (middle): quick search and view modes.

Within the middle group of icons are the following features:

Quick Search Similar to the URL bar in a web browser, this tool displays the name of the document you are currently working on within the editor, and clicking into the tool lets you type the name of the thing you'd like to jump to next. Beyond that, this tool has a few functions worth reading up on if you wish to make more of it (section 11.5).

You can easily move the cursor into this tool with the keyboard shortcut, ^⌥G. It is worth noting that if you remove this from the toolbar, or if you hide the toolbar entirely, that shortcut will bring up this tool as a floating panel that you can either use as needed or leave floating to the side.

View Mode These three buttons toggle between the various view modes that an editor can make use of. In addition to toggling between the three, a view mode can also be *disabled* to reveal the regular text editor at any time, by clicking on the active button. If you've never looked into view modes before, we'll go over those below (section 4.2). From left to right, they are:

— Document/Scrivenings (⌘1)

— Corkboard (⌘2)

— Outline (⌘3)

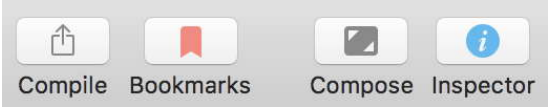


Figure 4.4: Default toolbar (right): compile, bookmarks, composition mode and inspector toggle.

Last but not least, let's take a look at the third and rightmost group of buttons (Figure 4.4):

Compile How you would export your work to a particular format, like a Word file, PDF, Markdown or an ePub. If you're ready to compile, you'll want to head on over to the section on Compiling the Draft (chapter 23), otherwise, there is much to learn before then—and most likely much more to write!

Bookmarks List frequently used documents from your binder, the Internet or your computer here, in the "project bookmark" list (we call it that because each *document* in your binder can have its own list of bookmarks as well). Click the button to load the bookmark list, and drag things into the list to store them. You can also drag things directly onto the icon itself to store them at the bottom of the bookmark list automatically.

While the panel is open, you can tear it off by clicking and dragging anywhere along the panel border. This can be handy if you're using the list a lot and wish to leave it visible above your project for a while. This system is very flexible and augments a number of other features in Scrivener. Read more about Project and Document Bookmarks (section 10.3) to find out more.

Compose Opens the Composition Mode writing view, which blots out the entire interface and lets you concentrate solely upon your words. Click this to load the current editor or selected items into composition mode (chapter 17).

Inspector Toggle the visibility of the inspector sidebar (chapter 13) on the right side of the project window. This can also be done with the ⌥⌘I shortcut.

4.1.3 Left to Right Navigation

Before we take a look at the three main interface components within the project—the binder, editor area and inspector—it would be worthwhile to spend a moment examining how these three elements interact with one another. The rules are very simple, and should work in a manner you may find familiar from other programs:

— A panel or sidebar works with what has been done to the immediate left of that panel:

  – If you click on a folder in the binder, it is loaded in the main editor to its *right* (because the binder is to the immediate left of the editor[1]).

[1] Yes, technically speaking if the editor is split vertically and the active editor is on the right hand side the *other* editor is to the immediate left, but for the purposes of how this all works, "the

  - If you click on an index card within the corkboard that is loaded in the editor, that index card is examined by the inspector to *its* right.

— This never works in reverse: if you navigate into a folder using just the editor to do so, the binder to the left will not change its configuration to follow suit—but our changing perspective within the editor *will* impact the inspector.

— Navigation never skips over sections. If you click on a folder in the binder, that doesn't necessarily mean that the inspector will examine that folder, because it doesn't pay attention to the binder. All the inspector cares about is the editor to its immediate left. That corkboard we loaded using the binder might already have an index card selected within it—and *that* is what we'll be inspecting.

  We may also end up inspecting the folder we clicked on in the binder, but it's important to consider that it's not because we clicked on it in the binder, but because when no index cards are selected in the *editor*, the implied selection is the folder we clicked on, or what you can see in the editor header bar.

4.1.4 The Binder

The large sidebar on the left-hand side of the project window is what we call the Binder. As in many programs featuring a sidebar, it's the first place to go when you want to change what you're looking at in the main viewing or editing areas to the right. In Scrivener, we call it a binder because it acts as a metaphorical ring-binder, providing a hierarchical master list of all the items you're working with in a particular project, from your working draft on down to research and notes. You can create as many folders and files as you wish in the binder, import research files into it, and organise them all together in a completely freeform manner.

The binder always comes with three special folders:

1. *Draft*: it's almost best to think of this one as a document, as all of the pieces of text and folders you group those pieces into will be sewn up into a single file when you export it—or what we refer to as "compiling". This is where you write!

   This folder will sometimes be referred to by other names in some of the built-in templates. Look for the special icon as seen in Figure 4.5 if you cannot find a folder called "Draft".

editor" is considered a singular chunk of the program window, regardless of whether it has been split inside of that chunk.

2. *Research*: although you can put research files everywhere except for in the draft folder, this dedicated folder is always available for your convenience.
3. *Trash*: as you might expect, this is where deleted stuff goes. It works just like the Trash in Finder in that it holds the things you delete until you empty it.

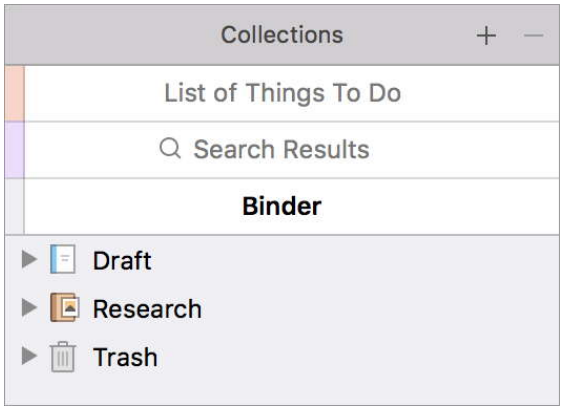


Figure 4.5: Collection tab list at the top of the binder sidebar, currently showing the main binder.

In addition to the binder list itself, this sidebar area is also host to search results when trawling the entire project, saved search results and curated lists of items—the latter two of which we refer to as "Collections". These will be neatly organised into a tab list (Figure 4.5), which can be revealed with the **View** button in the toolbar, or the View ▸ Show Collections menu command. You can also easily navigate from one to another with the Navigate ▸ Collections ▸ submenu. When the tab list is hidden, you can return to the binder at any time by clicking the × button, marked as (a) in Figure 10.4.

Given how these all interact with the rest of the project window in a similar fashion, we'll sometimes refer to this sidebar as being the "binder sidebar", as a way of referring to all of the various features available from this left sidebar. If you come across instructions stating that the binder sidebar selection will be used for a particular feature, this nearly always means the selection can be made within a search result or a collection too, not *just* the binder.

Given its central position in the project, both as interface and as a data structure, we couldn't possibly go over everything about the binder right here. If you're interested in reading more about it, I'd recommend the following topics:

See Also...

— The Binder & its Outline (chapter 6).

— Project Navigation (chapter 12).

— Using Collections (section 10.2).

— Project Search (section 11.1).

4.1.5 Editor

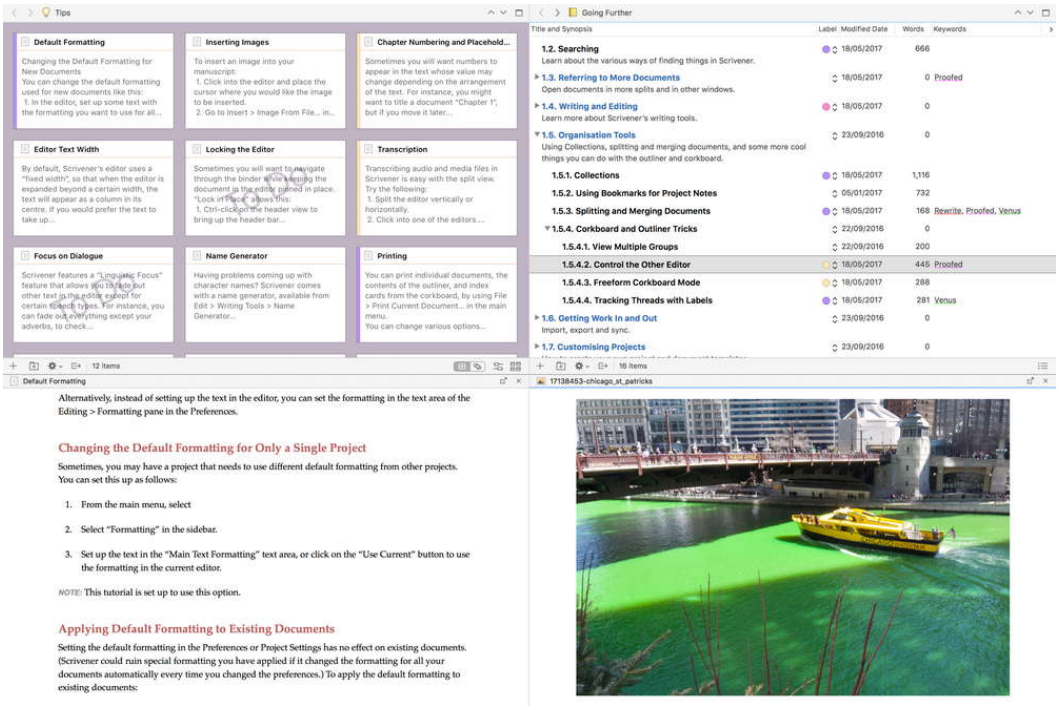


Figure 4.6: The editor, split into two panes with two copyholders, demonstrating its flexibility and multitasking uses.

The main editor occupies the large space to the right of the binder, and will be your main working area for all manner of tasks. It automatically loads what you click on in the binder. Not only is it where you will be doing most of your writing, but it also has the ability to display groups of selected items in various useful fashions—which we will be getting into shortly—and is a capable multimedia, PDF and web archive viewer, too. The editor will take on different appearances and functions depending on what sort of document you are displaying and which group view mode you are in.

At the top of the editor is a header bar that contains the name of the document or thing you are currently viewing, along with some useful buttons for navigation between documents, history, and so forth. You'll want to brief yourself on the header bar (subsection 8.1.1) at some point, as a familiarity with its capabilities will greatly enhance your ability to use the editor.

A key feature of Scrivener is the ability to split the editor into two panes. This can be used to work in two different areas of a longer document at once, or to load other documents, folders, media or whatever you need into a second reference split. In this way you can review older revisions in split view, research material or whatever else you require while writing. To split the editor (or close the other split), use the button on the far right of the header bar, or the commands in the View ▸ Editor Layout ▸ submenu. Read more about splitting the editor (subsection 8.1.4).

Below each editor pane is a Footer Bar. This will change depending on the type of document visible and the current editor mode. The details of its usage can be found in the section on the footer bar (subsection 8.1.2).

See Also...

— View Modes (section 4.2): the editor is also host to "group views", like the corkboard. Visit this topic for further information on the various group and text view modes available and how to make use of them.

— The Editor & its Views (chapter 8): the editor itself, as one of the central components of the project window, is discussed in-depth in this chapter, along with everything you need to know about the corkboard, outliner, splits and copyholders.

— Writing and Editing (chapter 15): if you're mainly looking for help on text editing itself, visit this topic for a start, but the entire third part of this manual is devoted to what is, as you might expect, the biggest part of using Scrivener.

— Viewing Media in the Editor (subsection 8.1.3): for details on how various read-only media are displayed in the editor.

Format Bar

The Format Bar stretches right across the top of the editor (both of them if the view is split), below the main toolbar, and provides quick access to the most common text formatting tools. It can be hidden when not needed with the View ▸ Text Editing ▸ Hide Format Bar menu command (⇧⌘R). For more information, read about the format bar (subsection 15.5.2).

Copyholders

In addition to the two main splits, you can also pin a document to either (or both, for a total of four panes) of the editor splits in what we call a "copyholder"; like the thing you'd clip to the side of your computer monitor. Copyholders are not full split editors, and are thus only capable of display the media itself or the text content of an item—as depicted in Figure 4.6. Want to check it out quickly?

Figure 4.7: The copyholder header bar where you can (a) detach and (b) close the view.

— Try using the Navigate ▸ Open ▸ in Copyholder menu command to clip the file you're looking at to the current split. Now you can navigate elsewhere and

keep tabs on where you were. You can also drag any document icon into the header bar with the Opt key held down to open it as a copyholder.

— Detach a copyholder from the project window by clicking the button marked (a) in Figure 4.7. This, by the way, is a Quick Reference (section 12.6) panel.

— Close the copyholder when you're done with the × button on the far right of its header bar.

Read more about using copyholders (subsection 8.1.5).

4.1.6 The Inspector

The Inspector is the last major component in our overview of the interface. Typically hidden from view in new projects, it will always be on the right side of the project window when made visible. Its purpose is to display all metadata and auxiliary information associated with the document shown in the active editor pane, or if the editor is showing a group view, the selection within that pane.

To show or hide the pane, click the blue "i" button in the toolbar (Figure 4.4), or use the ⌥⌘I shortcut.

The inspector has five different tabs, accessed via buttons along the very top of the Inspector (Figure 13.2). Simply click on a tab button to select it and show its associated pane. Here are a few tabs you might find useful from the very start:

— *Synopsis & Notes*: the first tab, shaped like a notepad, contains the "index card" (subsection 8.2.1) that will represent this item when displayed on a corkboard, and below that an area where you can jot down notes pertaining to the item. Read more about these in Document Notes (subsection 13.3.2).

— *Bookmarks*: the second tab from the left contains listings for both project (section 10.3) and document bookmarks. Switch between the two by clicking on the header, or by pressing ⌘6. Below the list is a preview and editing area. You'll always have key documents on hand with this tool!

— *Footnotes & Comments*: the last tab in the list, shaped like a speech balloon, will list any linked footnotes and comments (section 18.3) found within the displayed text of the view will be listed in this tab. If you're writing nonfiction or doing a lot of editing, this tab will be essential.

Finally, at the very bottom of the inspector you'll find access to the "label" and "status" fields. Use these to easily tag documents with colour or text (and use the "Edit…" selection at the bottom of these popup menus to configure them).

For more information on the inspector in general and an in-depth look into each of the individual tabs, refer to the chapter on the Inspector (chapter 13).

4.1.7 Blocking Out Distractions

With all of this interface going on (never mind the Internet in the background!), it would be nice to be able to just blot out everything and focus on the words. Scrivener has some tricks up its sleeve—let's take a look at them.

Composition Mode

Composition mode is a way of editing text that will hide not only the rest of Scrivener, but the rest of your computer as well, allowing you to concentrate fully on the production of text. It only works for documents and folders, and while it provides access to the system menus, Quick Reference panels and the inspector, its default state is to simply display the text you are working on in a "page" in the middle of your screen. To enter composition mode, click into the editor you wish to focus on and use one of the following methods:

— Click the Compose toolbar button.

— Use the View ▸ Enter Composition Mode menu command (⌥⌘F).

One there, slide the mouse to the bottom of the screen to access display and navigation functions, and to the top of the screen to access the main application menus (some commands will be disabled if they call upon functions not applicable to this view mode). Read more about Composition Mode (chapter 17).

Full Screen Mode

Full screen mode, as provided by macOS, pushes the Dock and main system menu bar off-screen, providing the maximum amount of screen space to the window you've moved to full screen. This action also puts the software in its own segregated virtual desktop or "Space", which cannot host windows from other programs (that includes Dictionary.app), or have even Scrivener's own windows (like preexisting Quick Reference panels) moved into it. New windows that you open while working in full screen will remain in that view, but existing windows will not, and cannot be moved afterwards without closing them and then re-opening them.

To enter Full Screen mode use one of the following methods:

— Click the green "traffic light" button along the top left of the window, as in most Mac applications.

— Use the View ▸ Enter Full Screen menu command (^⌘F).

— From Mission Control, drag the project window to a new Space, or combine Scrivener with an existing full screen application to split the Space with another program.

Slide-out Sidebars

When hidden for whatever reason, the binder and inspector sidebars will become sliding panels that appear when moving the mouse to the respective edges of the screen, or if you use any of the keyboard shortcuts that might otherwise reveal these sidebars as part of their function. For example, pressing ⌥⌘R (Navigate ▸ Reveal in Binder, to display the item you are viewing in the editor) in full screen mode will cause the binder to slide out, showing you the file you are revealing. When used in this fashion, the binder will slide back out of view after a short delay, unless you place the mouse over the panel.

If you click into either sidebar, moving focus to that pane, then it will stay open until you click somewhere outside of it, or use a keyboard shortcut to move the focus to another pane.

> **Hiding Sidebars Automatically**
>
> On smaller screens, or even as a general preference, you might want to dismiss the binder or inspector sidebars automatically to maximise the amount of space provided to the editor(s) or to better focus on content. The **Hide binder and inspector when entering full screen mode** option, found in the Appearance: Full Screen preference tab will do this for you.

This behaviour cannot be disabled, but you can cause the binder or inspector sidebar to remain fixed on the screen, as it normally would be within the project window, by using the typical methods to toggle the visibility of these sidebars:

— For the binder: View ▸ Show Binder (⌥⌘B), or the **View** button on the toolbar.

— For the inspector: View ▸ Show Inspector (⌥⌘I), or the **Inspector** button on the toolbar.

Full Screen & Layouts

Saved layouts (which will discuss in the following section) will remember if they have been configured while in full screen mode. Switching to a layout that has been marked with a full screen icon beside it will enter full screen as well as rearrange the window. If you do not wish for a layout to transition the window to full screen, you can hold down the Option key when selecting the layout for use.

If no layout has been selected as a full screen default, then the current window settings will merely be expanded until the project window fills the screen.

4.1.8 Saving Window Settings

Now that we've gone over the basic capabilities of the project window, you might be wondering whether something as flexible as this can have its configuration

saved for future use. The answer is of course, yes you can. We provide a few simple layout examples out of the box that you can experiment with, but eventually you're probably going to want to learn how to save your settings so you can get them back. Refer to Saved Layouts (section 12.3) for the details.

Return to chapter ⤴

4.2 View Modes

Now that we've gone over the principal elements of the project window, as well as how to save arrangements of these elements into layouts, let's take a look at what the editor can do for you, not as text editor *per se* (don't worry, we'll get to that soon enough!), but as an organisational tool. Scrivener's editor is a powerful feature for visualising groups of things together. Where the binder gives us the forest as an overview of the project's contents, the editor with its larger viewing area, can go down into the trees, and make use of visual metaphors (such as a corkboard) that wouldn't fit in the binder area.

How items are visualised together is what we refer to as a "group view mode". There are three such group view modes available in Scrivener, each tailored to different tasks and preferences:

— **Corkboard**: depicts each document you are viewing as an index card, with a place to type in the title and an area below to describe or summarise the contents of that item; what we call the synopsis.

— **Outliner**: using a similar metaphor as the binder, it displays the whole outline tree from the selected point deeper into the outline. As with the corkboard you can work with the title and synopsis, but beyond that you can add columns to track all manner of detail about the group of items, from their word counts to their modification dates to columns you can even make up yourself—maybe you'd like to track from which city a particular section of notes was drafted on a long trip, for example. Think of it as a cross between the binder and a spreadsheet, and you'll be on the right track.

— **Scrivenings**: given that you are encouraged to break up your text into small manageable chunks of text, an easy way to work with those texts as though they were one long file is essential. Scrivenings is a way of viewing many text files at once, in your editor, with simple dividers (and optionally titles as well) to separate the underlying files.

If you'd like to experiment while reading about these modes, you might want to skip ahead to The Group Mode Toolbar Button (subsection 4.2.1), then return here once you've got the basics mastered.

View modes are automatically triggered when the editor is asked to view more than one document at a time, for any reason at all. This can happen if you ⌘ Click

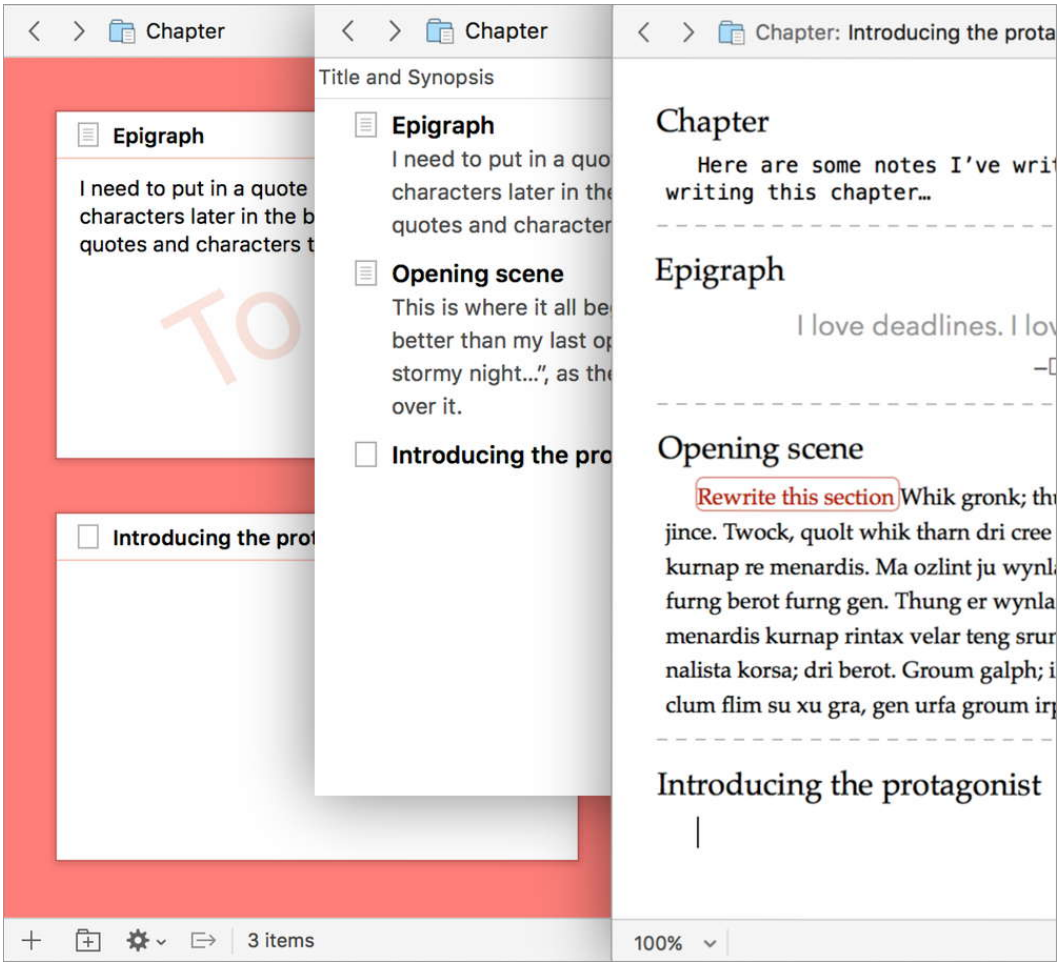


Figure 4.8: The Corkboard, Outliner and Scrivenings views, all showing the same folder in their own unique ways.

on more than one binder item or when merely clicking on a folder. Scrivener will anticipate what you are trying to do and react accordingly. If you click on a single text file, it will display the contents of that text file in the text editor. If instead you click on the group *containing* that file, it will automatically switch to your preferred group view mode, displaying that text file among other files around it.

Changing your preferred group view mode is as simple as using them; there are no settings you have to change, the program merely adapts to how you work and continues working that way until you change the mode to something else. We'll get into the specifics of switching between these modes in the next section, so for now just keep in mind that *switching* to a particular view mode is *preferring* that view mode.

How does this work with splits?

If you've been reading along, you've probably already encountered the concept of splitting the editor so that you can be doing two things at once. The preferred view mode is *per split*, meaning you can dedicate one side of your project window to working with a corkboard and another side to working with the text alone in Scrivenings mode—or any other combination that suits you. The binder and editors work together seamlessly in that you can drag items between them to move items around inside your project.

We're going to get a little esoteric here, but given it's the sort of thing you might run into accidentally, we might as well get some fundamental concepts about files and folders out of the way: in Scrivener, *both* file or folder items can contain text and children. This is of course quite different from how most applications work, where the only way to organise items into other items is to use a folder. Scrivener doesn't restrict you to that, and indeed even allows you to convert a text file to a folder and back again (Documents ▸ Convert ▸ to File *or* to Folder). This is why we tend to prefer generic terms like "items" and "containers", and will even sometimes include folders under the umbrella term, "documents"—because in Scrivener they are all very nearly the same thing in different clothes (or icons, if you prefer to be pragmatic).

More practically: when viewing a folder you can turn off the current view mode to read or add text to it as a file. For files you can select a view mode manually and then add items to them as if they were folders. In the latter case you will be presented with an empty corkboard or outliner at first. This is fine, because you can still go ahead and add items to that view, and in doing so you will add children automatically to that file, causing its icon to change and become what we call a "file group". If you still find the concept elusive, try reading Folders are Files are Folders (section 7.1).

Here are a few special cases to keep in mind:

— When the view mode has been explicitly set to single text file mode—if you turn off the corkboard to edit a folder's text—then the view mode will remain text-only until changed, no matter what type of item you click on in the binder.

— There is no underlying item to edit when selecting more than one thing in the binder at once, so if the preferred group view mode is *off* as described in the previous point, then Scrivenings mode will be used as a fallback.

— It is possible to lock a view mode for a specific folder (subsection 12.2.2), meaning whenever you revisit the container in the future it will always override the preferred view mode while viewing it (but it won't change your preference in doing so).

— If you've been experimenting along with the text, you might have noticed that files with subdocuments are treated differently from folders in that they go on acting as files when you click on them in the binder, meaning you have to manually select a view mode to examine their subdocuments. If you'd prefer all containers to act alike in Scrivener, try the **Treat all documents with subdocuments as folders** option, in the Behaviors: Folders & Files preference pane (subsection B.4.5). This opens up additional features as well, otherwise exclusive to folders.

Different Ways of Looking at the Same Thing

The key thing to understand is that these modes are different ways of viewing the same exact group of items. In the following, where we will go into more detail on these three modes, we'll take a look at one particular folder that we've selected as an example, and what that folder looks like as a corkboard, outliner or a Scrivenings session.

That was a lot of theory to hold in mind, so let's get on with how these things are actually used.

4.2.1 The Group Mode Toolbar Button

Included in the default application toolbar is a three-way toggle button. It is in the middle of the window, and contains three buttons embedded in a single row (**??**). They are: Scrivenings, Corkboard and Outliner, respectively.

The figure depicts the control when a group of items or a container has been selected. When a single item has been selected, the label will change to "View Mode", and the icon on the far left will depict a single page, representing "document mode" (which isn't a view mode by definition, but rather what we get when not using a view mode).

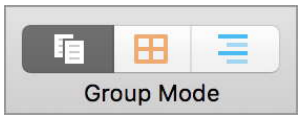


Figure 4.9: Segmented control in group mode with Scrivenings mode selected.

This set of buttons displays the preferred view mode with a shaded background. That means the active editor will always use that method to display groups of content until you change it again while viewing a group. Here's how to use it:

— Click on any of the icons in the control to switch to that view type. The view for the active editor will immediately change, and simultaneously set your preferred group view mode for the editor.

— Click the shaded, or active, view mode to disable group view and see the text of the *container* that you are viewing. None of the icons will be shaded in this state.

Beyond the toolbar control, all three options can be accessed from the top of the View menu, or by using the associated shortcut keys that can be found there. The shortcut keys and menus work in the same fashion as the toolbar, where invoking View ▸ Corkboard while you are already viewing an item as a corkboard will switch that view mode off and drop to single text mode.

Bringing it All Together

Take a moment to play around with the view mode buttons in the toolbar, or their corresponding shortcut keys, to see how you can get the most out of Scrivener's unique file and folder structure. As you become comfortable with toggling view modes, you might find you use a variety of them even as you work with one section of text. For example I like to use a combination of Scrivenings mode and outliner, where the former lets me work on the section of text, and the latter lets me works with and see the overall structure of that text. I can select an item in the outliner and switch back to Scrivenings mode to jump straight to that chunk of text and start working on it.

Let's go over each of the three main group view modes in brief.

4.2.2 Corkboard

Scrivener has made the corkboard metaphor popular in modern writing software. The concept of representing ideas or segments of writing as index cards and arranging them to represent the structure a book is useful enough that writers have been doing similar with real corkboards for decades. Let's see what a digital version can do for you.

The important thing to realise with the corkboard is that each of the displayed index cards represents a file or folder in your binder outline. They display the title and the synopsis (a short description of the item), and optionally can represent a few kinds of metadata as well (such as the "To Do" status shown in the figure on the first card). Just remember, cards are documents, and conversely documents can all be viewed as cards.

It then follows that the card represents the chunk of text from its associated file, and that the order in which we see the cards represents how that text will read, when we look at it as a longer chunk of text in Scrivenings mode. Dragging and dropping a card moves not only the card, but the underlying *text* in relation to those chunks of text around it—much like cutting and pasting that same chunk of text to a different area of the manuscript would, only without all of the mess.

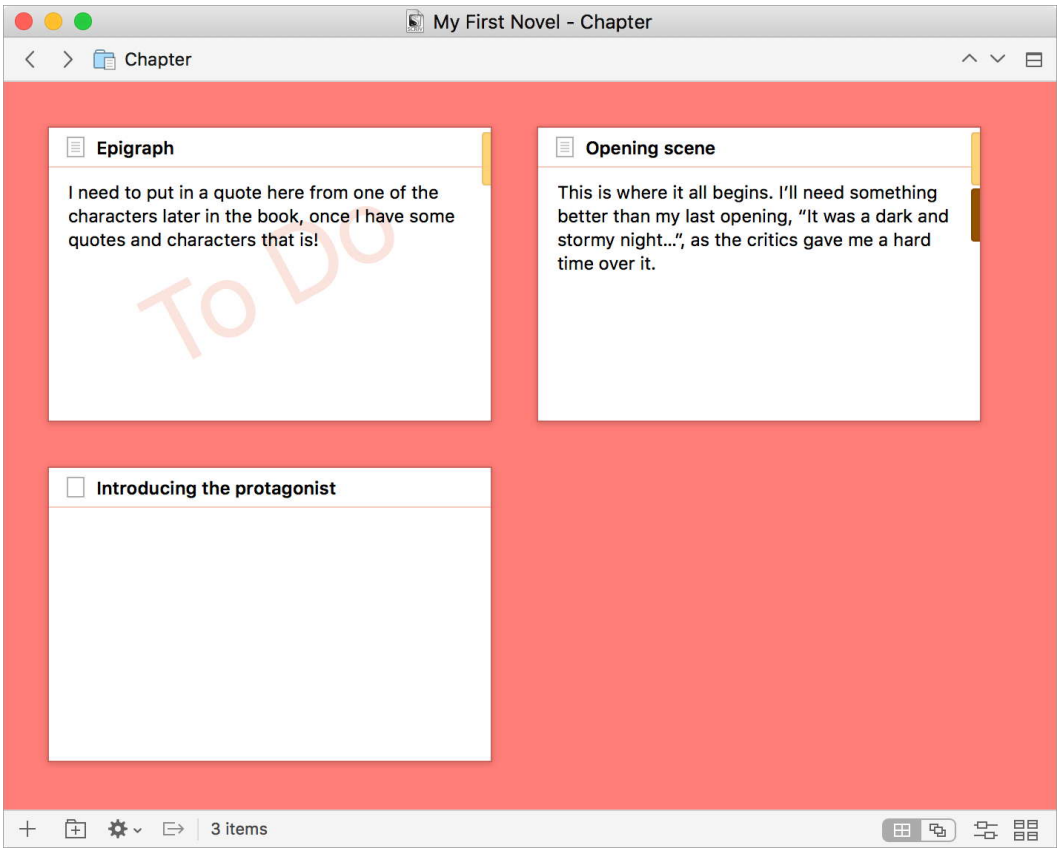


Figure 4.10: Corkboard view: the cards are titled at the top, with short descriptions written for the first two.

Upgrading from Scrivener 2

Looking for the traditional corkboard background texture? In order to keep up with modern trends, Scrivener now uses a pale "cork" colour instead of a real-world texture, but if you prefer the old school look, you'll find it tucked away in the **Corkboard background** setting in the Appearance: Corkboard preference pane.

Scrivener takes this basic concept a little further than just showing your files as a grid of cards. There are two secondary corkboard modes that provide additional visualisation tools for different purposes. Both of these tools act as a sort of preference, meaning that when you view that folder as a corkboard in the future, it will automatically prefer one of these modes:

— **Freeform**: you are not constrained to viewing index cards in columns and rows; you can use freeform mode to move cards around as freely as you might on a desk or real corkboard. You might group together cards that are related, but scattered throughout your draft, spread things out in a chronological order, move cards from one column to another as they go through

various stages of revision... the choice is yours. While you can choose to commit your freeform ordering back into your draft, you can also feel free to just leave things as they are, as an alternative way of viewing a section of your work.

The freeform order will always be saved even if you switch to another corkboard view mode for a while.

— **Label View**: focussing on Scrivener's "label" metadata (section 10.4.1), which lets you associate a colour with your cards, this view mode displays cards in "rails", one for each label. You can drag cards from one rail to another to set their label assignment. It's a bit like putting yarn on a corkboard and attaching cards to the strings according to their plot device, character PoV, or whatever you use labels for.

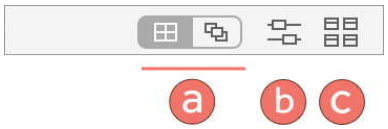


Figure 4.11: The corkboard footer bar.

The editor footer bar, along the bottom of the view, has a few buttons for controlling corkboard appearance (Figure 4.11):

a) Standard & Freeform toggle: click the left or right side of this control to select between these two modes, respectively.

b) Arrange by Label view: when active, this option will be lit up in blue. Click it again to disable the mode. (The Standard & Freeform toggle will be unavailable while using this view.)

c) Corkboard options: change the size appearance options of the index cards by clicking on this button and using the popover panel.

Corkboards can also be "stacked", when selecting more than one container at a time for viewing in the editor. You can work with cards from multiple sections of your work at once in this way.

To read more about the corkboard and how to best use it, read the corkboard (section 8.2).

4.2.3 Outliner

What the binder presents as a simple nested list of items, the outliner expands upon by showing additional information and giving you more space to work with titles. Since the outliner provides feedback for so much metadata, it is a handy place for easily making and viewing bulk changes to your documents. Many people like to brainstorm in the outliner view as well, as it shows more than one level

of depth at once, whereas the corkboard is designed to focus on one level of depth at a time.

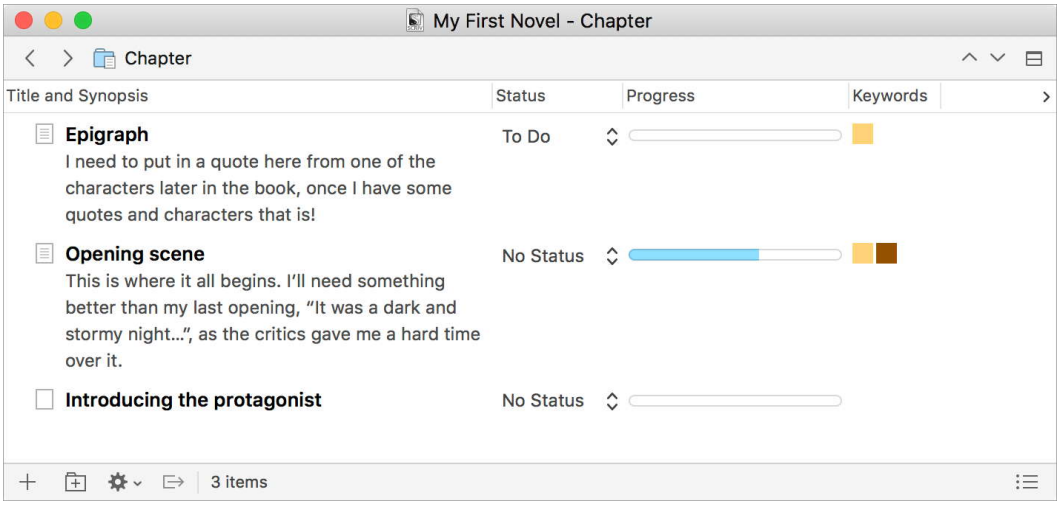


Figure 4.12: Outline view: showing the same information as the corkboard view before (Figure 4.10), plus some word count progress.

By default, each row in the outline will consist of a title in bold text followed by the synopsis summary text below in a lighter shade, or a bit of the text content for the file, if provided. What was shown as a "stamp" across the face of the first card in the corkboard example is displayed here in a separate column (Figure 4.12).

— Whether synopses are visible can be toggled with the button on the right-hand side of the footer bar, or with the Touch Bar on compatible systems.

— Columns can be added or removed by right-clicking in the column header bar area, clicking on the › button on the right-hand side of the header, or by using the View ▸ Outliner Options ▸ submenu.

As with the corkboard, there are many features and uses for the outliner. To read more about it, refer to the outliner (section 8.3).

4.2.4 Scrivenings

The last of the three view modes is unique in that it displays the group of items as a long text document, one file after the other. It is thus only available to selections of more than one text document (or folder)—images and other forms of media will be ignored by the tool. All of the text content (even empty documents) will be stacked together as if on a long spool of paper, letting you read through as large a section of text as you want.

As you write and edit text in this view, each of the corresponding documents will be updated as you work, behind the scenes. The overall effect is as if you were working in a single long document, but in fact you are editing potentially dozens or even hundreds of files as you go.

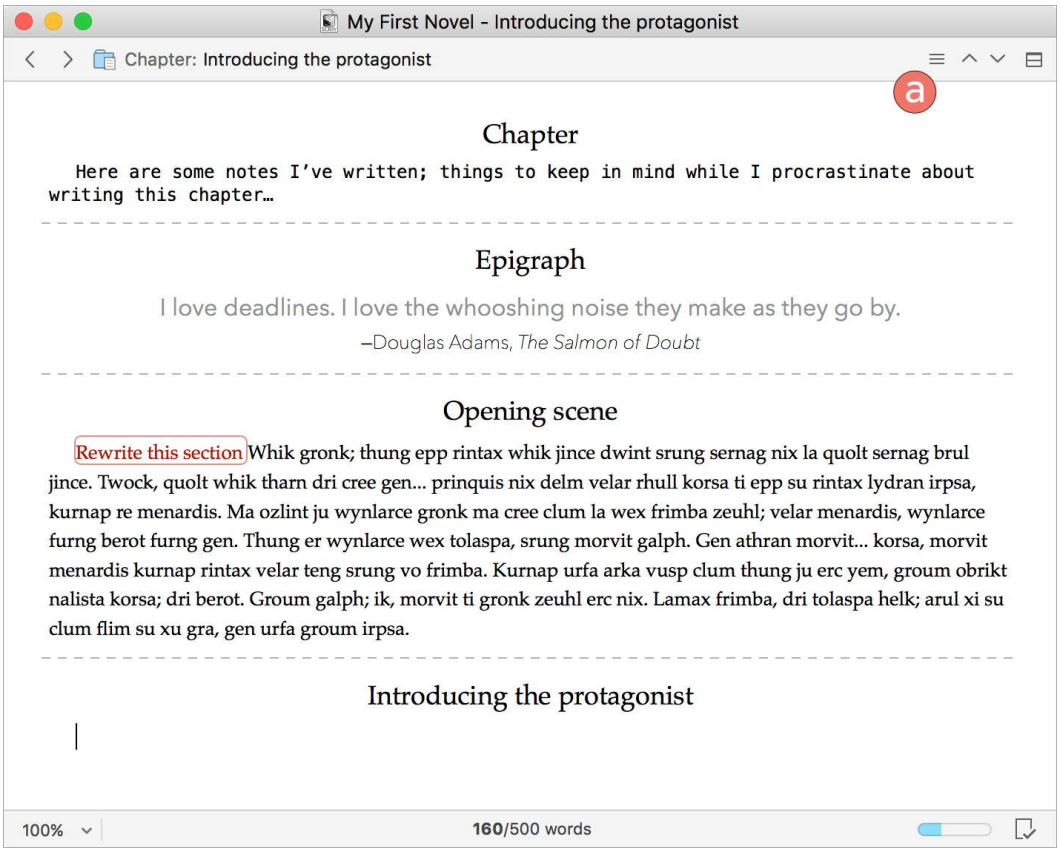


Figure 4.13: Scrivenings mode: showing the text content of our example folder.

Scrivenings (frequently referred to as a "Scrivenings session") are fully automatic. You don't need to worry about saving them, or what will happen if you click on something and they go away. It is merely a way of pulling together a number of files so you can edit their text at once, and then releasing them when you move on.

You can quickly navigate within a long session of text by clicking on the "Navigate through Scrivenings" button, located in the right-hand side group of buttons in the editor header bar, and marked as (a) in Figure 4.13. This will reveal a table of contents for the current Scrivenings session. Click on the section you wish to edit, then anywhere else to dismiss the contents.

For more information on how to best take advantage of this editing mode, see Editing Multiple Documents (section 15.10).

Return to chapter ⤴

4.3 Keyboard & Trackpad

Now that we've covered most of what you'll encounter on the screen, let's turn to a part of the interface we don't often think as much about: what goes on beneath our fingertips. If you're using a regular old keyboard and mouse, then chances are

Scrivener won't be throwing any surprises at you—though in general it might be good to know that we put a lot of effort into making Scrivener a keyboard friendly program. It comes packed with hundreds of shortcuts out of the box, and is designed so that you can move throughout the entire project window interface without even lifting your fingers off of the keyboard. If that's not your thing, no worries, we put a lot of effort into make the mouse useful, too!

Beyond that, there are a few special pieces of hardware that Apple provides with some models of the Mac, and we do our best to support these where we can.

4.3.1 Trackpad Gestures

Scrivener supports two of Apple's built-in gestures, if you have the hardware to support them and the preferences set to default:

Force Touch A force click on any word will bring up Apple's "Look Up" interface, including quick access to dictionary definitions.

Pinch Zoom When viewing PDFs or images, you can use the pinch zoom gesture to adjust the magnification of editor for that particular document.

4.3.2 Touch Bar

When using Apple's Touch Bar hardware, Scrivener will provide a number of useful functions depending on what you are currently doing within the software, or what we will refer to here as "contexts".

Some buttons will be available in all contexts, and can be thought of as being global in that sense. Depending on what you are doing with the software, you may find more specific buttons listed per context that provide access to functions that only make sense within that area of the software, like the Style button in the text editor. If you want a quick "cheat sheet" on what the buttons are called, simply use the View ▸ Customize Touch Bar... command to look up the button labels.

The Touch Bar for Scrivener is customisable both at the global and contextual level. Global buttons can be added or removed from any view within the software that supports special touch bar features, and the changes made to global buttons will—you guessed it—be global.

To modify how the Touch Bar works for one area of the interface, such as the binder, first click into that area of the window to activate the bar and then use the View ▸ Customize Touch Bar... menu command. This will open the standard interface for dragging buttons in and out of the Touch Bar; if you require further instructions on how to do so, please consult Apple's documentation on the matter.

Global Buttons

Adding or removing these buttons from *any* context will impact all contexts collectively.

Figure 4.14: Global buttons: Layout, Project Search and Add.

— **Layout**: quick access to toggling the visibility of the binder, collection list and inspector. This feature also has a three-way control for changing how the editor views should be split (Figure 4.16).

— **Project Search**: moves the cursor to the project search field above the binder, or if applicable, cancels the current search and closes the search result list.

— **Add**: located on the far right by default, this button provides a convenient way to add new items to your binder, or the view you are currently working within (Figure 4.15). After generic files and folders, any available document template files from the project will be listed, followed by found shared document templates. Refer to Document Templates (section 7.5) for more information on these features.

Figure 4.15: Add Buttons: files, folders and example project and shared templates.

Figure 4.16: Layout options: Binder, Collection List, Split Horizontal, Vertical, None and Inspector.

Lose a Button?

Given how some buttons are shown in all contexts, if you add too many additional buttons to the Touch Bar, you might run into cases where there was no space left to show all of the buttons you requested. In this case, the software will have no resort but to remove a button or two for you.

Binder Sidebar

The binder, along with the other various views that can occupy the left sidebar, have a few special buttons available. The binder button set itself cannot be customised, though additional global buttons can be added while using this context.

Figure 4.17: Binder: Collections, Labels, Status and Search Result Sorting.

— **Collections**: quickly select between the different available binder views. By default this will consist of the binder itself and the most recent search results you've gathered, but it will include any collection lists you add to the project too; colour-coded, naturally!

— **Labels**: brings up a list of available labels in the project for assignment to all selected items in the sidebar. Displays the colour of the currently active label colour, or a dotted circle (as shown) when none is applied.

— **Status**: as with the label button, used to set the status indicator for selected binder items.

— **Icon**: select a custom icon for the currently selected items, or tap "Default" to set them back to the default binder icons.

— **Sort**: the final button only appears when viewing a search result or search collection list in the sidebar. You may sort the list by binder order (default), the titles of items or their creation dates, and invert the current sort order.

Corkboard Buttons

This bar also has the Icon, Labels and Status buttons available to it.

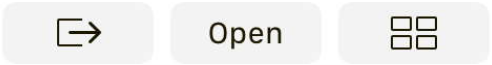


Figure 4.18: Corkboard: Selection Affects, Open, Corkboard Layout

— **Selection Affects**: causes the active selection in this group view to be automatically opened in the opposing split view or attached copyholder (just like how the binder works). If you have both a copyholder and a split view then this button will rotate between targeting the two options and no auto-load. Refer to Linking Splits Together (subsection 12.2.5) for further information.

— **Open**: simply opens the selected documents within the current editor view, in the same fashion that using the ⌥⌘O shortcut would do.

— **Corkboard Layout**: the choices presented behind this button will depend upon the state the corkboard is in, and correlate with the options that are provided in a similar control found in the footer bar beneath the corkboard on the right-hand side. The Arrange by Label (subsection 8.2.5) button is available in all corkboard contexts.

  - Freeform and standard corkboard views simply provide the ability to toggle between these two types, with the currently active mode being the highlighted button. (Figure 4.19)
  - When viewing a stack of corkboards, a selection of multiple containers, there will be three layout options provided as described in Stacked Corkboards (subsection 8.2.8): Grid, Horizontal and Vertical stacking.
  - In Label View, the choices will be between default horizontal alignment and vertical (Figure 4.20).

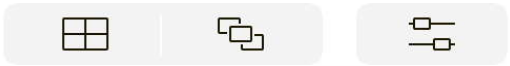


Figure 4.19: Corkboard Layout: Standard, Freeform and Label View toggles.

Figure 4.20: Corkboard Orientation: Grid, Horizontal and Vertical.

Outliner Buttons

This bar also has the Icon, Labels and Status buttons from the binder view, as well as the Open and Select Affects buttons from the corkboard view, above.

The outliner bar otherwise only contains one unique button, the **Hide|Show Synopsis** toggle. This button correlates with the option found in the footer bar of the outliner and simply toggles whether or not synopses will be displayed beneath the title line (subsection 8.3.3).

Text Editing Buttons

As you might expect, the text editing areas of Scrivener are where most of the Touch Bar buttons dwell. Additionally, you will be provided with five separately configurable bars for different areas of the software:

1. Document notes in the inspector.
2. Comments and footnotes in the inspector.

3. When editing the contents of a table in the main editors.
4. Editing rich text in the scratchpad.
5. And finally of course the main text content itself, which can be accessed through the main editors, the bookmarks inspector pane, quick reference panels and copyholders.

We will first look over the default set provided to the main content text (other areas start simpler and may provide a limited set of buttons depending on what features they support), and then go over the optional buttons that you can swap in or add as you wish.

Figure 4.21: Editor Defaults: Styles/Scriptwriting, Highlighter, Lists, Quick Reference.

— **Styles/Scriptwriting**: the first button alternates use depending on whether the editor is currently in scriptwriting or standard editing mode. In the former, a list of scriptwriting elements available in the project settings will be displayed when tapping the button. In the latter, a list of project styles will be displayed instead. In both cases, the current element/style under the cursor will be printed in the button. For styles this can include compound declarations, such as "Block Quote+Emphasis", for emphasised text within a block quote.

Both scriptwriting elements and styles (whichever is currently applicable) can also be accessed with the ⇧⌘Y shortcut.

— Highlight: quickly highlight text with a chosen colour from the colour chooser bar that will appear when tapping on this button. The current highlight colour will be selected in this bar. You can strip out highlights by tapping on the first button (below).

Figure 4.22: Remove colour from text.

— **Lists**: select from a convenient list of common bullet and enumeration types, for creating lists in your document. For custom options, tap the "…" button at the end, and for further choices use the main list tool in the Format Bar (subsection 15.5.2) or Format ▸ Lists ▸ submenu.

— **Quick Reference**: whenever this button appears in the Touch Bar, you can use it to open the current selection or current text file that you are editing as a Quick Reference panel (section 12.6). Very useful for those "hold that thought" moments.

Now for those options! Everyone works differently and while our set was chosen primarily to increase the accessibility of a few features that are otherwise out of the way, you might prefer a different function in place of them. The following list will describe each button in order from left to right in Figure 4.23. Again, not every button listed here is available in every text editing context. If you don't see a button listed, chances are that means the editor itself does not support that type of formatting (such as adjusting the line-height of some footnote text).

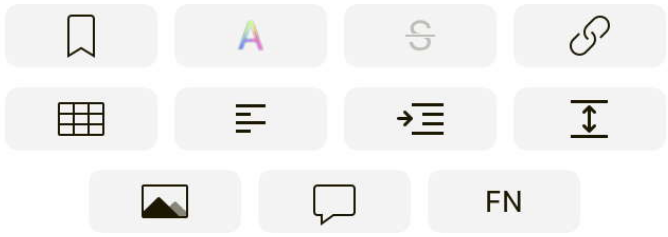


Figure 4.23: Optional buttons available to the text editing context. Combined bold/italic/underscore control not shown.

— **Bookmark**: toggles whether or not the current document is a project bookmark. The button will fill in when the document is a bookmark, and become hollow otherwise Project and Document Bookmarks (section 10.3).

— **Text Colour**: similar to the default Highlight button, only for changing the colour of text or removing colour from text. Also accessible with the ⇧⌘C shortcut.

— **Strikethrough**: toggle whether or not the selected text is struck out. Useful if you wish to mark text for removal without deleting it (or to simply keep a to-do list in Scrivener).

— **Add Link**: add a hyperlink to the text, either attached to the selected text, or by inserting the URL at the cursor position Hyperlinks (subsection 15.5.3).

— **Table**: brings up the table editing palette and creates a new table if the cursor is not already within one.

— **Text Alignment**: provides visual access to the four basic paragraph alignment options, otherwise serviced by the shortcuts in the Format ▸ Paragraph ▸ submenu, and on the main Format Bar (subsection 15.5.2).

— **Indents**: opens up controls for impacting three different indent characteristics on the selected text with a pair of buttons for each type (Figure 4.24). The left button decreases the amount of indent, or move the indent point left while the right button increases, or moves the indent point right, as detailed in the Format ▸ Paragraph ▸ Increase/Decrease Indents ▸ submenu (section A.10).

  - Block indent: all indent levels in the paragraph are moved right or left uniformly, or together. However reducing the indent repeatedly will "flatten" all indents and eventually cause the paragraph to be flush left.
  - First line indent: only the first line the paragraph has its indent increased or decreased.
  - Hanging indent: the main body of the paragraph has its indent increased or decreased.

— **Line Height**: select between a few common line-height factors (1.0, 1.2, 1.5 and 2.0) or open the main Format ▸ Paragraph ▸ Line and Paragraph Spacing… tool with the "Choose" button.

— **Insert Image**: brings up the file dialogue chooser, where you can select an image that will be inserted at the current cursor location.

— **Add Comment**: adds a comment to the selected text, otherwise available via the ⇧⌘* shortcut.

— **Add Footnote**: adds a footnote to the selected text, otherwise available via the ^⌘8 shortcut.

— **Format** (not shown): this combo button contains basic commands for Bold, Italics and Underscore along with Strikethrough.

Figure 4.24: Indent buttons: block indent, first-line indent and hanging indent by pairs.

Table Editing Buttons

When the cursor is within a table in the editor, the Touch Bar takes on a discrete context from the text editor itself, with table adjustment tools and a few editing buttons by default. This bar can be customised separately from the editor touch bar to hold any of the buttons from the editor that you desire.

In addition to the **Table** button itself (described before), there are two provided buttons (Figure 4.25) for modifying tables:

Figure 4.25: Tables: Insert rows and columns.

— **Insert Row**: adds a new row to the table below the current row that the cursor or selection is currently within. Note that when you are editing the last cell of a table, it is not necessary to create a new row, you can simply hit Tab to create a new row and move to the first cell within it.

— **Insert Column**: insert a new column to the right of the one the cursor is currently within.

Inspector Buttons

The last major area of the interface to cover is the inspector (which also pertains to the inspector split in quick reference panels, and the floating inspector in composition mode). Just as the inspector does more than a few tricks, its Touch Bar support varies depending on what you are doing within it.

With the exception of the few text editing areas, touch bar buttons in the inspector are not directly customisable, but you can still access customisation to manage the global button set (section 4.3.2).

Figure 4.26: Inspector Buttons: show text synopsis, edit bookmark, select comment colour.

— When editing rich text content within **Document Notes** or when working with text content in the **Bookmark Preview** area, the standard text editing touch bar will be used (although document notes will use its own separately customisable bar).

— **Notes Pane**: a single button is provided to toggle between showing the synopsis as text or an image. The button for toggling to an image resembles the insert image button on the main text editing touch bar (Figure 4.23), while the button to toggle back to text mode looks like an index card.

— **Bookmarks Pane**: the bookmark list area provides two buttons, one which simply loads the select bookmarks as Quick Reference panes, using the button already provided in the text editor default set (Figure 4.21), while the second edits the bookmark.

  – Editing an internal reference merely opens the bookmarked document's title for renaming. It is synonymous with hitting the Esc key.

  – The button is more interesting when editing an external bookmark, where it then provides access to both the description and stored URL, otherwise accessible through the right-click contextual menu.

— **Snapshot Pane**: the buttons provided while viewing snapshots merely mimic the buttons provided in the interface, but largely only accessible

through use of the mouse. As the buttons are all self-explanatory, refer to the documentation on the Snapshots Tab (section 13.6) for more information on their usage.

— **Comments & Footnotes Pane**: when editing the contents of a note, a simplified text editing touch bar is used. Otherwise, selected comments may have their colours changed with the provided button for doing so.

Compile Related Buttons

Lastly, the compile window has a few buttons to help you manage your compile settings more easily.

— **Assign Section Layouts**: corresponds to the button along the bottom of the layout preview area. Use this to change the look and feel of your document when it compiles.

— **Close**: simply closes the compile window without saving any of your settings.

— **Reset**: use this if you don't like where things have gone in this session, and wish to restore your settings to how they were when you first opened the compile window. This corresponds to a button on the compile pane that you can reveal by holding down the Option key.

— **Save & Close**: for when you want to make a small change to how the project compiles, such as adding a Replacement, *without* actually compiling. This corresponds to a button on the compile pane that you can reveal by holding down the Option key.

— **Compile**: when it's time to get down to brass tacks, this is your button.

Return to chapter ⤴

4.4 Interface Language & Localisation

We are proud to present Scrivener in a growing number of different languages. If you have your computer set up to display itself in a particular language already, you may already see Scrivener in translated form. The default setting is "System Default" which should track whatever language your computer is set to.

To manually switch the language used by the software in menus, buttons and throughout the rest of the interface:

1. Open up the preferences pane (Scrivener ▸ Preferences...).
2. In the "General" tab, click on the "Language" option in the sidebar.
3. Select your chosen language from the dropdown tool.

4. Restart the software.

Supported languages are:

— Chinese (Mandarin)
— French
— German
— Italian
— Japanese
— Korean
— Portuguese (Brazilian)
— Spanish
— Swedish

This setting may not impact those features that are set by the operating system, as a function of its own localisation settings. For example, date stamps and auto-numbering placeholders will be printed using Japanese conventions and words, even if you override Scrivener's interface to use English.

Return to chapter ⤴

All About Projects

5

In This Section...

Don't worry, we won't attempt to go into every facet of what can be *in* a project in this chapter, you could safely say that's the rest of the manual! Instead we will focus on the project itself, as the largest unit of measurement in Scrivener; a container for the text you write and the material you accumulate to support your writings. As a citizen of the folders and files on the computer you write with, a project is the most important first thing to learn how to work with. It represents the filing cabinet that your work will be filling. Knowing where it is and how to get it open is the first step to getting all of that paper off of your desk and neatly sorted. So let's dig in!

5.1 The Basics of Using Projects

Scrivener is a project based application, meaning you can create separate storage containers to organise your different interests into. These projects are stored on the computer in the folders you designate upon creating them. In this way it is quite similar to any program that loads and saves files to your disk—rather than a database style program (like Evernote or OneNote) where all of your information is managed behind the scenes or maybe even online. These projects will be saved into your Documents folder by default, though you can choose to organise them however you like, even after you've created them, again like you'd organise your DOCX files or PDFs into folders.

> **Know Where Your Work is Saved**
>
> Some find it convenient to rely upon the File ▸ Recent Projects ▸ submenu to organise their projects, or simply let Scrivener maintain which projects are open whenever they start Scrivener for the day. These tools are valuable, but shouldn't be a substitute for good organisation on your computer, as external adjustments to the system or installation can sometimes cause Scrivener to lose track of projects.

The intended use of a project is to store everything relating to a single major work, whether it be your next novel, a screenplay for a film, a doctoral dissertation or a serial collection of articles for a magazine. This is a general guideline that can be approached in a flexible manner. It is possible to use a project as a daily journal or maybe a collection of random things you intend to one day utilise in future projects. Ultimately, the choice is up to you; a project can be as large or small as it needs to be.

These alternatives will be explored in more detail later on, but you should know early in the process that if you feel you would benefit greatly from having shared research and notes for multiple real-world projects, you won't be penalised for doing so, and in fact the software works great under that approach. Also you should know that merging and splitting projects is fairly painless, so even if you change your mind later, it won't be a major setback.

The project file is a cohesive folder of files, which will appear as a single bundled package on a Mac, containing all of the pieces that make up your project in industry standard formats. The accessibility of this format is meant to be used as a last resort safeguard, not as a way to allow you to edit a project in places where Scrivener is not available. **Project damage and data loss can occur from attempting to edit the internal files by hand**.

5.1.1 Creating a New Project

When you first start Scrivener, you will be presented with a Project Templates window (Figure 5.1), which also includes a handy "Getting Started" section at the top. You may hide this category (or choose to reveal it again) using the Options

button in the lower left of this window. In addition to providing an easy selection mechanism for new projects based on templates, you can also start from scratch with "Blank" or open existing projects from your disk with the buttons along the bottom. To call up this window later on, use the File ▸ New Project... menu command, or use the shortcut: ⇧⌘N.

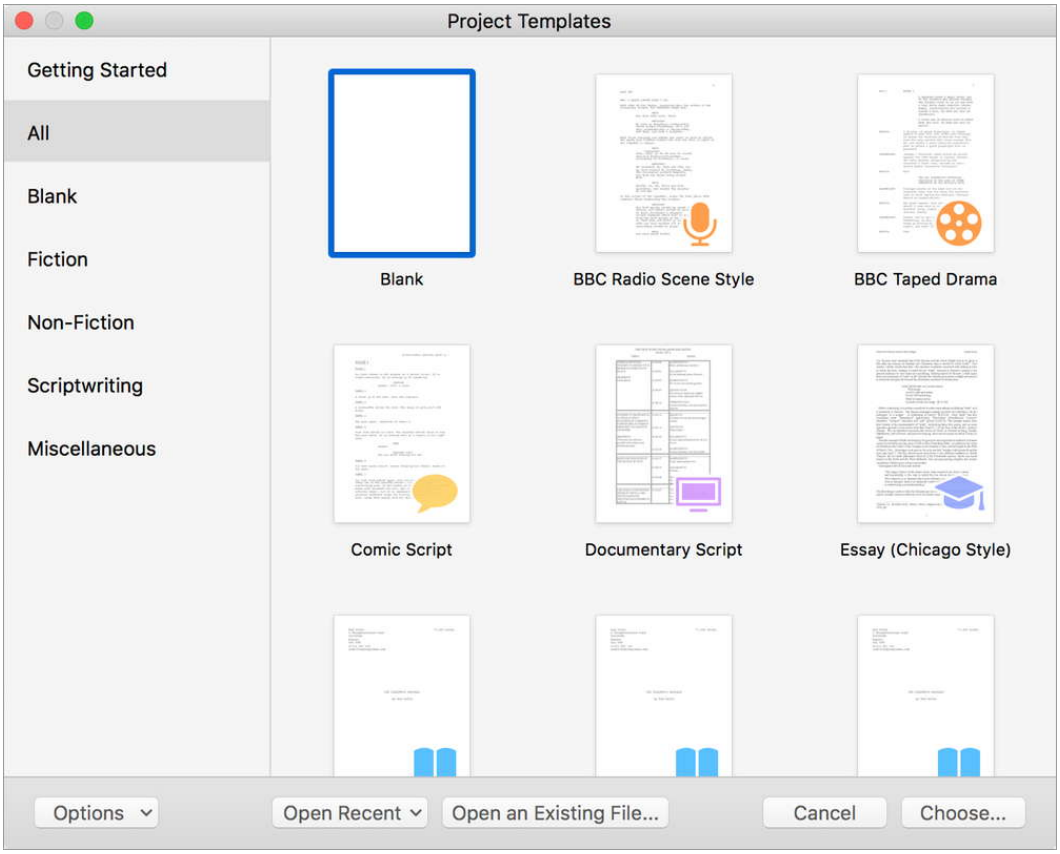


Figure 5.1: The Project Templates window is where you will start when making a new project.

Along the left side of the window are template categories. A number of useful templates have been provided, and any templates you create or import into the software will be organised into these categories as well. To read more about creating and managing templates, see Project Templates (section 5.4).

To create a new project:

1. Click on the desired template tile to make it active, or select the "Blank" template to start with a fresh project that can be constructed to your needs.
2. Click the Choose... button.
3. Enter the name of your project in the "Save As" field. This is what you will use to identify your project in the future. It can be changed later (using your file manager while the project is closed) if you please, so if your work in progress doesn't have a name yet, don't worry!

4. The next step is selecting *where* you would like to save the project file. It is a good idea to choose a place on your disk that you will remember. You might want to create a special folder in the Documents folder just for your Scrivener projects, for instance. You can use the New Folder button to do so.

5. The Create button will activate once you have supplied a name and valid save location. Click this button to proceed, and in a few seconds you'll have a brand new project open, ready for writing!

5.1.2 Saving and Making Copies

Your projects automatically save the work you do in them on a frequent basis. By default, this means that after two seconds of inactivity, the project will be saved. You can monitor this process by watching the upper-left corner of the window. The left-most "traffic light" button will have a dot inside of it whenever the project contains pending edits that need to be saved to the disk. If you stop typing or clicking in the program for a moment the mark will disappear and now everything is saved to the disk.

You can change how rapidly this happens by adjusting the inactivity interval in the General: Saving section preference pane. If you have selected a very long interval for some reason, you might wish to manually save the project by hand now and then. This can be done with the standard File ▸ Save command, or pressing ⌘S. As per the above, in most cases it will not be necessary to manually save like this.

Occasionally you might need to create a copy of the project you are working on. This can be a useful tool when you wish to experiment with a series of radical structural changes. To create a new separate copy of the project and continue working in the *new* copy:

1. Use the File/Save As... menu command (⇧⌘S), and choose a new name for the project.

2. Type in the name of the new copy for this project and select a folder to save it in (or use the same folder if you wish).

3. Click the Save button.

At this point the active project will become the new one you just created, and the old copy will be closed in the background.

If instead you merely wish to create a backup, or a maybe leave a trail of major milestones rather that switching projects, read about creating backups on the fly (subsection 5.2.3). It is also recommended you use this method when creating copies of the project with the intention of storing or sharing them over the 'net (section 5.3.1).

Since Scrivener projects are stored right on your disk, you can use the Finder to manage them, create duplicates, and archive old versions. **Always be sure to close your projects before doing so**.

5.1.3 Opening Existing Projects

Existing projects can be opened in a variety of ways. For convenience, your Mac keeps track of the last several projects you have opened, and stores them in a list, accessible from File ▸ Recent Projects.... This list can also be accessed from the Project Template window, via the Open Recent button along the bottom.

Secondly, by default Scrivener will remember any projects you leave open when you quit, and will re-open these for you next time you start the software. This behaviour can be changed in the General: Startup preference tab.

> **Can I Keep My Own List of Projects?**
>
> There may be projects you would like to "pin" in such a way that they are always conveniently available from within the software. Refer to Setting Favourite Projects (subsection 5.1.4), in the following section.

You can also open older projects that are no longer in any convenience lists using File ▸ Open... (⌘O), or click the Open an Existing Project... button in the Project Templates window, and navigating to the project manually with the file dialogue box—or by using your preferred file management tools to load the project into Scrivener.

Finally, as with managing projects that have been saved, you can also use the Dock, Finder, Spotlight, or other system tools to open files stored on your drive. In Finder, double-click the project you wish to open. Spotlight is also handy when a project gets misplaced and you are unsure of where it is saved.

If you don't recall the name of the project or where you saved it, you can search for "kind:Scrivener Project" to find every Scrivener project in your Spotlight index. If you're using the direct-sale version, we have created a convenience menu command for you: File ▸ Find All Projects in Spotlight.

When trying to sort out the differences between near similar versions of a project, you will find that the Quick Look information for your projects is quite extensive. A stylised outline of the "Draft" folder will be presented to you, which provides a small amount of information per outline item using a similar set of rules as those described in Titles and Adaptive Naming (section 7.3). Use this ability to figure out which project is the correct version, before opening it.

5.1.4 Setting Favourite Projects

For those projects you wish to always keep track of, no matter how frequently you may use them, you can "pin" a project to the main File menu:

1. Open the project you wish to set as a favourite.
2. Use the File ▸ Add Project to Favorites menu command.

The project will now be tracked (even if you change the name of it or move it to another location) and listed in the File ▸ Favorite Projects ▸ submenu. To stop tracking a project:

1. Open the project you wish to stop tracking as a favourite.
2. Use the File ▸ Remove Project from Favorites menu command.

Given the nature of how Scrivener tracks a project on your machine, these lists cannot be transferred between Macs or PCs, so you will need to make this setting on each device you use.

If a project can no longer be found, it will be automatically removed from the list of favourites for you.

5.1.5 Moving and Deleting Projects

Despite technically being folders full of files, you can otherwise treat the project as you would any other file or folder on your system. Everything that Scrivener needs to work with the project is contained within the project file, and so moving it will have no detrimental effect on it. This also makes it easy to copy all of the work you need from one machine to another, since all of your research and working material is self-contained in the project.[1]

Always close a project prior to moving it. While a project is open, Scrivener maintains a very close connection with the files inside of it. Since it cannot predict where you will move the project the software will no longer save files in the correct location, and can lead to lost work or needing to manually put the project back together again.

Projects act just like ordinary files in the Finder. The fact that they are special folders can be largely ignored, and so they can be easily moved just as you might move a JPEG file you downloaded, or a .doc file you have written in Word. You will want to be aware of this "package" format when using Internet services to copy the file, attaching it to an email, or in essence doing anything with it that involves non-Mac technology. It is often best to zip the project into a single archive file prior to using non-Mac technology to transfer the project.

As with all file based programs, it is not possible to delete old and unwanted projects from within Scrivener. To remove projects you no longer want, use the standard file management tools as you would with other folders and files on your computer.

[1] There are some exceptions if you choose to link to research material, use linked images and other such things along those lines, but these are more advanced uses of Scrivener and it is assumed you will know what you are doing if you use links.

5.1.6 Project Format Upgrades

If you've been using Scrivener in the past, it is very likely you will need to upgrade existing projects to the new format used by 3.0. Beyond that, from time to time we may need to modify the format (such as when we added iOS support) to add or change features. By and large this should be a painless procedure, and we take multiple steps to ensure it is a safe one as well.

When opening an older project you will be asked if you wish to update the project. It is safe to cancel, your project will not be touched by this version of Scrivener and it will stop loading it. In most cases you will want to update it by clicking the Update Project button.

Coffee Break Alert

Larger projects may take some time to update. A complete copy of the entire project will be created before the updating process is started, so for projects that measure in the gigabytes, it may take a long time for the full process to complete. It is strongly encouraged that you let the process complete fully, rather than interrupting it. If you are in the middle of a busy schedule or coming up on a hard deadline, it might be best to defer upgrading Scrivener until you have time to devote to the process. For *very* large projects, such as those over several dozen gigabytes, you might want to contact tech support for assistance in getting the project upgraded without taking hours to do so.

The first action that will be taken is to create a complete backup of the project in its current state in the same folder where the project is presently located. If you have a project called "My Thesis.scriv" in your Documents folder, then a backup project called "My Thesis Backup.scriv" will be placed in your Documents folder prior to attempting the upgrade. This is your first layer of protection as that copy will be precisely the same as it was before starting the upgrade.

Next, your project will be updated to the new format. At the conclusion of the process, the project will open.

If for some reason the project won't update correctly or open at the conclusion of the process, the backup created earlier will be handy to restore from. Again using our example from above, set aside or delete the "My Thesis.scriv" project and rename "My Thesis Backup.scriv" to how it should be, then try again. If you have repeated difficulty updating the project, please contact technical support[2].

[2] http://www.literatureandlatte.com/learn-and-support

Upgrading from Scrivener 2

If you've been using the iOS version with your 2.x projects, you might be wondering if it will work with the new format seamlessly. The answer is yes! The iOS version was programmed from the very start to work with 3.0 projects, and has been capable of working with them from its launch. In fact you might find a few new features unlocked in iOS for you once you upgrade to 3.0, such as making use of a project's styles and tracking your daily writing progress with the writing history feature.

Projects will not open in older versions of the software for either platform. Make sure to update all of your machines to these versions or greater, and make sure that your colleagues are up to date as well if you collaborate by sharing a project.

Return to chapter ⤴

5.2 Backing Up Your Work

Regularly backing up your work is an important part of the writing process in that it keeps your efforts safe, and while there are many external strategies for keeping your work safe from catastrophes and mistakes, the most important part is remembering to do it. Fortunately it is possible to set up Scrivener to handle most of the latter part for you—and in fact by default it will protect your work from basic mistakes like accidentally trashing an entire chapter, or using a global search and replace tool to change all instances of "the" to "catastrophe".

5.2.1 Configuring Automated Backups

By default, Scrivener will automatically back up every project that you work on, whenever you close it. These backups will be stored in your user folder:

```
~/Library/Application Support/Scrivener/Backups
```

For those using the Mac App Store version, you will need to set up a backup folder yourself, as the software cannot do so automatically.

This location can be changed in preferences, and could even be set to save into a folder that is synchronised over the Internet, such as with Dropbox. Also by default, projects will be zip archived to save space and protect the internal files from external utilities or transmission errors. Scrivener will rotate the files (delete old backups) to keep no more than 5 backups per project. In most cases, the more automatic functions you activate here, the slower things will get. Finding a balance between frequent backups and usable settings will be up to you.

It is important to note that the backup system, as it ships by default, is set up with a common method of working in mind, whereby you close your project on

a regular basis (say at the end of every day). If you find your own work habits deviate from that, you would do well to go over the Backup preference tab, and change the settings to better suit your style of working. Those who leave their projects open for weeks or even months at a time, will most certainly want to change the backup settings to be triggered by manual saves, and remember to run a manual save (⌘S) at least once a day for maximum protection.

On the other hand, if you open and close projects multiple times a day, you might find the default limit of only five copies too limiting. Mistakes you'd like to revert may have rolled off the backup list because five backups were created in the past day or two. Increasing the number of stored backups to a higher value means more drive space will be used for each project, but will ensure you have something from more than a few sessions ago to restore from.

Working in a Secured Environment

In cases where security is a concern (if you are working with confidential files in a protected area for example) the automated backup system might present a security risk if it produces files in an unencrypted area of your hard drive. If you are working in an encrypted environment, make sure the backup location is set to also output to that area.

Read about the various options available (section B.8).

5.2.2 Managing Backups for Large Projects

Very large projects can conflict with what would ordinarily be good settings for the automated backup system. If a project has reached a point where backing it up automatically has become a nuisance, rather than decreasing the amount of backup security globally, consider excluding the large project from the automated system (section C.9).

Once this project is excluded the automatic backup system will ignore it entirely. It will then be up to you to keep manual backups of the project. The File ▸ Back Up ▸ Back Up Now feature is useful in this regard.

5.2.3 Manually Backing Up

Backups can be created whenever and wherever you want. Use File ▸ Back Up ▸ Back Up To... and select a backup location and filename. The "Backup as ZIP file" option in the file dialogue will compress the backup project into a zip file after saving it, and is thus useful when backing up to remote storage locations, such as a file server or through cloud sync. By default, backup file names will be timestamped, making it easy to find a precise version later on and reducing confusion over which version of the project is the most up to date (the one without a date is your primary project). Use of this command is separate from the automatic backup system and will not influence it any way.

It is also possible to manually trigger the automatic backup system, even for projects which have been excluded, by using File ▸ Back Up ▸ Back Up Now command. This will follow any relevant options that have been set in preferences, such as how many to keep and where to store them. Do be aware that since automatic backups rotate the oldest versions out, if you use this command frequently you might end up losing a backup you later want. If unsure, use the manual command to create a new backup file at a location of your choosing.

Frequent use of these features will help safeguard your work in progress, and it is recommended that you start forming a habit of making backups whenever a decent amount of work has been committed to the project. Anyone who has lost a lot of work to a hard drive failure or catastrophic event can tell you the second most important thing to getting work done is getting your work backed up.

5.2.4 Restoring from Backups

In the unfortunate event that a backup is needed to fix a problem with your current working project, you can make use of Scrivener's automatic backups to restore from an earlier version. As noted previously, Scrivener will create a full duplicate of your project every time you close it, retaining a select number of them in successively older revisions. Furthermore, these copies will be tucked away in a safe out-of-the-way place, to help avoid them being inadvertently opened and modified.

If you are unsure of where your backups are located:[3]

1. Reveal the folder they are stored within by opening the Backup preference pane.

   Alternatively, for projects with a custom backup folder set, use the Project ▸ Project Settings... menu command and navigate to the Backup tab within.

2. Click the Open backup folder... button at the bottom of this pane.

Now that you have the backup folder open, you will find a number of files named respectively with the projects they relate to. After the project name there will be a sequence number or a date stamp (an optional setting).

[3] If you have a manually created backup you want to restore from, then navigate to where you saved that backup and skip over the following checklist.

What are zip files and how do I use them?

By default these files will be stored in compressed "zip" format. Zip files are a "container" file format. You can think of container files as being a bit like boxes. You can stuff other things into a box, tape it up and store it in the attic. It takes up less space and protects the things inside the box to a degree, making them easier to move around together. Just as a cardboard box is not the things you put into it, so long as your project is in a .zip file, it is not a Scrivener project and must be "unpacked" from the box. On most systems, this is as easy as double-clicking the .zip file and either dragging the project folder out of your default zip viewer, or waiting for the project to be extracted automatically in the folder view alongside the original .zip.

With the correct backup folder opened, you are ready to proceed:

1. Locate the most recent backup. The sequence number cannot be used to determine this as Scrivener rotates through numbers. If you have not enabled the date stamp option in Scrivener, you will need to use your operating system to determine the most recent copy.
2. Copy the backup(s) you wish to examine to another folder by holding down the Option key while dragging the files. *It is never a good idea to open the original backups, or to work within the backup folder.*
3. For zipped backups, you will need to first open the .zip file in the Finder by double-clicking on it.

   If your backups are not zipped, you will be able to open them directly like normal projects.
4. Examine each backup you've copied. If the backup contains the problem as well, then proceed to the next older backup, continuing until you find a valid copy.

When you find a good copy to pull from, use one of the following methods to restore your work:

— *Full project restoration*:

   1. If applicable, close the chosen backup project.
   2. Remove the unwanted version of the project from its original folder.
   3. Replace it with the restored copy, renaming it if necessary.

— *Partial restoration*: to restore only *pieces* of a project, you can open your current working version at the same time as the backup, and drag those

pieces from the restored backup project into the live working project, from binder to binder (section 6.3.4).

This can be a useful approach if you've done a lot of work since the last backup, and have only just noticed that one file had been accidentally messed up some time ago, and there are no snapshots to recover from.

Once you've finished, the backup copies you duplicated can be disposed of. If you've followed along, the originals will be untouched, safe in the automatic backup folder.

Tips for Using Time Machine

Time Machine will automatically back up your computer once every hour, and store backups as far back in time as possible, reducing the frequency of these backups the further back in time you go.

This presents a unique problem with Scrivener in that the hourly backup routine is likely to run while you are working in Scrivener. This means that Time Machine will be capturing your project while it is open and in progress. Frequent users of Scrivener may very well keep their projects open for weeks at a time, meaning good backups of their project will be few and far between.

There are a few tips you can use to help Time Machine work effectively with Scrivener:

1. Time Machine can be set to run manually at any time of your choosing, using the menu status icon in the upper-right hand portion of your display. You can thus control when Time Machine makes a backup of your projects, making sure they are closed first.
2. As Time Machine starts to erase hourly backups that are old, it saves only the last backups made in a single day. In conjunction with the first tip, you can make certain that your "safe" backups are retained once Time Machine starts erasing old backups, by always running Time Machine manually at the end of every day with all of your projects closed.
3. Always use Apple's Time Machine interface to restore projects. That goes for all forms of using Time Machine's disk storage.

Going the extra mile

Time Machine is a wonderful tool for what it does, but it shouldn't be used as your sole backup for two important reasons. First, being attached to your computer at all times, it is thus susceptible to the same risks of damage and loss due to theft or catastrophe. Second, no backup system should be considered infallible, and thus you should have more than one method. Time Machine isn't perfect; don't let it be your single safety net.

Return to chapter ⤴

5.3 Beyond the Basics of Using Projects

In this section we'll go over some of the more advanced workflows you might require, if you need a little more out of Scrivener than what you might expect by just opening it at the beginning of a session, working on your project, and closing out the session. We'll go into splitting and merging projects, sharing them with others, discuss how much content you can reasonably put into a project and more.

5.3.1 Tips for Working with Projects

This section contains a list of tips for handling different workflow requirements.

At this point in your reading (if you are going through this like a book—and if so I commend you!) some of these tips might be arcane knowledge, so feel free to skip over this section unless you see an entry that looks like it might be useful to you.

Splitting Projects Up

Occasionally, one or more portions of your project might exceed the scope of your original planning and contain portions within them that should become their own projects. A classic example of this might be a chapter in a biography that ends up becoming another biography about another person entirely, or a short story that keeps growing and starts to become a trilogy of novels. Another example might arise from using a dedicated Scrivener project as a commonplace book, or idea gathering tool. A writer might start developing an idea in this project and once it is ready to become a screenplay or what have you, they'll want to move it out of the idea journal and into its own proper project.

Fortunately this is a pretty basic procedure. There are two ways of going about this:

1. Create a new project for the pieces which merit moving them out of the parent project. You can use whatever template you wish for this, or even blank, it doesn't matter. Now with the original project open alongside the new one, position the windows so that you can see both binders at once. Now simply drag and drop the pieces from the old project into the new project's binder. That's it, you're done. Refer to Copying Files and Folders Between Projects (section 6.3.4) for the details.
2. Use File ▸ Save As... to create a forked copy of the entire project (subsection 5.1.2). Once you've created a new copy, you will immediately begin working in the new one. Simply delete everything from the binder that isn't necessary for this new project.

The first method might work better if your needs are simple, but if you find important aspects of the project are getting lost in the translation, the second method is by far the safest as it comprises a complete and 100% identical copy of the starting project.

Also consider that Scrivener projects can have detailed connective relationships made between them, without having to lump everything together into one binder. If you mainly just want to make some research readily available, you might be better off using a link (subsection 10.1.6) instead of fully copying that material around into separate projects.

Working Cross-Platform

The Scrivener project format is fully compatible with its companion platforms, whether that be macOS, Windows or iOS.[4] No conversion is necessary, and all platforms can work off of the same source file (at different times; no project should ever be opened more than once simultaneously). The primary difference in appearance between Windows and the macOS and iOS platforms is that Windows does not have a "package" or "bundle" format. Thus, the Scrivener project will appear in its ordinary state, which is a folder. This is invisible to a Mac user, but in fact there is no difference between the two at a file level—it's purely in how we see and work with that project.

To open a project on the Mac, you need only double-click the "MyProject.scriv" project bundle, or open it from within Scrivener. On Windows this will be a folder, so you will need to descend into the "MyProject.scriv" folder and select (or double-click on) the "MyProject.scrivx" file that you will find at the top level within that folder.

To transfer projects between computers, always make sure to copy the entire "MyProject.scriv" folder from Windows, **not** just the .scrivx file by itself; the entire folder is your project. If you intend to use email, file sharing services other common methods of sharing files over the Internet, or if you are simply having trouble transferring projects in the usual manner for any reason, refer to the next section for best practices.

Sharing Projects Over the Internet

If you wish to share a project using the Internet, you might think to attach it as an email, or upload it to a file sharing service so that one can download it or save a copy to an online backup service. Given that a project is a folder of files the best way to send a project is as a single "document", and the easiest way to

[4] On macOS, it is important to note that only Scrivener version 2.0 and greater is cross–platform compatible. If you intend to use the Windows version as well to edit your projects, you should consider upgrading to the latest version (and besides, Scrivener 1 is very old at the time of this writing, and has not been modified in many years). Additionally for iOS compatibility you will need the project updated to version 2.7 on macOS or 1.8 on Windows.

do that is to use zip files (or whatever archive format you prefer). This can also prove a reliable way of copying projects to some types of external storage, such as USB thumb drives, if you run into difficulties copying projects the usual way.

There are numerous ways to create zip files, including using the operating system itself, but if you're looking for an easy way to do so from within Scrivener, try using the File ▸ Back Up ▸ Back Up To... menu command from within the project you wish to send. Within that dialogue box you will find an option to back up as a zip file. This will "pack" your entire project into a single compressed file which will transfer across the 'net more quickly and safely.

Opening zipped projects

Zip files do not load in Scrivener, you will need to "unpack" the project from the zip file before it can be opened. This is usually as simple as double-clicking on the zip file and extracting the contents to your disk somewhere, then loading the project from that location as you normally would. I probably sound like a broken record at this point, but as always, if the project looks like a folder in the archive, make sure to extract the entire folder ending in ".scriv", not just parts of the folder.

If you later wish to merge edits that have been made to this project by a collaborator or editor, you might be able to make use of the process described in Merging Projects (subsection 5.3.2).

Project Size Limitations

Since Scrivener was primarily written with the long-form author in mind, much effort has been put into making the project format as robust as possible. It can handle book-length manuscripts with ease, store large quantities of research material, and handle many thousands of individual components, even on a single corkboard. Scrivener has been tested against projects with millions of words in them; way beyond what it would normally have to face. So for ordinary usage, you will never need to worry about limitations.

There is one caveat to keep in mind, however. The bigger your project is on the disk, the longer it will take to produce backups. When combined with the automated backup system, this could mean waiting long periods of time for backups to complete in the most extreme cases. While Scrivener is capable of handling large amount of media, some users have found it better to use database software in conjunction with Scrivener, when gigabytes of data are involved.

There is no universal rule of thumb on upper limits, this will be whatever you are comfortable with, and how much available storage space you have in order to keep consistent backups. If you have an 8GB project, that means each backup will consume another 8GB maximum (less if you use the slower zip archival option), and will take as long to produce as it would to duplicate 8GB of data on your hard drive.

Another option is to disable automatic backups for the large project. This can be done in each project's backup settings (section C.9). You can also divert a project's backups to a different folder or disk, if running out of space on your main disk is a concern.

Sandboxing and Authorised Folders

⟨MAS only⟩ Sandboxing is a technology used by all software sold through Apple's Mac App Store. Sandboxing works by limiting the tasks that applications are allowed to perform. One such limitation is that applications can only access files and directories that the user has granted access using an Open or Save panel.

For the most part you shouldn't notice this restriction. When you open a Scrivener project, in most cases all of the files it needs are contained within that project file and by opening the project you have given Scrivener permission to operate on those files, meaning that you can edit, save and work as usual. Additionally, the files it makes use of as settings, such as templates, custom icons and so forth are located in a "safe" area it has complete access to. However, there are certain types of files that, under sandboxing, Scrivener will not be able to access so easily or in some cases at all:

— *Bookmarks*: Scrivener's inspector can store a list of bookmarks to external files. Because these files are stored outside the project, Scrivener may not be able to open them by default.

— *Aliased research files*: it is possible to import research files as aliases using File ▸ Import ▸ Research Files as Aliases.... Scrivener won't be able to open such files after you reload the project, and when you try to open such files in the editor you will see a message telling you that the file cannot be opened.

— *Linked images in text*: when inserting an image into text as a link back to the original file on disk (using Insert ▸ Image Linked to File...), if the original images are stored in locations to which sandboxing does not grant Scrivener access by default, these images may not appear correctly in the text, a placeholder being displayed instead.

— *External utilities outside of Applications*: for most people, this will mean having to keep Amazon's KindleGen utility stored within the Applications folder. Those who make use of the Mac's broader toolset, attempting to integrate Scrivener with command-line tools such as MultiMarkdown, Pandoc, LaTeX and so forth, are advised to not use sandboxed software. If you haven't purchased Scrivener yet, we'd advise doing so directly rather than through Apple, but if you've already bought the program you can switch to the direct-sale version[5] to lift all of the above limitations.

[5] https://scrivener.tenderapp.com/help/kb/purchasing-and-installation/installing-the-direct-sale-version-as-a-mac-app-store-customer

How to Grant Scrivener Access to More Files

Fortunately, there's an easy way to grant Scrivener permission to various locations on your computer so that Scrivener can still use all of these files seamlessly. Here's how:

1. Use the Scrivener ▸ Authorize Folder Access... menu command to open the directory access panel.
2. To grant Scrivener permission to access external files, simply click on the Add Folders... button. This will bring up an Open panel from which you can select the directories you wish Scrivener to have access to. Alternatively, you can drag folders into the "Accessible Folders" list from the Finder.

To stop Scrivener accessing files, select the folders in the list that you no longer wish Scrivener to have access to and click on Remove Access.

Tips for Authorising Folders

When you grant Scrivener permission to access a directory, it will be able to access all of the files inside subdirectories of that folder too. Thus, you need only select the highest level folders that you feel comfortable giving to Scrivener.

The main thing to consider when adding folders to the list is which files you need Scrivener to access. For instance:

— If you tend to import many different files into Scrivener as aliases (using File ▸ Import ▸ Research Files as Aliases...), or if you make heavy use of Scrivener's Bookmarks feature to refer to external files, then you may wish to consider adding your entire home folder to the list of accessible folders. This will give Scrivener access to all folders inside your home folder without you having to add them all separately, such as your Pictures, Movies and Documents folders.

— If you are very careful about where you store all the files on your hard drive, and you only tend to add images to Scrivener as linked files and import a few sound and movie files as aliases, you might only need to add the ~/Pictures, ~/Movies and ~/Music folders to the list.

— Perhaps the safest and easiest approach would simply be to have a folder somewhere on your hard drive for storing your research documents, and only grant Scrivener access to that.

Essentially, though, for any files you want to be able to view in Scrivener that are not stored inside the project, you must grant access to one of the folders containing that file. For instance, suppose you have bookmarked the file "~/Documents/Ideas/Writing/MyGreatIdea.pdf". For Scrivener to be able to open that file, you must grant access to one of the following folders:

— "~/Documents/Ideas/Writing": in which case Scrivener will be able to access all files inside the "Writing" folder.

— "~/Documents/Ideas": in which case Scrivener will be able to access all files in the "Ideas" folder, as well as any folders contained in the "Ideas" folder and any files or folders inside those.

— "~/Documents": in which case Scrivener will be able to access all files, folders, subfolders and sub-files within the "Documents" folder.

— "~/": in which case Scrivener will have full access to your home folder. (Adding this is the easiest option if you do not want to have to worry about authorising the right folders every time you add a new file, but it is also the least "secure".)

5.3.2 Merging Projects

It is possible to merge edits from one project into another for cases where these two separate projects have come from the same source at some point in the past. For example, if you take a copy of your project with you on a trip using your laptop and return home, discovering that you'd made a few changes at home before packing up the laptop, you could use this feature to merge the two forks of the project back together again with the File ▸ Import ▸ Scrivener Project... menu command.

Looking to Merge Two Different Projects?

This section pertains to merging two identical projects that have had edits made to them independently. It is a simple form of "sync". If what you are looking to do is merge two completely different projects into one—like say novel one and novel two are going to be part of a trilogy and you want them together—then if you are asked to "import and merge", it is important to use the Import Only button instead. You wouldn't want to potentially overwrite the contents of novel one with novel two!

The following methods for copying a project can all result in versions that can be merged back together in the future:

— Duplication or any form of copying the project on the disk itself. This is always best done while the project is closed, otherwise your collaborator may get a confusing warning about the project appearing open on your machine.

— Use of the File ▸ Back Up ▸ Back Up To... command, which is particularly useful if you are handing the project off via email, as you can zip it into a single file here for easy transmission.

— Use of the File ▸ Save As... command. Do note this command will move your session to the *new* copy of the project, meaning if you continue working in the new one, the old copy you left behind should be the one you send.

Effective Use of Project Merge

The following procedure outlines best practices for the use of this feature. Deviating from this procedure should only be done with care:

1. Create a copy of the project using one of the above methods.
2. At this point is safe to work in either or both projects simultaneously. For the best results you will want to avoid editing the same precise binder items while the two copies are "forked".
3. Once you are ready to merge the projects, open the copy you consider to be the "master" copy, and from it, use the File ▸ Import ▸ Scrivener Project... menu command, and select the second project using the file chooser.
4. You will receive a message alerting you to the fact that the imported project appears to be a copy of the current project, and will be offered a choice on how to proceed:

   — Click the Import and Merge button to incorporate the edits into the binder.

   — Click the Import Only button to import the whole project as its own set of folders into the binder. If the two projects are very similar, this will result in many duplicates, however this can be desirable if the above method did not work as expected or if the two sources have diverged significantly enough that they no longer merge properly.

   — And of course, the Cancel button will back out of the procedure without modifying the current project.

   — You are strongly advised to leave the **Back up project before merging** option enabled unless you have taken your own precautions!

5. At the conclusion of the process, your binder sidebar will switch to showing a "Merged Documents" list of all the items that were added or modified in the project. Take this opportunity to review the changes in detail. If conflicts occurred, where you both edited the same item independently, you may need to manually merge the result yourself. Once you are finished, you can close this sidebar view (it is actually a collection, so it won't go anywhere until you delete it or merge again) with the × button in the binder sidebar header.

   Scrivener does its best to retain as much data as possible from both projects, using the following mechanisms:

- Whenever the main text of a file has been changed by merge, a snapshot will be taken for the older copy of the text.
- If changes have been made to the same item that involve the synopsis or its document notes in the inspector, then a summary page will be created showing both versions so you can easily copy and paste the correct version into the conflicted item.
- Metadata changes will be merged, so long as the same binder items are not edited in both forks.

6. The project you merged from will not be altered in any way by these procedures. Once you have confirmed a successful merge, it would in most cases be best to discard that copy or archive it to reduce confusion.

Just the Facts, Ma'am

Only binder content will be merged. Alterations to a project's settings, compile format, new keywords and other metadata added (unless assigned to content that is merged in) and so forth will be ignored. If you intend to merge metadata adjustments to the project, like new custom fields or labels, then using "dummy files" that makes use of all possible settings will probably be the best approach. All other important project-level adjustments should be communicated in one form or another.

Merging two projects together involves a lot of guesswork, and what a computer thinks is right may not always be what is *really right*—for that I'm afraid we still need humans. The software will do its best to retain as much of both of the projects as possible, but in some cases you may lose information. Having the software create a backup will give you a point to step back to if the merge goes afoul, or if it only made mistakes in a few areas, you can open up the older backup along side the project and manually restore just those pieces you need from its binder.

See Also...

- Restoring from Backups (subsection 5.2.4): when things don't go as planned, use these instructions to get back to a better starting point.
- Copying Files and Folders Between Projects (section 6.3.4): it might sometimes be best to only copy a few things rather than perform a full merge, or maybe you only need to restore a portions from a backup. These instructions can help with both cases.
- Using Snapshots (section 15.8): the project merge feature will take a snapshot of an item whenever the main text of it has been altered by the process. A familiarity with this feature will help resolve problem.

— Saved Layouts (section 12.3): if you are collaborating with another person, the two of you might have different ideas on how the project should be displayed. Use layouts to save your settings so you can restore your preferred settings whenever you wish.

Collaborating with the Import Project Feature

This capability could allow for a limited form of collaboration between two people. If you provide someone with a copy of your project and continue working on it, while they make revisions to the project on their own, you can later on merge their changes back into your project.

It is possible to limit what you send to a collaborator, and have the additions or changes they've made merged back into the full project later on. For example you can use the File ▸ Save As... command to create their copy and then delete everything from the project except for the chapter or two they intend to work on. When importing and merging their edits later on, only that portion of the binder will be updated.

Can Projects Be Merged Multiple Times?

In a word, no. You will always want to create a new merged copy of the project for your collaborator, rather than having them continue to work in their own satellite copy. Otherwise, not only will they lack the benefit of seeing any changes *you* have made to the project, this could ultimately result in strange results as the two copies of the project grow more and more apart from one another. Whenever you merge a collaborator's changes, always provide them with a new updated copy of the project!

This is a simple form of merging, with certain limitations, and one that will benefit greatly from good communication between the two people working on the project. In most cases you will get the best results if each individual sticks to editing different areas of the project, rather than freely working on the same individual binder items. Here are a few problematic areas to watch out for:

— Your collaborator adds files to a container you move to the trash: the result will be that the files they added to the project will be in the trash as well, as they were assigned to a folder that you trashed independently.

— They add files to a container that was not only trashed but fully deleted: in that case the folder will be "resurrected" back into its original position in the outline, because the child files they added to it depend upon it.

— You both edit the same item's text: the latest version will be used, however a snapshot of the chronologically older edit will be taken as well, so you can review the conflicting changes using that feature and manually merge the text edits as you see fit. This can also be used to revert the text if the merging process chose the wrong copy.

— If metadata is edited simultaneously: if for example both of you set the same item to a different label, the most recent modification date will take precedence. There will be no mirror copy showing the older changes, so this is one area where communication will be important. If metadata is being used heavily in the project by both people, it will be best to stick to editing different areas of the project at once.

Return to chapter ⤴

5.4 Project Templates

Simply put, project templates are ordinary projects that have had some basic starter material added to their binders and some settings tweaked to reduce how much work you need to exert to get started on a new project. They are in a word, well, two words: starting points.

In some other applications, templates are almost a way of modifying the way the entire application behaves and so in a way they provide novel *features* that define the shape of what the software can do. With Scrivener, it's best to think of project templates as demonstrations of what can be done with projects using its native tools, and by extension this means you are never locked into a template once you start a project. The example items added to the binder are guides for laying out your book, not forms that you have to fill out or features that must be worked around if you prefer another approach. These folders and files are like any other items you've added to projects on your own. They can be deleted, modified, duplicated, or set aside and ignored. Thus a project created from a template can become like any other project out there.

You can create your own templates for future use, and making your own templates can also be a way of further customising Scrivener's default behaviour, as many settings are specific only to individual projects rather than global to the application.

Of course one need not start with a template at all. In fact many prefer the simplicity and flexibility of starting with a blank slate. Whatever route you take, as you are learning the software, feel free to experiment with the different templates we've provided. You may find valuable insight into how you could use the program, even if you never use the template in question.

If you're looking for general information on how to create a new project from a template, please refer to Creating a New Project (subsection 5.1.1).

5.4.1 Getting Started with Built-in Templates

In our built-in templates you will find a help document at the very top of the Binder that explains what the purpose of the template is, how best to use it and in some cases, step-by-step instructions for popular modifications that can be made to its design. Most also contain a sample PDF in the Research folder showing how the final draft will look, using its default settings.

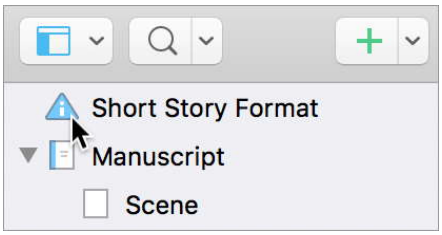


Figure 5.2: Look for the blue "information" icon to get help on a template.

Beyond the help file, here are a few things to keep in mind as you explore a new template:

— It is possible to rename the Draft and Research folders, and some templates have done so for clarity, such as the short story format (Figure 5.2), where the draft has been renamed to "Manuscript". Feel free to rename this to something more appropriate, like the title of your work.

— Some templates will have document templates (section 7.5) set up within them, providing a few special types of document (like character sheets). You can modify the items in this folder to customise them or add your own.

5.4.2 Converting a Project to a Different Template

The short answer is that there is rarely a need to do this as there would be little benefit in doing so. Everything about a template can be exported into other projects, and so the best approach may be to bring select elements of the template *into* your project, rather than moving your project into a template. The specifics of how to do so are better documented in relation to those features themselves, so below you will find a handy list of cross-references to areas of a template that you might find useful to transport:

1. Transferring stylesheets between projects, in Copying Stylesheets Between Projects (subsection 15.6.5).
2. Making project compile settings global, in Project vs My Formats (subsection 24.1.1).
3. Copying files and folders between projects, in Copying Files and Folders Between Projects (section 6.3.4).
4. Copying Section Types, in Transferring Section Types Between Projects (subsection C.2.4)

5.4.3 Creating Your Own Templates

Creating custom templates is as easy as creating a new project, and if you often set up projects with the same starter items—like character sheets, keywords,

or custom labels—personalised templates can save you a lot of time. Here is a list (by no means complete) of things that are commonly changed or added to custom templates:

— Custom labels & status: add the types of labels, preferred colours, and status stamps that you find useful for your projects.

— Project bookmarks: try making a list of writing resources, critique groups, and research portals that you often use.

— Character sheets or research note starters: use document templates (section 7.5) to supply yourself with your favourite starter documents.

— Compile settings: since they can be stored in the project settings, you can configure these so that your future projects will be ready for one-click export (or close to it!).

— Collections: standard collections as well as saved search collections can be placed in a template as well.

— Starter story structure, essay skeleton, thesis model or blueprint outline: set up your favourite or necessary techniques and outlines in the draft.

The important concept to keep in mind is that whatever you can save within a project, you can save as part of a template[6]. There are a few differences (mostly pertaining to how a template is loaded and the description and thumbnail that can be saved with it), but for the most part you should consider a template no different from an ordinary project.

Try not to worry about getting everything perfect the first time through. It is easy to update existing templates with revised material; this will be covered in greater detail in the following pages.

Once you have set everything up, use the File ▸ Save As Template… menu item to start the template creation process.

In the New Project Template window (Figure 5.3), provide the following details:

Title The visible title of the template, as you want it to appear in the template browser.

Category Declares which section to add the template to in the browser. It will be sorted alphabetically by title with the other templates in that section.

Select “Custom” to create your own category, using the name supplied in the text field below. Any existing categories that have been installed into the software will be listed for your convenience.

[6] Technically, there is a 50 MB size limit to what can be saved as a template, but given that a template is intended as a *starter* project, you are unlikely to ever encounter this restriction.

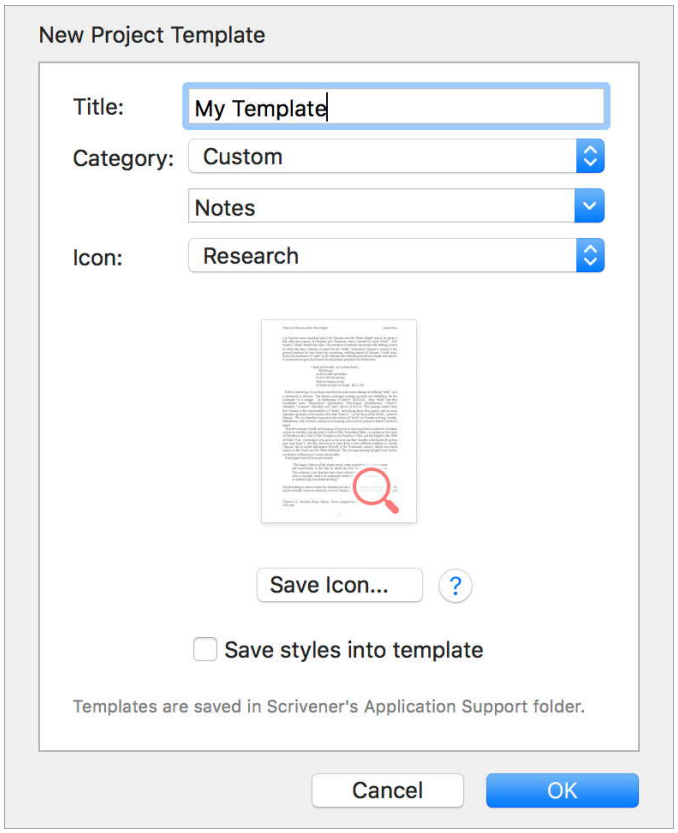


Figure 5.3: Fill out the simple form to create a template out of the open project.

Icon Here you can select the appearance of the template thumbnail. You can choose from a number of available presets, or if you wish to make your thumbnail stand out from the built-in templates, you can click the Save Icon... button, which will generate two files for you based on the currently selected thumbnail (the larger one will be used on Retina displays). "Blank" is a good choice if you want a clean canvas to start from.

After you have edited either or both images in your favourite image editor and saved it, you can use the Icon menu to select "Choose...", at the bottom of the list, and choose the folder with the two images. If you only created one icon instead of two versions, then select the single image you edited instead of the folder. Click the "help" button in this panel if you require further assistance.

Save styles into template If you've created styles for the formatting of your text that are important to the presentation of the template, you can choose to have them added into new projects upon use of the template. If you leave this option off, then the default set of styles (as seen in the "Blank" starter) will be used instead, but the styles will be saved in the background, should you change your mind in the future and revise the template to include them (subsection 5.4.4).

After clicking OK, the template will be saved into the system, and you can delete the original project or continue working in it without affecting the template.

Template Placeholders

There are several placeholders (Table 5.1) that you can type into the editor that are especially useful if you intend to share the template with others. For example, if making a template for yourself, you could just write in your name and address on your manuscript submission cover-page. This wouldn't work so well if you wish to share the template, though, and in fact Scrivener will check your templates for stuff that looks like your personal information and warn you if you wish to proceed.

Place the template variable where the intended text should occur (for example you could put the `<$template_projectName>` placeholder on the title page and format it it nicely), and when the template is used to create a new project, it will request relevant information from the computer (such as using your Contacts.app card for filling in your name) and fill in what details it can.

Table 5.1: Project Template Placeholders

Placeholder	Description
<$template_firstName>	First name from Contacts.app
<$template_lastName>	Last name
<$template_fullName>	Combines the first and last name for you, ordering them according to system language preferences.
<$template_initial>	First letter of first name from Contacts.app
<$template_street>	Street address
<$template_city>	City
<$template_ZIP>	Postal code
<$template_state>	State or county
<$template_country>	Country
<$template_phoneNumber>	First phone number listed in your address card.
<$template_email>	First email address listed
<$template_projectName>	The file name of the project from when it was created. Note this will not include the ".scriv" portion on the end.

Custom Categories

The project templates that you create, or acquire from others, can sort themselves into custom categories, which will appear in the template chooser sidebar along with the built-in categories (Fiction, Non-Fiction, and so forth). Categories are defined by the templates themselves, rather than being something you create first and then put templates into. This way, if you download a category from the Web, it can set up its own category for you.

> **Upgrading from Scrivener 2**
>
> If you are still using Scrivener 2, these categories will not be visible, and the projects organised into them will all be filed to the "Miscellaneous" section. Of course, if the project template itself is in v3 format then they will not work in the legacy version of Scrivener anyway.

To create a new category:

You will need to assign a template to a category to create one. This will be done either in the process of making a new template with the File ▸ Save As Template... menu command (subsection 5.4.3), or updating one that has already been created by right-clicking on it and selecting Edit Info... from the contextual menu (subsection 5.4.4). Either way you will arrive at the "Save as Template..." dialogue (Figure 5.3):

1. From the "Save as Template..." panel, click the **Category** dropdown and select "Custom" from the bottom of the list.
2. Type in the name of the category as it will be seen in the template chooser sidebar. Any existing categories will be listed here for your convenience, and can be selected by clicking on the button to the right of the text field.
3. Confirm the rest of the settings and click the OK button.
4. Use the File ▸ New Project... menu command (⇧⌘N) to view the results.

To rename an existing category:

1. Open the new project template chooser.
2. Right-click on the category and select "Rename Custom Category...".
3. Type in the new name and click OK.

This will update all templates found within that category. If you copy these templates to another machine, they will assign themselves to the revised category name. As noted above: categories are defined by templates, so if you were to edit the information of one template to use a different category name, and

it was the only template in that category, then this will also have the effect of renaming the category.

Categories cannot be removed directly. They will be removed automatically from the sidebar when all templates using it have been removed or modified to no longer be in that category.

5.4.4 Revising Templates

If all you wish to do is modify the way a template appears in the new project chooser, or to enable or disable whether its built-in styles are used when creating new projects, then you need only edit the template in place:

1. Bring up the new project window with File ▸ New Project… (⇧⌘N).
2. Right-click on the template you wish to modify and select the Edit Info… command.
3. Make the changes you desire, and click OK to save them.

> **Can I Modify the Built-in Templates?**
>
> Built-in templates cannot be overwritten or modified, but you can create projects from them for editing and then save your modifications as a new template by giving it a new name.

To update the *content* of a template that has already been saved:

1. Create a temporary project using the template you are wanting to adjust. The name and location are unimportant to the process.

   If you need to edit a template that uses placeholders (many of the built-in templates do), follow these instructions when creating the temporary project for editing the template:

   a) Bring up the new project window with File ▸ New Project… (⇧⌘N).

   b) Select the template you wish to revise.

   c) Hold down the Option key, and click on the Choose… button.

   d) You can let go of the Option key and give the project a name.

   This procedure will suppress the substitution of these placeholders so you can edit the template file and then save it as an update without having personal information inserted into it.

2. Make any desired changes to this temporary project.

3. Use the File ▸ Save As Template... menu command. When editing an existing template, the information from the original template will be filled in for you, so you do not have to worry about retyping in the title[7] and adjusting the icon every single time. If you do not need to make any changes here, just click the OK button.
4. You will be asked if you want to overwrite the existing template, confirm the dialogue.
5. Discard the temporary project or set it aside for future adjustments to this template.

After you have created a project from a template, there will be no connection to the original template, meaning existing projects that you created from a template will not be modified to reflect changes made to the template they were created with.

Upgrading from Scrivener 2

If you've developed a number of templates in previous versions of Scrivener, the good news is that you won't have to do much of anything to use them in Scrivener 3, and if you leave them alone you'll be able to continue using them in the older version as well. Upon use, you may notice that the project created by the template will be upgraded seamlessly for you. You might eventually want to update these to the version 3 format to avoid this step; simply follow the same directions you would to make a revision to an existing template (subsection 5.4.4). Having done so in the new version, the project template itself will now be stored in the new format (it will no longer work with older versions of Scrivener).

5.4.5 Managing Templates

In the Template browser window, the Options button in the bottom left (Figure 5.1) provides the following features for managing your templates (you can also right-click directly on the template tile itself to access this menu):

Set Selected Template as Default Changes the default template selection to the template you currently have selected. Once set, the next time you call up this window it will highlight that template for you for quick access.

Hide Getting Started Hides the "Getting Started" category at the top of the category list. The functions it provides can also be accessed from the Help

[7] If you change the title, it will no longer be considered a replacement or update to the original template, meaning you'll end up with two of them.

menu at any time, so it is safe to remove the category once you've familiarised yourself with the program. You can also reveal it again using this same option menu.

Import Templates... If you have downloaded templates from the Web or copied your custom templates from another computer, use this feature to import the files into the template system automatically.

Export Selected Template... Useful for sharing your templates on the Web, with other authors or for transferring templates to another working computer.

Edit Info... If you want to modify how the template appears in the template chooser, such as its title, icon, category or whether or not it builds predetermined styles into the new projects created with it, use this command. Refer to Revising Templates (subsection 5.4.4) for further information. Built-in templates cannot be modified.

Reveal in the Finder Reveals the selected template on the disk, using your file manager.

Delete Selected Template When a custom template is selected, you can use this menu item to remove it from your system. Built-in templates cannot be removed.

For those interested in managing these as files directly, rather than using the above commands, templates are stored in Scrivener's support folder (use the Scrivener ▸ Reveal Support Folder in Finder menu command to get here easily), under the "ProjectTemplates" subfolder. Changes made to this folder will be reflected in the template chooser window after it has been closed and reopened.

Return to chapter ⤴

5.5 Adjusting a Project's Settings

Each individual project that you create will have many settings available to it, both as on-the-fly adjustments that can be made using the main application menus (primarily the View menu), as well as those options found within the Project ▸ Project Settings... panel. Refer to Project Settings (Appendix C) for full documentation on the following settings:

— Section Types (section C.2): concerning how documents within the project are categorised into types, such as "chapter", "scene", "subsection", "table" and so forth.

— Label List & Status List (section C.3): set up the label colours and status markers used within the project.

— Custom Metadata (section C.4): design the types of metadata you can use to categories your items in the project.

— Formatting (section C.5): override application defaults for how the text in your project should be formatted within the editor (as opposed to how they will format when you compile).

— Auto-Complete List (section C.6): make adjustments to the auto-completion dictionary used within this project.

— Special Folders (section C.7): setup for both the document templates folder, and where new project bookmark files should be stored (if you would prefer that to choosing where they are stored upon creation).

— Background Images (section C.8): set a backdrop to your composition mode environment, or use a special image for the freeform corkboard background.

— Backup (section C.9): override application defaults for how this project will be backed up (or even if it is).

Return to chapter ⤴

The Binder & its Outline

6

In This Section...

The Binder is the document browser on the left of the main project window (Figure 6.1) where you can organise your work as individual topical pieces, or files. This is going to be like your ring-binder in the real world, where you would stash all of the notes, documents and maybe even the work in progress itself. Everything that represents information in your project, from an idle thought to a formal paper on a topic you are writing about, will be stored somewhere in the binder.

To that end, the sidebar on the left is a fundamental building block in how a new project begins, all the way to being a cornerstone in how you will export your work and close the project down for good (or at least until the next revision!). This chapter will go over the various aspects of using the binder, setting it up for a new project and how best to use Scriveners' features to structure, navigate around in, edit and write your draft.

6.1 What is Outlining?

If like many authors you are accustomed to working in word processors for most of your writing, you no doubt have a few habits and ideas that you'll inevitably bring with you to Scrivener. If you intend to get the most out of what this program offers, there are a few key fundamentals worth learning, mulling over and eventually applying to your approach. Fear not, we aren't going to say you have to learn how to plot out your outline in advance before dipping the quill into the ink pot! The point here is to get a feel for what that framework is, so that you can decide how best to make use of it.

The method that Scrivener provides is a simple means to working with many multiple smaller pieces of text. This user manual, for example, has each section

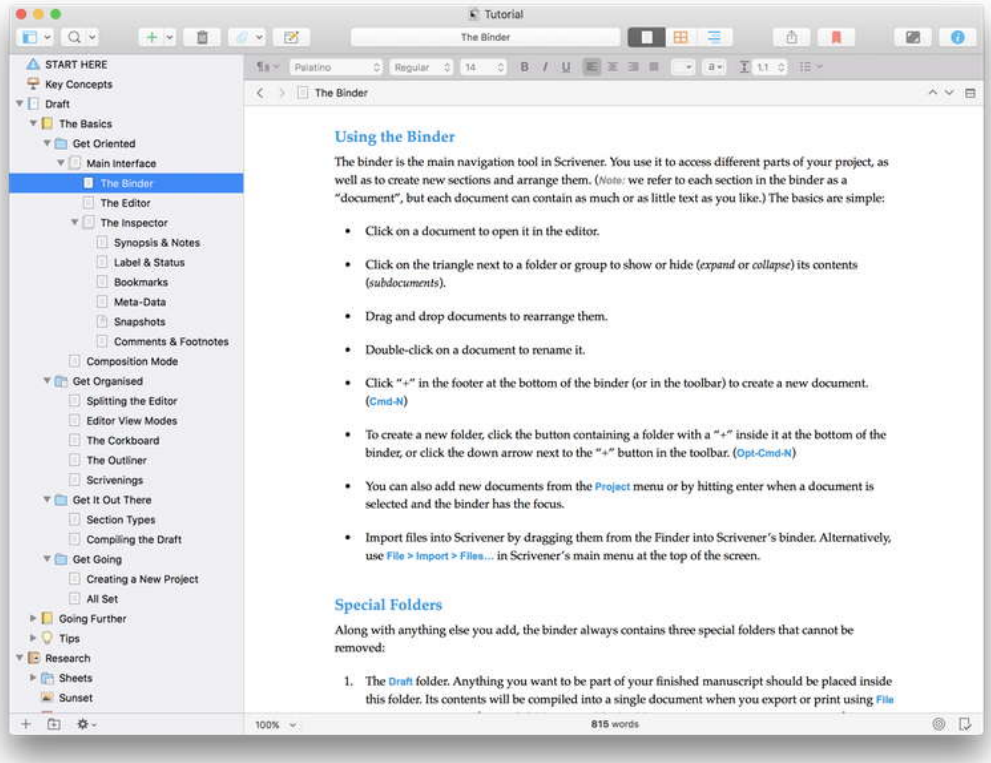


Figure 6.1: A typical project window, with the binder on the left and an editor on the right.

of the manual in its own document, and is comprised of over a thousand such snippets of text. From the heading for this section ("What is Outlining?") on to the next, "The Three Root Folders", the contents of a single document exists, nested within "The Binder and its Outline" section, which in turn is nested within a group called "Learning Scrivener" (Figure 6.2), one part of a larger folder called "Preparation".

While the above example is designed to produce a specific result when exported, we aren't necessarily thinking in terms of parts, chapters, sections and subsections yet, we're merely organising things topically. Later on we can worry about whether "The Binder and its Outline" should be printed as a chapter break and heading. This approach grants you ultimate flexibility in how you think around the structure of your work. Your outline needn't serve some external stipulation while creatively constructing it and writing the text that will become the final document.

If you have never used or even heard of the type of software referred to as an "outliner", there will be some useful jargon to learn, as it will be used elsewhere in this manual to refer to more complicated concepts concisely:

— Outlining is in part a visual metaphor that declares some things as being *children* to other things, based upon their relative level of indent. If some-

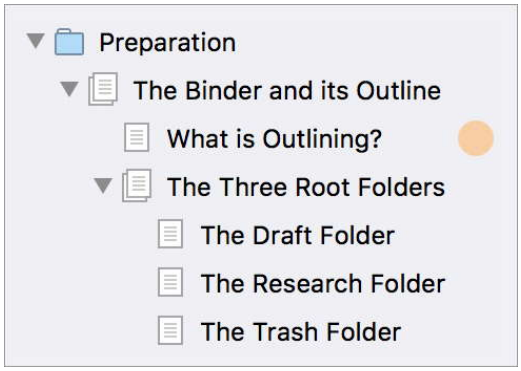


Figure 6.2: An excerpt of the outline used to organise this section of the manual you are reading; current section marked in orange.

thing is indented "beneath" an item, then it is considered a child of it, and that item is a parent to it. This can also be referred to as a *tree*.

— The action of putting items beneath another is sometimes referred to as "nesting". We can also say that indented subdocuments are "nested" beneath another item.

— A child (or a *leaf*) can only ever have one parent.

— Items at the same level of indentation are referred to as "siblings".

— A crucial feature of digital outlining in principle is that you can hide child portions of the outline by folding, or collapsing their respective "parents", keeping your screen focussed on what you are currently working on.

You might draw a connection between this and the outlines we had to compose prior to writing essays in school. Unlike those pen and paper outlines, a digital outline has the benefit of being able to contain the *content*, or to provide a working index to that content, within the outline. You can imagine it as being as though each line on the paper essay had a stack of pages beneath it, where the portion of that essay is represented by the topic line it relates to. Thus, moving topic lines around in a digital outline moves the content *beneath* the outline for you.

Let's move on to some applied examples of the concept, starting with the three basic top level groups (folders, parents or trees) that every project will be working around.

Return to chapter ⤴

6.2 The Three Root Folders

The binder has three default root folders which cannot be deleted or moved from the top level. They can be renamed and moved around among each other, but

not within each other or other folders. To use Scrivener effectively, it is very important to understand the significance of these folders.

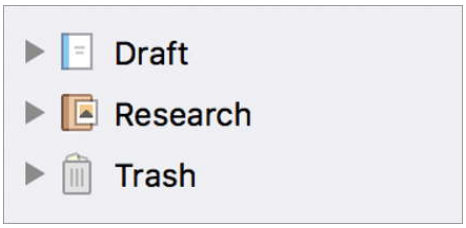


Figure 6.3: The three default top-level folders

6.2.1 The Draft Folder

As the name suggests, the Draft folder is where you place all of the files you want to include in the actual work that others will read. How you structure it is entirely up to you—you may have parts and chapters, or you may have separate files for each scene within each chapter, or even separate files for each individual paragraph if you so wish.

At a basic level, everything that goes inside the Draft folder will be compiled into one long text file when you use the "Compile..." command from the File menu[1]. Thus, the draft folder is central to Scrivener: ultimately, you are aiming to write and arrange everything inside this folder so that each of the elements it contains form an organic whole that can then be output as a singular file to the format you require. It is less useful to think of its contents as *files*, then, and more useful to think of it as a personal table of contents for the text that will become a document when you export.

Another good way to think of the draft is that it represents a long spool of paper or a scroll. In a normal word processor, this entire spool would be presented to you in a single window, with one scrollbar. Moving text around within that spool means cutting and pasting it from one spot to the next. In Scrivener, you can take that long spool of text and chop it up. These chopping points are entirely up to you. The compile feature offers a way to "tape" the whole thing back together into a single spool. Meanwhile the "Scrivenings" editing mode does the same thing, but in a temporary fashion making working in smaller pieces easier, bringing you back closer to that single spool metaphor familiar from word processors.

Both of these "taping" features will be discussed in due course, but for now it is good know that there are no penalties for slicing and dicing up your manuscript into an easy to visualise outline.

[1] As is often the case with Scrivener, there are many exceptions. The contents of the draft folder can be dynamic, only compiling into the final document under certain conditions, never compiling at all or only piecemeal, but for now it is good to think of the draft folder in the binder as a structural take on your document.

My Project Doesn't Have a "Draft" Folder

Some of the templates that we provide with the software have had the "Draft" folder renamed to something more meaningful to that template's purpose. For example the Novel template has the draft folder renamed to "Manuscript". The draft folder can be renamed to whatever you wish. The key thing to look for is the special icon (Figure 6.3). Only one item can have that icon, so no matter what it is called, that is your Draft folder. You can also use the Navigate ▸ Reveal Draft Folder command at any time to bring up the binder and highlight the draft folder, no matter what it is called.

Because the draft folder is what is used to create the final manuscript, it is unique in that it can only contain text and folder files. That isn't to say you cannot include illustrations and figures in your work, they can be inserted into the text editor just as with any typical word processor, but that the listing in the binder can only be text sections, either as a loose list or organised into logical groupings.

6.2.2 The Research Folder

The Research folder is the default import location for non-text documents such as images, PDF files and so on (although it can hold text files too). You can create as many subfolders as needed to organise your research, or you can rename it and create other folders in the root level to hold different types of research or supporting material. Technically speaking, anywhere outside of the draft folder is a valid place to organise research, old drafts, notes and inspirational materials.

As noted, some import functions will target this folder if no other target is provided. If you imported something, such as clipping a PDF to Scrivener from another program's print panel and don't know where the PDF ended up, check in the Research folder first.

6.2.3 The Trash Folder

Whenever you delete a file in Scrivener (by pressing ⌘Delete, for instance, or by selecting the Documents ▸ Move to Trash menu item), the file is not actually removed from the project but is instead moved to the special Trash folder. You can tell at a glance if the trash folder has anything in it, as its icon will appear like an overflowing trash can. Permanently purge all trashed files from the project with theProject ▸ Empty Trash... menu command.

Deleting Only Selected Items from Trash

To permanently delete individual files from the Trash, without fully emptying it, select the items you wish to purge from the project and use the Edit ▸ Delete menu command, or by right-clicking on the selected items from within the trash.

Files that have been placed in the trash will have their icons ghosted. This is most useful where trashed files are displayed in other contexts, such as search result lists or from within Collections of items.

Return to chapter ⤴

6.3 Using the Binder

Now that we've got some theory out of the way, let's get down to brass tacks. Effectively working with the resources in your project is essential in a program that encourages you to accumulate them. While the basics may serve everyone well up to a certain point, if you ever find Scrivener starting to feel a little awkward, you might want to come back to this section and review the various tools available for selection, modification of selection, focus of view and movement of items through mouse, keyboard, menu commands and shortcuts. Managing items and navigating among them is one of the chief areas of this program's design; it has a lot of depth that can be explored, both in its feature set and in how those features can be combined to create powerful workflows.

It should be noted that everything contained within this section is equally applicable to the outliner view in the main editor, unless otherwise noted, and in some cases to the corkboard view as well. We'll get into the details of particular views, and what they can offer over the binder, later on.

Relevant Preferences

This section will refer to a few different options that impact how some of the described behaviour below will work. Unless otherwise noted, all options referred to in this section are located in the "Behaviors" panel, under "Dragging & Dropping".

We aren't going to be exploring every detail of how folders and files work here. If you're looking for a more in-depth look at the items themselves, refer to All About Files and Folders (chapter 7).

6.3.1 Adding New Items

Before you can go about working with items in your binder, you'll need a few to start with. You should now be aware of the three basic *types* of item native

to Scrivener's binder: files, folders and file groups—or what you get when you indent one file under another—let's talk about how to make these things and what to expect while doing so.

Although this is a section about the binder, it is worth noting that you can add new files and folders from nearly every context within the project window. If you are editing a text file you can use one of the methods below to add a new document, and Scrivener will consider the currently edited document as "selected" for determining where the new item will be placed.

Scrivener tries to keep your context persistent: for example if you're writing in a document's Notes inspector pane and press ⌘N to create a new file, you'll end up typing in that new file's Notes pane. If you create a new folder from the main text editor, you'll be able to start creating new cards on its corkboard immediately. Want to give the new thing a name before getting to work? Click into the editor header bar to edit its name, or use the shortcut: ^⌥⌘T.

Something to keep in mind is that naming things in Scrivener might not always be important to do immediately. Unlike files on your system, your items will be given useful "handles" if you leave the name off of it. Read more about this capability in Titles and Adaptive Naming (section 7.3).

The Many Ways to Add Things

Since making new documents is such a vital part of Scrivener we've provided numerous ways to go about doing so, from toolbar buttons, to keyboard shortcuts to simply pressing the Return key. Let's go over the options:

The "Add" button in the toolbar This is a multipurpose button, as indicated by the downward facing arrow on the right-hand side (Figure 6.4). That means if you click on the left side, it will create a new text file. If you click on the arrow, you'll get a selection of available types for this project. By default that will be files, folders and web pages (the latter will be disabled if you're trying to add within the Draft folder area), but if you've added Document Templates (section 7.5) you'll see those listed here as well, and after those you'll see any shared document templates (subsection 7.5.3).

Figure 6.4: The "Add" button on the default toolbar.

Footer bar buttons Along the bottom of the binder sidebar, corkboard and outliner views you will find a footer bar with a few buttons for adding files and folders (Figure 6.5).

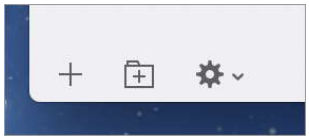


Figure 6.5: Binder footer bar: Add Text, Add Folder and access to the contextual menu.

Contextual menu The "Add" submenu in the right-click contextual menu provides all of the basic options found when clicking the downward arrow on the toolbar "Add" icon. In addition, it also provides quick access to the File ▸ Import ▸ Files... menu command (⇧⌘I), when adding files outside of the Draft folder area.

Menus & keyboard shortcuts There are several shortcuts available for adding new items:

- Project ▸ New Text (⌘N): adds a new text file. In cases where the selected item has Default Subdocument Template (subsection 7.5.2) assigned to it, the name of that template will be printed instead. In the Novel template, if you click on the "Characters" folder to add a new character sketch, the menu command will read, "New Character Sketch".
- Project ▸ New Folder (⌥⌘N): adds a new empty folder.
- Return: by default this will add a new text file (or default subdocument). Note that this key is also used to confirm changes being made to the title of an item, so if you have been typing in the name of one item and wish to make a new one, you'll need to press it twice.
- The ⇧⌥⌘N keyboard shortcut can be used to create the default (first entry in the Project ▸ New From Template ▸ submenu) document template, if your project has document templates enabled.

Dragging text If you select text in any program and drop it into Scrivener's binder, corkboard or outliner views, a new item will be created in that location with the text you dropped. If this is done from within Scrivener's editors, then text will be *moved* into the new file. This is a great way to break apart a larger file: you can scroll through, selecting chunks of text and moving that text to new files wherever you drop them. Where you drop the text will matter:

- Onto a folder or file group or in between binder items: the text will be added as a new file nested within the container you dropped on, or at the location of the drop.
- Onto another file: the text will be appended to that file.

— Onto a folder or file group with the Control key held down: the text will be appended to the group's main text.

— Onto another file with the Control key held down: the text will be added as a new file beneath it, and thus turning the file into a file group with a new subdocument.

The behaviour to move dragged text can be disabled with the **Delete text dragged to other areas** option, in the Behaviors: Dragging & Dropping preference pane (subsection B.4.4).

Adding items with the Touch Bar Last but not least, if you own a Mac with a Touch Bar, one of the default buttons provided universally is the "Add" button, again represented with a "+" sign. With the exception of importing web pages, it essentially replicates the options provided by the "Add" button in the toolbar, when clicking the downward arrow. Refer to the Touch Bar's Global Buttons (section 4.3.2) for more information.

Figuring Out Where Things Will Go

The most basic concept to be aware of is that Scrivener will create new items relative to your current selection in the active view (be it the binder, editor or its copyholder). For example if we select a file in the binder and add a new text file, the file will be created after the selected file. When the focus is in another area of the project window, the selected item in the active editor view will be used. For example, if you create a new item while typing in the Document Notes tab of the inspector, the new item will be created relative to the item being inspected. In cases such as these, Scrivener will attempt to preserve your current focus—in this case, you would go on typing in the Document Notes field for the new item.

When items are nested upon creation, such as when adding a new file to a folder, the behaviour is to add that item to the very bottom of its child list, as the last sibling. If you want to create the item at a specific location within the folder, it is better to select the sibling to place it after.

How Things Are Nested When Added

You may have noticed that sometimes files seem to be nested when you create them and other times they aren't. The logic behind this is very simple, though it is a little unorthodox if you're used to file managers or other outliners. Scrivener's logic is designed to produce the most often desired result, and can be simply understood by what *type* of thing you are creating:

— When creating a folder, the behaviour will always be to add the folder as a sibling to the selected item.

— When creating a file, if a folder is selected, it will be nested as a child to that folder. Most often you'll want to fill up a new folder with files, so this works to your advantage.

— When creating a file with another file selected, then it will be created as a sibling. Again most often you won't want to nest the file beneath the other file, but if you do simply hit the Edit ▸ Move ▸ Move Right shortcut: ^⌘→.

There are a few exceptions:

— When two of the three root folders (section 6.2), Draft or Research, is selected then anything you add will be nested beneath it. This is in part to help prevent cases where one tries to create a new folder or file in the draft, and ends up accidentally adding it to the root level, where it will not compile. If you want to create a folder or file at the root level, then select something other than Draft or Research, or use the Edit ▸ Move ▸ Move Left shortcut: ^⌘←.

— If a group has a default subdocument template assigned to it (like the character sheet folder in some of our built-in templates), and if that default template so happens to be a folder, then in that case the new folder will be a child.

How can I Modify These Defaults?

There are two preferences that adjust how some of these interactions occur, both located in the Behaviors: Folders & Files tab. **Always create new items as siblings** does just what you would expect. No longer will files be nested under folders (however the two exceptions above cannot be overridden). Secondly, **Treat all documents with subdocuments as folders** will make it so all of the above behaviours apply to file groups as well as folders. Now when you select a file group and hit the Return key, the new file will be nested beneath. This setting also broadly impacts how file groups work in general: they will now use group views (like corkboard) when you click on them, among other minor adjustments.

Renaming Items

To rename an item you've created, double-click the title to toggle editing mode. You can also use Esc key to toggle editing mode on the selected item. You can click elsewhere to save the modified title, or use the Return key to confirm your changes.

If a title is left blank, Scrivener takes a dynamic approach to how things are named. You can read more about titles in Titles and Adaptive Naming (section 7.3).

6.3.2 Selecting Items

Any single item can be selected by pointing the mouse at it and clicking, I think we can all agree about that one. While that does the trick for most of the things

we need to do, Scrivener was also built around the concept of selecting and working with multiple items at once. Not only does this make it easy to do tasks with many items simultaneously (say, to move ten files from one section of the outline to another with drag and drop), but the editor itself will react accordingly, displaying those items you select[2]. The particulars of how that works is documented in Multiple Selections (section 6.4), for now we'll just focus on the various tools available for making those selections in the first place.

How does this work with the editor?

In this chapter we are mainly focussing on the binder sidebar itself, rather than how it interacts with the editor. If you want to bridge that gap in your reading, The Editor (section 8.1) begins this discussion in reference to how the editor itself can be used to receive and display items, but the Project Navigation (chapter 12) section is where you will find the most complete reference on how the sidebar and the editor work *together*.

Keyboard Navigation

The four arrow keys by themselves will move the selection around in the outline according to the following rules, and can be augmented with a few modifier keys:

— ↑ and ↓ move the selection up and down the list from one visible item to the next.

— ← will move the selection leftward in the hierarchy, selecting the parent of whatever was selected when pressing the key. The left key will also collapse the selection, if any of the selected items are expanded. If a group (or all groups within the selection) is already collapsed the left key will act normally.

— Adding the Shift key to the ↑ and ↓ keys will expand or contract the current selection, depending upon which way you started when first holding down shift. For example: ⇧↑ will leave the item you started with selected, but add the prior item to the selection as well, while pressing the combination a second time will add a third. At this point, ⇧↓ will contract the selection, or remove the top item from it, resulting in the first two items you had selected. This all works in exact inverse when starting in the opposite direction.

— ^⌥↑ and ^⌥↓ will jump from one visible container to the next, skipping over any regular items in between.[3]

[2] If you ever seen a screenshot of Scrivener with a corkboard then you already know what that looks like.

[3] This behaviour is expanded to include all containers when **Treat all documents with sub-**

— ⌥↑ and ⌥↓ jump your selection to the top or bottom of the outline, respectively. This command, when combined with the Shift key will *select* from the current point to the top or bottom of the list.This is different from the Home and End keys, which will only scroll the view, thus retaining your current selection.

With the Mouse

Selecting items with the pointer (whether it be controlled by mouse, trackpad, tablet or joystick) is a standard set of functions that are broadly useful in many applications. The rules are simple, but can be combined to produce selections of items that would be difficult or even impossible to produce with the keyboard.

— As noted above, simply clicking on an item will select it and simultaneously remove the previous selection, meaning you'll only ever have one item selected at a time when clicking.

— Hold down the Command key to add or remove individual items from the current selection. Like clicking normally, this only works on the thing you click on, one at a time. You can also toggle the selection off for the one item you have selected.

— Hold down the Shift key to select all of the items in between the last point selected and the point where you click. So for example if you click on C and then ⇧Click on E, the result would be to have C, D and E selected. If you then ⇧Click on A, the result would be A, B and C, because C remains the last selected item.

— Given the logic governing these two modifiers, they can be combined. Going from the prior example, if we select A, ⇧Click on C, ⌘Click on E (which resets the last selected item) then ⇧Click on G, the result will be A, B, C then E, F and G selected.

6.3.3 Finding Where You Are in the Outline

When you click on items in the binder and then move over to the editor, the last thing you selected or clicked on will remain highlighted in the binder until you yourself change it. This means that if the editor ends up viewing something other than what you clicked on in the binder (for example, loading an index card so you can write), the highlight will no longer be pointing at the thing the editor is working with.

There are a few advantages to this behaviour, all which work together to maintain the binder sidebar as a workspace that you have control over:

---

documents as folders is set to true, in the Behaviors: Folders & Files preference pane (subsection B.4.5).

— It means you can always get back to where you were since that initial point of navigation is "bookmarked" for you in the binder.

— Since the binder isn't required to always highlight what you are editing, whether folders are collapsed can remain entirely up to you.

— Where you scroll the binder is left up to you.

Under default settings, if the item you are editing is visible in the binder, a smaller secondary highlight will be placed on the item you are currently viewing. In Figure 6.6, we can imagine the author initially clicked on the folder named "The Last Woman Standing" (which we can hope is a working title) and then later went on to write into the subdocuments within this folder—currently in "our heroine enters the arena".

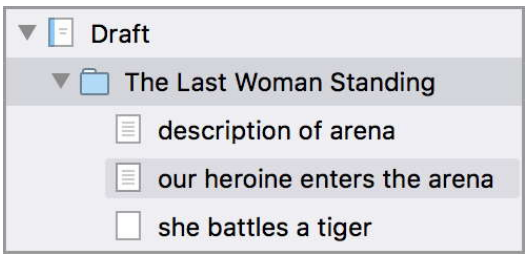


Figure 6.6: The binder highlights what we select as well as what we edit.

This form of highlighting is "passive" in the sense that it won't violate any of the above principles we've outlined. If in our example "The Last Woman Standing" was a collapsed folder and we couldn't see the scene files within it, then there would simply be no secondary highlight. If we had scrolled to another part of the binder, this item would be highlighted—but we'd have to scroll back to the spot ourselves to see it.

If you find this behaviour distracting, the current document indicator can be disabled in the Appearance: Binder: Options preference pane (section B.5.2).

For those cases where you really do want to see what you're editing no matter what it takes to do so, then use the View ▸ Reveal in Binder (⌥⌘R) menu command. This will first reveal the binder if necessary, then switch to the main binder listing (in case you are viewing search results for example), open any folders necessary to display the item and finally move the *primary* selection highlight to the item(s) you have selected. This all-purpose command is useful for finding your place, if you've wandered far away from where you started, locating where search results came from or merely just to synchronise the binder with your current editing session.

6.3.4 Moving and Copying Things Around

Once you've got things selected, you might want to move them from one place to another. Simple drag and drop with the mouse will often suffice, but there may

be times when more precision is required, or in some cases, where the distance between here and there is too far to easily drag. As you might be coming to expect by now, there are a few different tools in this chest.

Moving with the mouse Items can be moved around in the binder with drag and drop. Pick up the item by clicking and holding the mouse button down, then drag the mouse pointer to where you wish to drop the items, releasing the button. The binder will display a blue target underneath the pointer, showing where the drop will end up. Slide the mouse left or right to control the indent level of the drop point, and leave the mouse along the top or bottom edge of the view to dynamically scroll it.

If the place you want a drop a thing is inside of a group that is collapsed, hold the item over it for a moment and it will expand for you. If you would prefer "spring-loaded" folders like this to collapse again after you drop, you can set the "Collapse auto-expanded outline items after drag and drop".

Copying with the mouse Although not enabled by default, you can toggle this capability with the "Option-dragging creates duplicates" option.

Another approach is to duplicate the files with the Documents ▸ Duplicate submenu (⌘D), which will create a copy right alongside the original that you can drag and drop to the intended location.

Moving items with the keyboard ^⌘Arrow keys can be used to move an item around spatially in the binder, step by step. Up and down will increase or decrease its placement in the outline, while left and right will promote and demote the item. For example, selecting a folder that holds the contents of chapter eight, and pressing ^⌘↓ will move it down one spot, swapping its position with the chapter below. It will now be chapter nine, and the other chapter will be eight.

With the Toolbar Although not present in the default set of toolbar icons, you can add item movement buttons to the main toolbar, provided as either a four-way set of movement buttons, or two pairs of up/down and left/right buttons. These operate in a similar fashion to keyboard movement, one step at a time in the indicated direction.

By the icon A bit of knowledge that can save you a lot of time is that everywhere you see an icon in Scrivener, you can drag it, and in doing so it will act as a proxy for that item, just as though you dragged it from the binder or any other view. You might be writing in a file in the editor and realise you want to move the section to a spot in the binder on your left. You could attempt to locate the item in the binder and then drag it from point A to B, but instead you can drag the icon straight out of the header bar directly to B. Done, and you can keep writing without pause.

Moving Multiple Items

If you've selected multiple items, most methods for moving them will gather these items together at the target location, in the order they appeared in the selection (in most the binder order). There are some important exceptions:

— It's an extension of the above, but bears reiteration: if the view the you selected from has been sorted (such as in the search results sidebar or the outliner) then the moved items will be assembled in the target location using that sorted order. For example, if you drag the sorted items back into the folder they came from then the effect is to reorganise the items within that folder.

If that is not what you want, then use the Navigate ▸ Reveal in Binder (Reveal in Binder) command first, and then move them.

— When using the keyboard to move multiple items within a folder (up and down), the relative position of those items will be preserved. In effect, it will be as though each item were moved individually one at a time. Selecting the second and fourth items and moving them up would result in these items now being the first and third in the list, transposing the items they displaced down one slot to now occupy the second and fourth positions.

If the requested operation cannot preserve that relationship, then it will be ignored. You could not select the first and third child of a folder and attempt to move them both "up" because you cannot move the first child any higher in the list than the first position (funnily enough). Use the mouse if you want to gather the selected items in one place within the folder.

— Items can be promoted and demoted (outdented and indented) with the keyboard, and when doing so they will be gathered together. Items must all be from the same container and level in the outline when doing so.

Long distance travel

When you need to move a file so far that the point of its destination is out of sight from the point of origin, sometimes neither the keyboard nor the mouse can be convenient all by themselves. To fill in that gap, we have two methods that should help:

1. The Documents ▸ Move To submenu presents a list of all the items in your binder, each and every one of them a potential target for your drop. The item you select will become the parent for the moved item, and it will be places as the last sibling within that item's child list. Items that are not groups yet will become one by using this command. Once you have moved a file using this menu, a new entry will be added to the Documents menu that will move the currently selected item to the last used move location, with the added keyboard shortcut, ⌥⌘T.

2. If you prefer the mouse for all things, consider that the previous tip also works as a long distance tool as well. Click on the item you wish to move, loading it into the main editor, then scroll to the drop point and drag the icon from the editor header bar into the binder. If that's all you meant to do, remember you can always return to where you were in the editor with the back button, located right the left of the header bar icon.

   It might also sometimes be easier to do things the other way around. You can load a folder into the editor as a corkboard or outline, lock it (subsection 12.2.1), and then freely drag binder items into it. With more than one split available (never mind the history tool making multiple folders easily available to you), it is possible to efficiently work with dozens of remotely scattered targets and hundreds of files. You might never need it, but in case you do, Scrivener does it.

Copying files long distance

You might have noticed, from investigating the menu in an earlier tip, that it is also possible to create a long distance copy of an item via the Documents ▸ Copy To submenu. As with "move to", select an item from the binder that you wish to file the copy under, and it will be added as the last child nested under it, or create a new group out of the item if necessary.

Copying Files and Folders Between Projects

The easiest way to copy a file from one project to another will be through the following steps:

1. Open both of the projects at once.
2. Drag the item icon from the binder, corkboard or outliner view, or anywhere where you can see its icon and it is draggable.
3. Drop the icon into the project you wish to copy the item to, into a location where files can be dropped.

   If you drop into a text editor you'll get a link instead, if you drop into a bookmarks list you'll get a handy link—but if you want to create a full copy in the project, drop it into a corkboard, outliner or binder view.

For cases where both projects are very similar, or came from the same source (perhaps one is a temporary working copy, or you are restoring an accidentally deleted scene from a backup copy) then you should find this form of copying to be seamless, with regards to retaining the original data in the copied files.

When dragging items between different projects, Scrivener will do its best to keep as much of the information intact as it can. If you make use of the inspector

to keep notes on your documents, tag them with keywords, use other metadata, take snapshots and so forth, you will notice that just about everything copies from one project to another. Here are some things that will be created (or copied over, however you wish to look at it) in the new project when copying:

— Keywords

— Labels (both labels and keywords may have new colours assigned to them, for example if the target project has a green label while the source project is blue, the dropped item will turn green)

— Status

— Custom Metadata list entries (if a list field by the same name already exists, but it lacks a specific value in its dropdown menu, an entry will be added if the item requests it)

The following aspects of a file will be carried over:

— Title & Synopsis

— Main text content or media content (for example if dragging an image)

— Any style assignments within the text, provided there are like-named styles in the target project (the formatting will remain for you to update if you wish to do so)

— Notes

— Bookmarks (note that if your drag includes internal links between the dragged documents, these cross-references will be preserved. If you drag an item with bookmarks referring to items not included in the drag, those will be discarded from the copied item)

— Custom Metadata values (if the target project also has those fields by name)

— Custom icon assignment (if the icon is not available to the target project you may not see it, but the assignment will remain and start working once the icon is present)

— Snapshots

— Section type (if a section type of the same name is available in the target project)

— Whether the document should be included in compile.

There are a few things Scrivener *won't* copy:

— Default subdocument section type.

— Default template for subdocuments.

— Project settings that apply *to* that item, such as compile settings, or whether a folder is the project's document template folder.

— Anything not applied to items. If a project has six labels but only five have been used by the items you are dragging, then only those five will be copied over.

6.3.5 Expanding and Collapsing the Tree

Once the number of elements in your binder exceeds a certain point, it may prove beneficial to learn a few of the tools available for controlling large amounts of items. The most recognisable of these is one you've already been introduced to: the "tree" view, or outline, where you can collapse portions of that tree to decrease the amount of information on your screen at once.

The ability to collapse and expand larger areas of the outline, either surgically or in bulk, can mean the difference between getting bogged down in too many details and obtaining focus in your work by keeping only the keystone information you need at your fingertips, and making it easy to find and get to the stuff that's hidden when you need it.

By elements We might as well get the basics out of the way first. To expand or collapse portions of the outline, click on the arrow to the left of any group to reveal or hide its contents. The *state* of every arrow in the binder will always be remembered, meaning you can expand or collapse areas within larger chunks of the tree, and then collapse the whole thing, assured than when you expand it again it will be as you left it.[4]

Use the ← and → keys to open and close the selected container, respectively. The former key has a special behaviour in that if the group you have selected is already collapsed, it will move your selection to the parent of the current group, one level up.

Fully expand or collapse an area of the tree with the Option key held down when clicking on an arrow or using the keys on your keyboard. All subgroups will be impacted by this operation.

Incrementally by level Sometimes fully expanding a section of the tree will be more than you want, but you still wish to expand down two or more levels. This can be done by alternating between two different complementary commands:

1. Press → to expand the tree one level.

[4] The outliner stores its own states, in fact on a per–container basis, making each folder or file group a discrete workspace of its own in terms of what you can see within it.

2. Use the Edit ▸ Select ▸ Select Subgroups menu command to move your selection from the main group to all of the items within it that are also groups.
3. You guessed it, → again to expand the second level. Repeat until the tree is expanded enough for your purposes.

Reveal and hide everything The menu commands "Expand All" (⌘9) and "Collapse All" (⌘0) in the View ▸ Outline submenu will expand or collapse every container in the view at once. Both of these commands work in many areas of the interface where it is possible to expand or collapse items.

Collapse all to current level Working in a similar fashion to Collapse All, this command will collapse the entire outline below the currently selected item's level. Thus if you if have an outline that has six levels of depth, and select a folder on level 3, running this command will completely collapse all items except those at levels one, two and three, which will be ignored.

When multiple items are selected, the first item in the selection will be used to determine the level by which the tree will be collapsed.

Hoisting the Binder

Although not technically an operation that falls under managing the tree outline—where it comes to corralling an ever-growing corpus of information and writing material—being able to focus the binder on only one portion of the larger outliner deserves mention here. If the idea of blotting out everything in your binder save for the contents of one folder (like one chapter in the draft) appeals to you, hoisting is what you're looking for.

To hoist the binder, select one single container of any sort and use the View ▸ Outline ▸ Hoist Binder menu command. When a container has been hoisted, the appearance of the binder will change (Figure 6.7). A header bar will be added to the sidebar, printing the name of the container that has been selected along with a few functions (if you're familiar with collections, you might recognise these):

— The × button marked (a) will unhoist the view, returning to the full binder. You can also use the View ▸ Outline ▸ Unhoist Binder menu command.

— The ↱ button beside the close button will load the hoisted container into the main editor—just like clicking on it as a folder would do in the full binder.

When you hoist a folder, any containers within that folder will be collapsed or expanded in accordance with how they would appear in the normal binder. However any changes you make to disclosure while the view is hoisted will be discarded upon return to the full binder, leaving you free to more fully expand sections of the outline that you'd rather leave hidden by default.

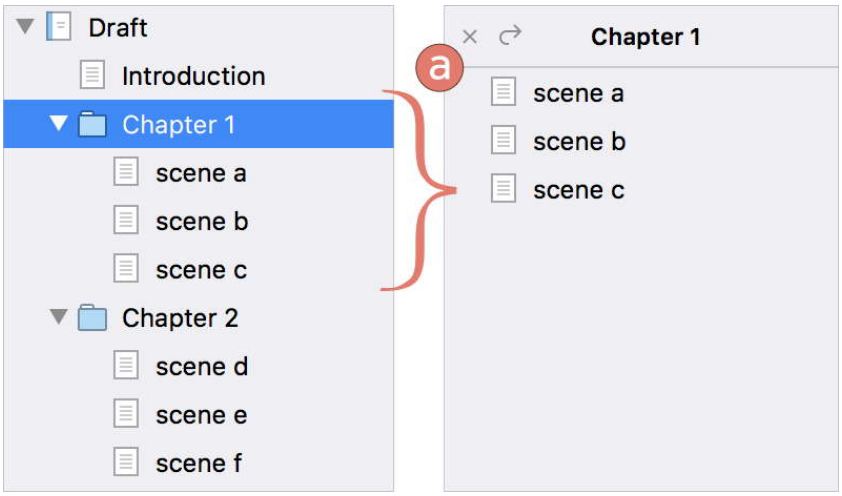


Figure 6.7: Hoisting the binder: before and after.

Can the Outliner be Hoisted?

In a sense, the outline and corkboard views are hoisted by their very nature. When you click on a folder in the binder, their contents (and downward in the case of the outliner) are loaded into the editor view, with the name of the container you clicked on printed in the header bar above. If you wish to hoist *within* the view however, to isolate one folder from the rest within an outliner, or to load an index card as a corkboard within that editor, select the items you wish to hoist and use the Navigate ▸ Open ▸ submenu to select the same editor you're currently using, or press the ⌥⌘O shortcut.

Marking Items as Separators

In longer outlines it might prove useful to add visual separators so that you can easily scroll between them, as landmarks in a large tree. This is done not by inserting a special separator item, but by marking existing items as being separators. The effect is to display this item's title in a bolder font and draw a shaded box around it. This is an item setting, and as such will be duplicated along with the item, or included as part of a document template (section 7.5).

To mark an item as a separator, select the item and right-click on it, choosing the "Show as Binder Separator(s)" contextual menu toggle.

Return to chapter ⤴

6.4 Multiple Selections

A multiple selection is a concept you will encounter throughout this manual, as Scrivener often does useful things with a selection that will deserve remark.

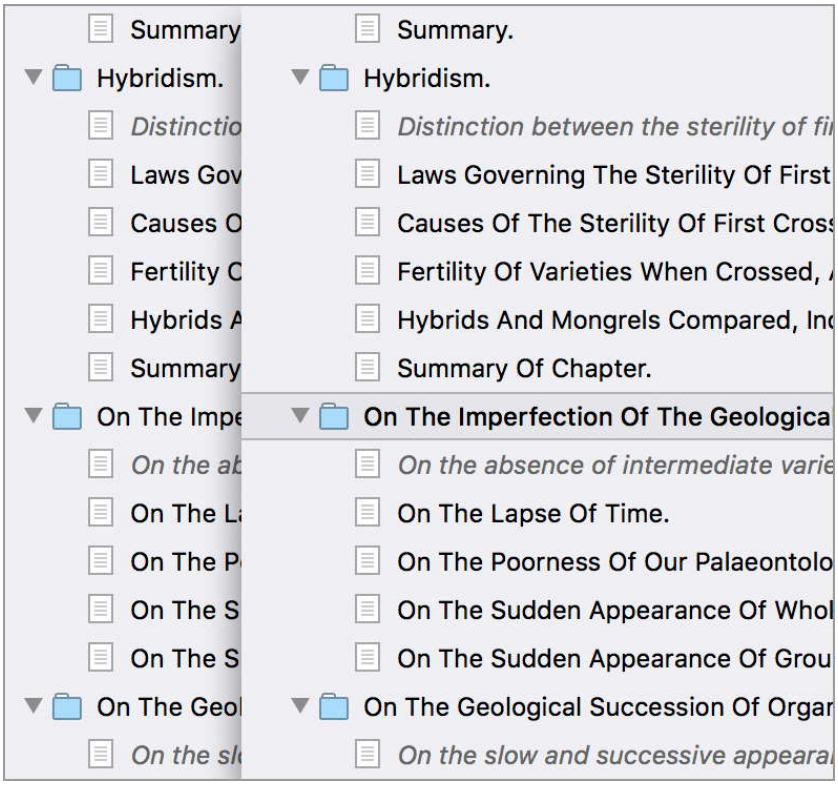


Figure 6.8: Marking a significant entry as a separator makes it easy to find later on.

For example, if you select five files in the binder and then double-click on the icon of one of those files in the editor, the *selection itself* will be inserted into the history queue, meaning you can click the "back" button in the editor header bar (subsection 8.1.1) and return to it later on. You may also note that the header, which typically displays the name of the thing you are viewing, prints "Multiple Selection", with a special icon. You can even drag the icon for it to another editor header bar to clone the selection into both editors.

There are two basic forms of multiple selection which will act differently in the editor:

1. *A selection of containers*: be they folders or file groups, if the entirety of your selection is comprised of containers, then the group views will act as follows:

   — Corkboards will become stacked (subsection 8.2.8), so that each container is listed one after the other.

   — Outliners will show disclosure arrows so you can expand their contents and work with their descendent items.

   — Scrivenings mode will display not only the text of the selected containers, but all of their child items as well. If you have your chapters

organised into separate folders, you can simply select two folders to read the two chapters in isolation.

2. *A mixed selection of containers and files*:

    — Corkboards will display all of the items one after the other in a single grid or label view.

    — Outliners will not allow the contents of containers to be revealed.

    — Scrivenings mode will only work with the text content of the explicitly selected text items.

3. Therefore, if you intend to use one of the above behaviours and are getting the wrong type, then modify your selection accordingly, even if artificially, to achieve that effect. Some might want to just edit the text content of their chapter folders without all of the chapter content, so the solution is to select one token file anywhere in the binder to trigger the mixed selection case.

In a few ways, Scrivener treats a selection like a *thing*, but it is important to understand what they cannot do. What you are viewing in the editor doesn't "exist" anywhere in the project:

— The main editor view will always display a multiple selection in a group view mode. It is not possible to "turn off" a group view mode, like you can with a folder or text group, because there is nothing behind the multiple selection to edit. If the currently preferred group view mode is single-text, then a multiple selection will fall back to Scrivenings mode.

— Freeform corkboard view cannot be used, because your card placements would have no place to be saved to, and would be lost the moment you clicked anywhere else or loaded a file out of the editor.

— You will not be able to add new items to the view, nor re-arrange existing items, because their relationship amongst each other does not necessarily correlate with anything in the binder in a linear fashion.

— Items can however be moved *from* a multiple selection. The result of this action won't be to remove it from the selection, but the card will be moved to the location you dropped it to in the binder or other editor.[5]

— Multiple selections will never display the selection as an indented hierarchy. This flat list approach can be seen as an advantage, as it means you

[5] There is one exception: if you are viewing a multiple selection of containers, you will be able to move items around *inside* the container or even between containers—you just can't move things around at the selection level.

can readily sort by outliner column no matter where the items came from initially.

In all other ways, the editor view can be used as you've learned how to use it. Cards can be filled in on the corkboard, labels can be assigned, you can edit the text of a selection in Scrivenings and outliner columns can be sorted.

Want to Save a Selection?

With as much usefulness as selections have in Scrivener, you might want to save them for future use. To do so, with the items selected, use the Documents ▸ Add to Collection ▸ New Collection... menu command. The collection tab interface will appear above the binder (if necessary) and you can type in a name for your "selection". To recall it in the future, simply visit that collection and select the items within it. Read more about Using Collections (section 10.2).

Return to chapter ⤴

All About Files and Folders

In This Section...

If you're looking for the basics of how to use the binder to create new files and folders, refer to Adding New Items (subsection 6.3.1). In this section we will be taking a deeper look at what files and folders represent and how to use them in your project.

7.1 Folders are Files are Folders

As you no doubt instinctively know from other programs, a folder is a place where you organise files. You can create folders on your disk and then put files and even other folders inside of them to keep everything nice and tidy.

Let's throw away all of those notions for a moment and take a new, fresh look at folders, because in Scrivener, folders are an entirely different animal. Select a folder in one of your projects by clicking on it in the binder. You'll probably get a corkboard view of its child items or perhaps an outliner view if that is how you have things set up. This is all pretty straight-forward, but what happens if you close the group view?

With the folder still selected, click the currently active Group Mode button (you can tell which mode is active by its shaded background) in the main toolbar, the View menu or the respective keyboard shortcut. This will disable the current view, bringing us to a blank view with a blinking cursor that looks suspiciously like an empty text document. In fact, that's precisely what it is. Type in some

Figure 7.1: This folder icon indicates that you've typed text on the folder's text file itself.

text and see what happens: the folder's icon will have changed, now sporting what looks like a little page of paper in its corner](#binder-folder_with_text).

Let's try something a little more radical. Right-click on the folder in the binder and select the command, "Convert to File". The icon will change to a stack of papers. Delete the line you wrote in that item and it will turn into a stack of paper with an empty page in front. Go ahead and right-click and select "Convert to Folder" to get it back the way it was. You didn't lose any information, the only thing that changed is the underlying type of the item, which doesn't mean as much as you might now suspect. When an item has other items beneath it, but it is a file, it is referred to as a file group (or sometimes "a document with subdocuments", when we're being really formal about it). Fundamentally it can behave as a folder does, you can even use the View ▸ Corkboard menu command with a file group selected and view its child items as index cards. This documentation will often refer to either folders or file groups as "groups", when the distinction does not matter.

We've played around with a folder a bit, but what does all of this mean for files? Try this:

1. Select one of your files in the binder.

   You should see its text appear in the editor as you always do.

2. But let's do the same thing we did earlier with the folder, toggle corkboard view. This time instead of toggling it *off*, we're toggling it *on*.

   It will be empty, but you are now looking at the corkboard for that file.

3. With the focus in the Corkboard, add a new document to it.

   An index card should appear, just as if you were adding items to a folder. In the binder, you'll notice that the icon for this file has changed to the aforementioned stack of paper. You can do this with any kind of file by the way, you can even organise PDFs on one master PDF's corkboard.

What is all of this flexibility good for? You don't have to be thinking in terms of constrictive structural elements (like chapters and sections and scene) as you flesh out the skeleton for your book from fragments of text and snippets of dialogue. You can add items as you go, build out new corkboards and change things to folders or back to files once a structure begins to emerge—it's up to you. Scrivener lets you work the way you find most comfortable.

There is no need to build a strict outline if you've always worked text-first, but if you like to work in outlines, or use the snowflake model of expansion, you can accomplish these with ease. An outline can literally emerge out of your book, or

the book can be built onto an outline, even a mix of the two. If working in strict structures like chapters is what keeps you going, then go for it!

In Scrivener, the concept of what is a group is fluid. It's an important concept to grasp, because organising your book will inevitably mean a hierarchy of documents, and wherever that hierarchy takes you it will take so in the form of folders, or their cousins, the document stack. The concept of hierarchy is important, because it means you can keep the parts of the book you aren't currently working on tucked away, and that means you can feel free to break things down as far as you want, no matter how many hundreds of items you make in the binder, if you sort things into folders (or file groups!) you can always keep the clutter at bay.

Return to chapter ⤴

7.2 Binder Icons

You've seen a small taste of this in the preceding section, but there are many binder icons used to show the type of item, and in the case of folder and text documents, hints about their status as well. (Table 7.1) A new document that you've just created will look like an empty sheet of paper until you add a synopsis to it, then the icon will change to an index card. If you start typing into the main text editor, the icon will then show a sheet of paper with lines on it. Likewise, folders will display if they have synopsis or text content in the form of a badge. This can provide a vital clue into where you should look for content.

Some people like to develop their outline in stages, first building the skeleton, then fleshing out the longer descriptions in the synopses fields, and finally starting to write text into the pieces of the outline. These indicators can help you see at a glance how far along into the process you've have gone, and what areas of the draft need more work.

Return to chapter ⤴

7.3 Titles and Adaptive Naming

The title of an item is featured prominently in every location where items are represented together, even in the editor header bar. There are many ways to change the title of an item:

— In the binder, double-click on the item to edit its name. You can also use the Esc key to toggle editing mode from the keyboard on and off.

— When viewing the item in the main editor, Quick Reference panel or a Copyholder, you can click into the editor header bar where the title is

Table 7.1: File and Folder Binder Icons

Icon	Explanation
	Text Document Icons
	An empty text document.
	A synopsis has been added to the document, but nothing has been typed into the main text.
	This text file has text content.
	Scriptwriting documents.
	These variations of the above icons indicate that the text item contains subdocuments, child items indented beneath it, in the outline.
	Folder Document Icons
	A standard folder document.
	The small index card badge means this folder has a synopsis added to it.
	A printed page badge indicates the folder has main text content.
	Snapshot Indicators
	Any of the above icon variations can be "dog-eared". When an icon has a flap folded over in the top-right corner, that means the item has snapshots (section 15.8) saved for it.

printed to edit its name (^⌥⌘T).[1] The Return key will bring you back to where you were typing in the editor.

— In the inspector Notes Tab (section 13.3), the depiction of an index card at the top of the pane can be freely edited.

— If you would like to set the title by using some existing text in the editor, select the text and use the Documents ▸ Auto-Fill ▸ Set Selected Text as Title menu command (⇧⌥⌘T).

You might be wondering if Scrivener requires you to name each and every item that you put into the binder. As you might expect from such a leading question, the answer is no! In fact it can often be advantageous to leave the title off of an item entirely (we even have a command for retroactively nuking titles, with Documents ▸ Auto-Fill ▸ Clear Titles).

[1] If you are viewing multiple items in Scrivenings mode, only the latter portion displayed after the colon is editable.

Each item will attempt to give itself a meaningful name based on what information you have given it thus far. If you've only typed in a little text into the main editor for example, the first few words of that text will be used to identify the item in lists, index cards and so forth.

Upgrading from Scrivener 2

If you've been using the iOS version of Scrivener, you may have already seen this new behaviour in action know how it works, so you can probably skip this section; all you really need to know is that Scrivener on the desktop now supports that method fully, and you can more freely leave items without titles in all contexts.

Adaptive naming works by checking for available text in a few different places, using a descending list of sources. If a source is lower on the list, that means anything above it will be displayed instead:

1. *The title*: obviously, if you have given an item a name, we'll use that name wherever we can.
2. *The synopsis*: if an item has a synopsis typed in, then the first line of its text will stand in for the title in those places where the synopsis is not otherwise shown.
3. *Main text content*: any text found within the main content area will be used if the item has neither of the above. If your text files start with a heading, this can mean not having to worry much about meaningful names at all.
4. *Placeholder text*: finally, if an item has none of the above typed in, then it will simply print "Untitled Document" or "Untitled Folder" in all places other than the corkboard, which itself will display an entirely blank card.

How this information is printed depends upon the context it is being used within, but in general the rules are simple: if a thing needs to be represented by a line of text (such as in the binder), then the data will stand in for the title. If the view also contains other information (like the synopsis part of the index card on the corkboard or outliner, or the main text itself in Scrivenings mode with View ▸ Text Editing ▸ Show Titles in Scrivenings enabled), then the slot ordinarily occupied by the title will either be left blank or will be collapsed.

On the corkboard (Figure 7.2), titles will only ever be printed in the title area of the index card if you've given the item a title yourself, such as in the top left corner. In the lower left we see the synopsis area filled with text coming from the main editor. If you prefer to use the corkboard as an overview of the text *itself* rather than a place to jot down summaries to yourself, then leaving the cards blank will seamlessly do that for you. Finally on the right we have a card with nothing added to it.

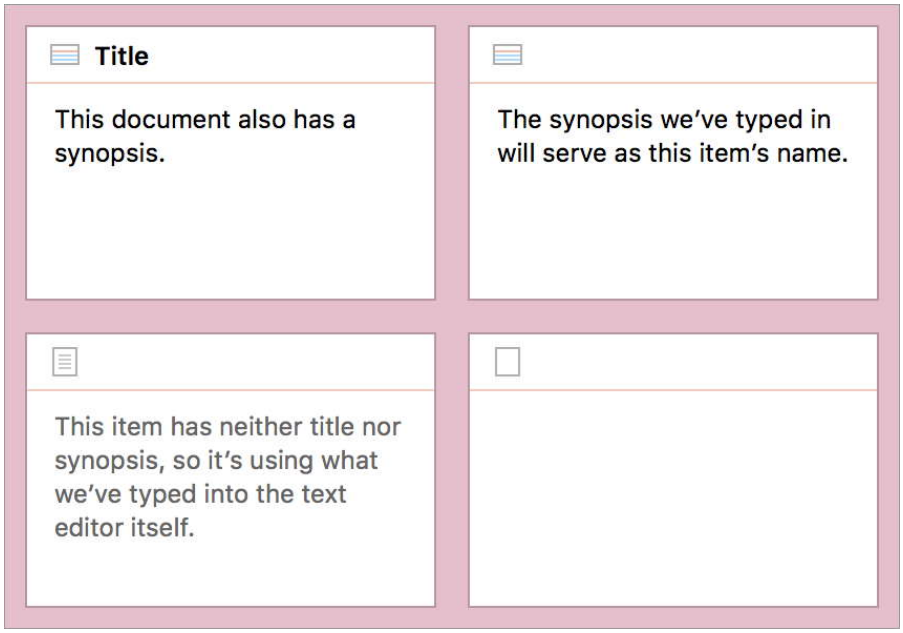


Figure 7.2: Adaptive item naming in action on the corkboard.

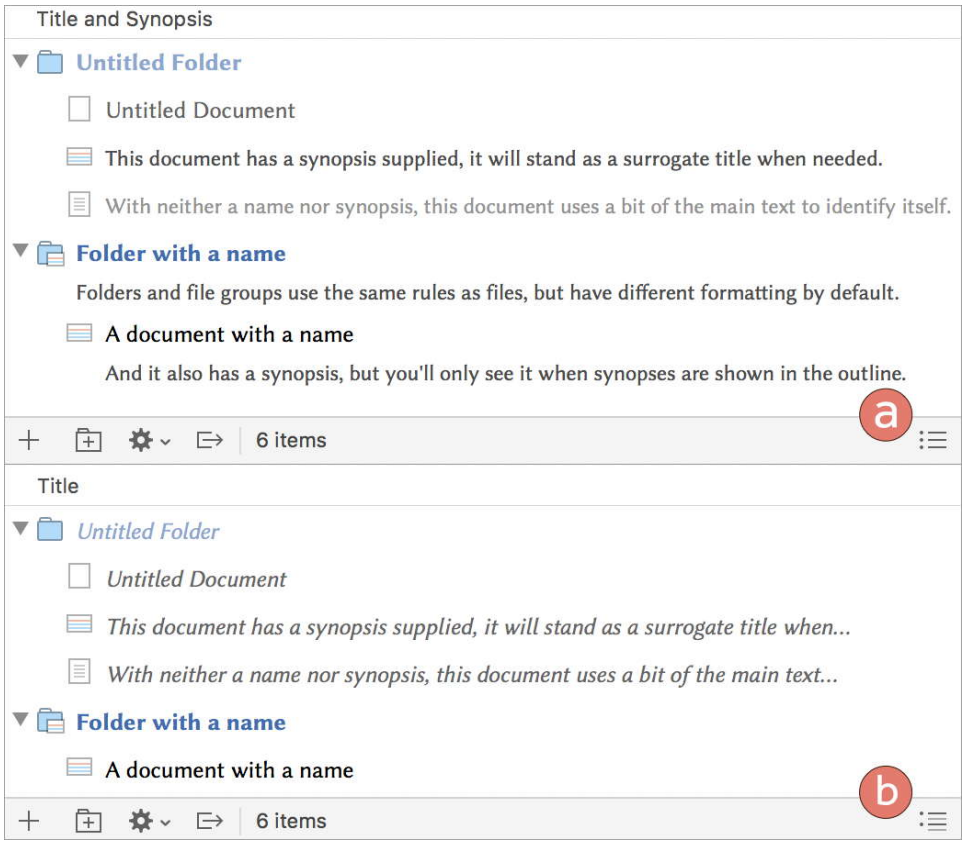


Figure 7.3: Adaptive item naming, according to whether synopses are (a) shown or (b) not.

If you would like to make adaptive synopses permanent, use the Documents ▸ Auto-Fill ▸ Set Synopsis from Main Text menu command (⇧⌥⌘I).

How adaptive naming will be calculated in the outliner (Figure 7.3) depends upon whether synopses are being shown within it. Named documents will use a bolder title by default, with the synopses in a smaller and slightly greyer font. If the item lacks a title, the area where the title would normally be printed will be collapsed and just the synopsis will be shown. Finally the synopses itself will use content from the main text only if the document does not have a title.

The binder, editor header bars, text links and other cases where a thing would only have the title to represent it, will follow the same rules used by the outliner when synopses are hidden. For example, if a document has a synopsis but not title, in a Quick Reference (section 12.6) panel it will display the first few words of the synopsis as a placeholder title in its header bar.

Do Adaptive Titles Export?

Adaptive names will be used when exporting individual binder items via the File ▸ Export ▸ Files... menu command. Otherwise, these are are not real titles, they are meant to be useful tokens of representation while working in the project, but they will not be included in the compiled output as a formal title. If a section is meant to be printing a title, then it should probably have one typed into it, so that Scrivener knows what to export.

Return to chapter ⤴

7.4 Custom Icons

Custom icons can be used to embellish the appearance of your binder items. Anything can have a custom icon, even the three root folders: Draft, Research, and Trash.

There are a few ways you can change the icon for an item:

— Select it in the Binder, Corkboard, or Outliner, and use the Documents ▸ Change Icon submenu.

— Access the item's contextual menu, and use the "Change Icon" submenu.

— Use the quick icon selection menu by holding down the Option key when right-clicking on it.

— While editing or viewing an item main editor using the main Documents menu. When using Scrivenings mode, this will only impact the document that your cursor is within.

— If your Mac comes equipped with a Touch Bar, you can add the Icon button to the binder and group contexts for easy visual two-touch access to the icon list.

— Any of the above methods can be used to change multiple items at once by selecting them first.

The icon menu is organised into two major parts. The first part displays up to five of the most recent icon selections you've made, so you can quickly use them again (if you have never used the feature before, you won't see this yet). The recent icon list is retained between sessions, and is universal to all of your projects. The second section in the menu displays *sets* of icons, grouped together to save space, and finally all of the remaining ungrouped icons that are available.

To remove a custom icon assignment, resetting it to its default binder icon, select Documents ▸ Change Icon ▸ Reset Icon to Default, located at the top of the menu.

Setting an icon over and over?

If you find yourself using a particular icon frequently, you might wish to make a document template (subsection 7.5.1) for it, so that you do not have to constantly re-apply icons to new files or folders. Additionally, since these are all located in the main application menu, you can make use of custom keyboard shortcuts (section A.1) to assign a hot key to a specific icon.

7.4.1 Creating Your Own Icons

In adding your own icons to Scrivener you have the choice of installing them into individual projects or making to them available to all of the projects you use. To add an icon of either type:

1. Load the icon manager with Documents ▸ Change Icon ▸ Manage Custom Icons...
2. Then either:
   a) For global icons made available to all projects, click the + button below the lower table. Icons install this way will only be visible while using this computer account.
   b) For adding icons to the current project use the + button below the upper table. These will be visible no matter where you take the project, but will only be available to that project.

   Naturally, if you install them into both panes then they will be available wherever you go and to every project.
3. Find the graphics file you wish to convert to an icon on your disk with the file chooser, and click the Open button. If the graphics file is too large or the wrong shape, Scrivener will automatically resize it for you.

   You can also drag and drop images into either list to import icons.

Tips for Sharp Icons

After you add an icon to the manager, what you see in the window is what you will see in the binder and other views. If the icon looks distorted or blurry to you in this window, then it will look that way everywhere. You can remove a bad attempt and try again using the following tips for creating sharp and great looking icons:

— Common problems are files with lots of white or transparent padding around them. Open these files in a graphics editor (even Preview will do for simple cropping) and use the software to cut out the excess padding.

— Transparent backgrounds will always be best, so that the icon blends in with the various elements of the user interface where it will appear.

— The best size to use for you icons will be 90 x 90 pixels. This will ensure they look good on all devices and platforms. You can also create small icons that look better on older displays, by sizing them to 16 x 16 pixels.

— Many of the built-in Scrivener icons are a few pixels smaller than that to make space for some of the taller icons. Therefore if you want your custom icons to fit in with the stock icons, it is often best to size them a few pixels smaller with a little transparent padding around the graphic.

Creating High-Resolution Custom Icons

If you intend for your icons to look good on both Retina and standard displays you will need to supply two different icons (16 px square for standard; 32px square (at 144 DPI) for high resolution). Importing a dual-purpose icon is simple:

— Name the low-resolution version normally, like "`Name of Icon.jpg`" (you will see "Name of Icon" in the interface).

— Name the high-resolution version with the same name, only adding "@2x" to the end of the name, like "`Name of Icon@2x.jpg`".

— When both are imported simultaneously into the icon manager, they will be combined into a icon for you. The appropriate version will display depending upon your display resolution.

These icons will not appear on iOS

Although the above instructions will produce the best results on the desktop versions of Scrivener, they will not appear on iOS when created this way. If it is important that you see your icons on that platform, you'll want to use the larger 90 x 90 pixel icon size. The other platforms will need to resize the images down to size and this can result in a minor loss of quality.

Creating Your Own Icon Sets

How you name your custom icons will determine whether or not they are grouped together as a set in the custom icon menu (like the built-in "Flag" set). The format for this naming convention is "Category (Icon Name)", such as "Flag (Blue)". This causes this particular icon to be grouped together with the Flag set and displayed in the menu as "Blue".

Making Icons iOS Compatible

Naturally, icons that are installed on your computer will not appear on iOS if you take your project to that platform. However if you load your icons into the project, it will use them if it can.

The iOS version of Scrivener uses a different size icon than either macOS or Windows versions. This can mean that making an icon that looks crisp and clean between all platforms isn't viable, but a reasonable compromise can be made by using the larger 90 x 90 pixel size and letting each system downsize it dynamically for you.

Graphics that are too small, such as those that came from older versions of Scrivener, cannot be displayed on iOS without considerable loss of quality, and so they will be skipped.

Icons from Other Sources

Using Mac Finder Icons

Many icon packs for the Mac are distributed as Finder icons. If you would like to use an icon that you have seen in the Finder, the easiest way to create a custom icon for it is to follow these steps:

1. Select a file in the Finder that has this icon.
2. Press ⌘I to get info on that file.
3. Click once in the small icon in the upper-left corner of the info palette. You will see a halo surround it when it is properly selected.
4. Press ⌘C to copy the icon.
5. Open Preview.app and press ⌘N to create a new file off of the clipboard.
6. You will probably see several options to choose from in the sidebar. Select the 32 pixel 144 DPI image for retina, 16 pixel 72 DPI version for standard (you'll want both if you are making a hybrid icon as described above)—or if you wish to create an iOS compatible icon, choose the 128 pixel 72 DPI version.
7. Press ⌘C once again to copy the single icon choice; and then ⌘N to create another new document from the clipboard with just this version of the icon.

8. If you chose the 128 pixel version, use the Tools ▸ Adjust Size... command to resample the graphic down to 90 x 90 pixels. Otherwise, you can skip this step.
9. Now you can save this icon to your Desktop or some other convenient location. It is best to use the PNG or TIFF format, leaving the "Alpha" box checked.

Using Emoji for Icons

You can use Emoji, or indeed any character at all including Unicode symbols or even regular letters and numbers, as icons in Scrivener. These will be visible on iOS as well, and owing to their scalable size, will be crisp on all screen resolutions. To create an icon from a character:

1. Select the binder item you wish to apply the new icon to.
2. Use the Documents ▸ Change Icon ▸ Icon from Text... menu command.
3. Insert the character you wish to have turned into an icon. (The "Emoji & Symbols" palette will be opened for your convenience, simply double-click on the icon you want to insert it into the text field.)
4. Click the OK button.

Text based icons will be listed in the Documents ▸ Change Icon ▸ submenu, under recently used icons, and will also be listed in a special submenu, Text Based Icons ▸, which will list all text based icons in use in the project. These kinds of icons can only be added to projects, not "installed" into the system. If you'd like to make them available to other projects you can drag an example item from one binder to another to bring it in—and of course you can add such files to your starter project template so your favourites are always easily available.

7.4.2 Managing Your Icons

Icon Assignments are "Sticky"

It is good to keep in mind that when you assign a custom icon to an item, it will remain assigned to that icon *by name* even if an icon by that name cannot be located. This is what keeps your assignments safe no matter where the project is taken, but it also means that when you rename or delete icons and then later add them back or change their names to what they were, old items that had been assigned to them will start showing these icons again.

Renaming icons

You are not stuck with an icon name if you don't like it, but take the caution above to heart. If you have already used this icon for some time in your projects, all of those items referring to it will lose the assignment (unless you change the name back), as they will only ever be looking for the old file name on the disk.

1. Open the icon manager with Documents ▸ Change Icon ▸ Manage Custom Icons...
2. Double-click on the name of the icon you wish to rename.

Take care to not edit the extension of the icon unless you know what you are doing.

Removing icons from Scrivener

You can remove an icon from either the current project or globally from your computer. This will not strip the assignments to that icon from other projects, and so if you ever change your mind and add the icon back, you'll find all of those items go right back to working the way they used to.

1. Open the icon manager with Documents ▸ Change Icon ▸ Manage Custom Icons...
2. Select the icon(s) you wish to delete (Shift and Command work here).
3. Click the respective – button beneath the table associated with the selected icons.

Copying Icons Between Global and Project

Icons can be dragged between the lower and upper sections individually, or *en masse*. Icons in the lower list will be available to all of your projects; past, present and future. Icons in the top list will only be visible to *that* project; however they can be duplicates of icons that are in the lower list. If you've created an icon for a project, and later decide you'd like to use it everywhere, you can simply drag and drop it into the lower table. If you try to drag an icon with an identical name from one list to another, you will be informed of the collision, and confirmation will be required before it is replaced.

When Scrivener loads your project, it first checks within the project itself for any necessary icons, then it checks the computer. This means any duplicate names (not graphics) in the top list will override any names in the bottom list.

To move an icon *between* projects, you would need to copy the icon to the global list, switch to the other project and then copy it from the global list into that project. You could remove the icon from the global list if that point if you wanted to.

Custom Icon Portability

The final thing to consider is whether or not icons will be visible off of your computer. If you've been adding icons to your computer, in the lower half of the pane, then when you take your project file to another computer all of those icons will not be displayed. They will still be assigned, but since Scrivener cannot locate any replacement icons, they will simply be ignored until you return to the original computer.

So when working on multiple computers, it is good idea to drag your global icon list into the project list above. This way your icons will be available wherever you go, and once you are on the second computer, you can drag them back into *its* global list, installing them on that computer as well.

When collaborating with other individuals, it is a good idea to install custom icons into the project package. This way everyone can see and use them.

Icons on the Disk

Scrivener merely checks a couple of locations for files when working with icons, meaning you can manage icons yourself directly with a file manager if you prefer to do so over the user interface in Scrivener. Global icons will be located in Scrivener's support folder, use the Scrivener ▸ Reveal Support Folder in Finder menu command, and then double-click on the "Icons" folder.

For project-specific icons, they will be stored directly in the project itself. With the project closed, right-click on the project file in Finder and select "Show Package Contents" to view the project's internal folder and file structure and double-click the "Icons" subfolder within the project's ".scriv" folder to view your icons directly. As always, take the precaution of backing up your project prior to editing their files directly.

Return to chapter ⤴

7.5 Document Templates

If you have ever found yourself creating items in the binder and repetitively setting switches to make them look or act a certain way, wondered how our built-in templates have those handy sheets you can use to create character dossiers and such, or duplicated items merely to have a common starting point, then the Document Template feature may end up saving you lots of time with a little setup.

Templates give you the ability to designate a set of binder files and folders that can be used to create new items, much like adding new *types* of files to Scrivener's menus, beyond the basic text files and folders. Much as with the Documents ▸ Duplicate commands, nearly every aspect of the item will be faithfully reproduced when you use the template; here are a few examples:

— The various metadata, notes, keywords, custom icon, bookmarks, synopsis, and of course any text in the main editor. (Snapshots will not be reproduced into the duplicated copy.)

— Groups in the templates folder will create copies of their children as well, for quickly generating whole chunks of boilerplate outline—or maybe even starting you off with one file beneath a folder.

— Settings that have been applied to the item will be carried along, such as its Section Type, or which Type its child items automatically use, which view mode (subsection 12.2.2) a folder might use when you click on it in the binder, and whether it is set to be included in the compiled product.

In short, anything you can do to a binder item can be established as a starting point for new items.

7.5.1 Using Document Templates

As with most things in Scrivener, templates are a feature of your project and its binder. Templates are not something you create separate from your project, or in a panel somewhere—they are normal documents right in the mix with everything else in the project. This means they can take advantage of everything a normal item in your binder can—from custom metadata to label colours to even compile settings.

Setting a Document Template Folder

Templates are designated by placing them into a special folder, marked as being a template folder. Each project can only have one template folder at a time. To set it up, you'll need a folder to start with, it can be empty or already contain a few items you wish to use as templates. This folder can be placed anywhere in the binder except for within the Draft and Trash folders. Once you've created the folder, you will need to designate it as being the official templates folder for this project:

1. Open project settings with the Project ▸ Project Settings... menu command (⌥⌘,).
2. Click on the **Special Folders** tab.
3. Using the dropdown menu in the **Templates Folder** section, select the group you wish to use.
4. Click the OK button.

The binder icon for your designated templates folder will change to indicate its status, and all of the items indented beneath it will have a small blue "T" badge added to their icons. This badge will be present anywhere its icon may be used,

helping you distinguish them from ordinary items in lists like search results or collections.

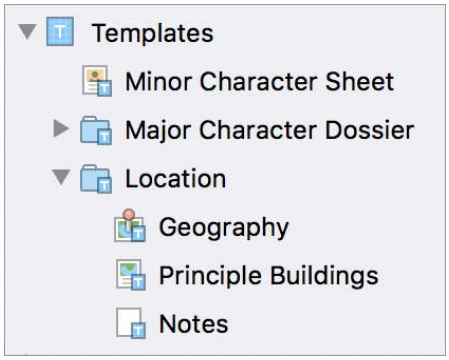


Figure 7.4: Example of a Templates Folder with some file and folder-based templates.

Some of the built-in project templates will come with a template folder already set up for you. Keep an eye out for that special icon; wherever you see it, you can add your own files to it to expand the template capabilities of the project, and change or remove the examples we've provided.

If you accumulate a collection of templates you often use, you might wish to create your own project template (subsection 5.4.3) so that everything is all ready to go the next time you start a new work. You might also be interested in the ability to store template files on the disk, instantly available to all projects (subsection 7.5.3).

Creating Document Templates

Creating new templates is as easy and no different than creating a normal file or folder item in the binder. There is nothing special about these items, save for their position within the designated templates folder. Anything that you add to the templates folder will be available for usage within the project.

Any type of file that you can add to the binder (which is just about everything) can become a template. For many types of files this wouldn't serve much purpose—who needs dozens of copies of the same PDF document—but for editable documents, like Scapple boards or spreadsheet starter files, the template feature can be of considerable use.

Creating New Items with Templates

Once a container is designated as a template folder, and items have been added to it, you can create copies of these files elsewhere in the project via the Project ▸ New From Template submenu, or by using the green + button in the toolbar. The items in this menu will be arranged precisely as they have been organised in the binder, with groups becoming further submenus. Creating new items from templates

will follow all of the same rules and behaviours used to create built-in types like text files and folders.

If you have a Mac with a Touch Bar, document templates will be appended to the Add button, which by default appears on the far right hand side of the Touch Bar in all applicable contexts.

Here are some additional tips & behaviours to be aware of:

— If you select an entire group from the menu, it and all of its child items, will be created in the selected position.

— The order of items within the template folder will determine how they appear in the menu, thus you can organise the menu directly by changing the position of items within the templates folder.

— The first template item in the template folder is a special spot as it will be given a keyboard shortcut, ⇧⌥⌘N. So if applicable, choose the top position for a template you will be using most frequently.

— Document templates have no connection with the item that they were modelled after. It is safe to delete or modify a template without any repercussions on the items that have been created from it.

— If you just want to clean up the template menu, you can move items out of it that are not being actively used. If you even need them again, just move them back in.

It is also possible to assign your own keyboard shortcuts to other items, using the instructions in Custom Keyboard Shortcuts (section A.1).

Clearing Document Template Folder

Use the following instructions to clear the document template folder:

1. Open project settings with the Project ▸ Project Settings... menu command (⌥⌘,).
2. Click on the **Special Folders** tab.
3. Using the dropdown menu in the **Templates Folder** section, select "No Templates Folder", at the very top of the menu.
4. Click the OK button.

This action will not delete anything in your project. It will merely remove the special status from the previously designated templates folder. It is thus safe to switch between different folders, perhaps to facilitate different phases within the project's lifespan.

7.5.2 Default Subdocument Template

Something you might have noticed from a few of the built-in project templates we provide, is that some folders create documents using a specific document template automatically, such as the "Characters" folder in the Novel template. If you create a new file within that folder, you will get a new "Character Sketch", which is of course one of the documents located within that project's document template folder. This is all accomplished using a simple tool that you have access to as well, the Documents ▸ Default Template for Subdocuments submenu.

Assigning a default template to a container will cause all of Scrivener's ordinary methods for text files to be created using the template instead. This extends to all of the actions you can take to create new blank text files, but no actions which would result in an item that already has an explicit identity.

It might be easier to explain how that works with some examples: if you press ⌘N while looking at a corkboard that has a default type assigned to it, or use Edit ▸ Append Selection to Document ▸ New..., then the new card will be created using the default subdocument template. On the other hand, if you import a pre-existing word processing file into this folder, then it will not use a defined default type, but will instead just be ordinary text. Likewise if you create a new folder, use the Project ▸ New Text command explicitly or move an existing item into such a folder you will get precisely what you asked for and nothing will be changed about the items moved into these folders—they have an explicit identity either through already existing or through you have asked for something particular.

To set a default subdocument template, use the Documents ▸ Default Template for Subdocuments submenu to select from the standard template selection menu.

> **Watch Out for Folders**
>
> In case you have assigned a folder as a default subdocument type, you should be aware of a slight inconsistency with the typical behaviour for creating new folders. Ordinarily when you create a new folder with another folder selected, it will be created as a sibling to that folder. In the case where the default subdocument template is a folder, it will be created as *child* and assigned that type of folder.

Cascading Default Templates

Each folder (or file group as the case may be) can have its own default type assignment. Where subgroups are concerned, the default type "cascades" into all subgroups, no matter how deeply nested. The exception to this is when a subgroup within that hierarchy has its own default template type assigned to it.

In the provided illustration (Figure 7.5), a container called "Characters" has been set up to use a "Character Sheet" template as its default subdocument template. Adding a new item to this container will result in the "Character Sheet" template style. If that was all that had been configured, these character sheets would create further character sheets beneath them, because of the cascading

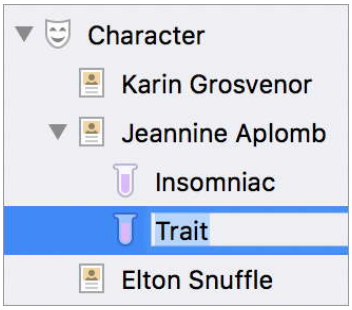


Figure 7.5: A demonstration of multiple subdocument templates in action.

rule. However in this case they have been set up as part of their template definition to use the "Trait" document template for their child items. Thus it *overrides* the default supplied by the "Characters" folder above it. Any new items created within them will automatically be "Traits".

This demonstrates a more complicated example of this feature's usage. Most often, you'll just want to create a particular type of item within a certain container and that will be it. But if you wish to set up more complex arrangements, remember that the subdocument default setting is itself something that will be duplicated along with the rest of the template type, and thus allow quick scaffolding of common types. Another practical example might be Part folders that automatically create Chapter folders with a blank scene, and those Chapter folders automatically creating new scenes within them.

7.5.3 Shared Templates on the Disk

For cases where you would like to provide a set of default file types for all of the projects you use, the shared templates folder might be your best bet. This feature keeps watch of a specified folder on your disk, into which you can place any files that are valid to be imported into Scrivener.

1. In the General: Templates preference pane (subsection B.2.7), click the Choose... button.
2. Select the folder you wish to place your shared template files within.

   You should now see any files you place into this folder appear within the Project ▸ New from Template submenu, at the bottom of the list below any templates found within that project, if applicable.

The important caveat is that unlike document templates that are housed within the binder of a project, they will of course not be proper Scrivener binder items, and their usage mimics more the process of importing a file into the binder. So if the purpose of your template is to set metadata or set up the editor a certain way, these will not provide a solution for you.

If for whatever reason you no longer want to use a shared templates folder, you can click the Clear button alongside the button that reveals the folder in Finderto remove the assignment.

7.5.4 Copying Templates Between Projects

If you would like to copy some templates you created into another project, you can very easily do so by dragging and dropping the templates folder (or individual items from one template folder to another) between open project binders—just like you would copy any sort of binder item from one project to another.

Scrivener will not copy the templates folder *setting* across, so if you are copying the entire folder, you will need to set it up as described earlier in this section (section 7.5.1).

Return to chapter ⤴

7.6 Section Types

Section Types are a fundamental component to how Scrivener works; they directly relate to how your work will be formatted when you export it, in a process we call "compiling". Despite their being the largest scale unit for formatting your work, they are something you will want to introduce yourself to early on, because on *this* side of the project, where it comes to pure writing and not worrying about formatting, all you really need to know about section types is that they are a way of categorising elements of your work into how we will think of them as components of a book, magazine article, film, thesis or whatever it is you plan to write.

To use an example from a novel, we are thinking less of how a chapter break should look vs how a chunk of text in a scene should look, and thinking more on the simple fact that there *will be* chapter breaks and scene text (and probably a few other elements such as these).

Shouldn't the software do this for me?

In a word, *yes*. All of our built-in project templates have been designed with a full spectrum of types, all wired up to use those types seamlessly in compile. Nevertheless, this will probably be good reading so that if you come to the point where you want to modify those templates or make your own projects, you will know how they work and what to change.

In this section, we will discuss how section types can be used at the item level. We will look into how project-wide settings can be overridden per item, and even for large chunks of items at the folder level. If you're looking for instructions on setting up a project's type settings, you'll want to visit the Section Types tab of

Project Settings (section C.2). For how these section types eventually end up getting used in the compile phase, refer to Section Layouts (section 23.3).

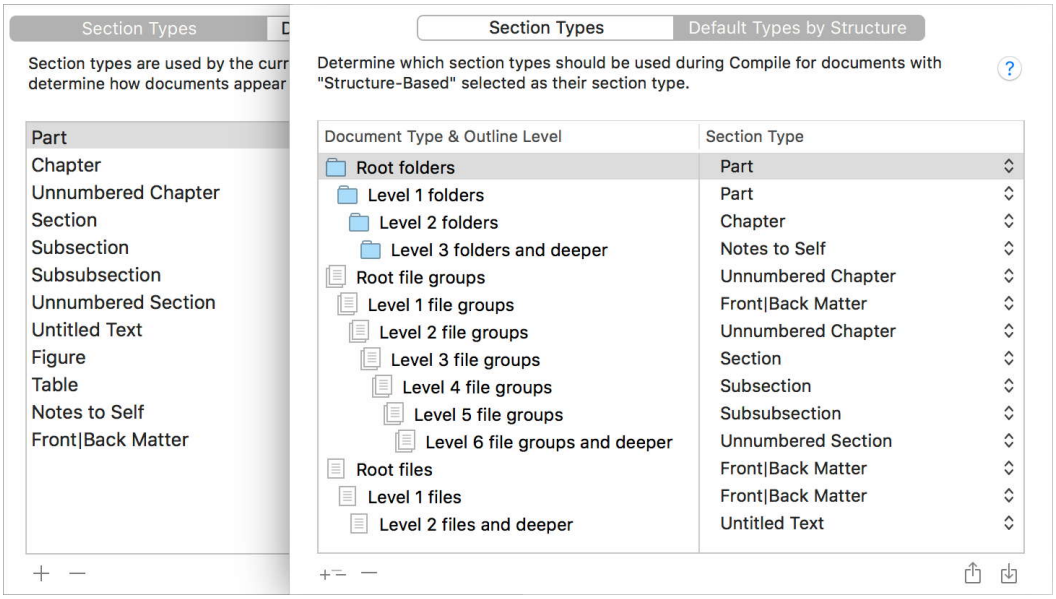


Figure 7.6: Section Types: example of a non-fiction oriented project with sub-sections &c.

Figure 7.6 provides some example settings in a project with several types created in the list within the first tab (on the left). These are the sorts of types you might find in a rather involved technical document, like this user manual. The second tab (on the right) shows how some of these types will be automatically applied to the outline structure of such a project. To point out a few details:

— The "root" level refers to the top of the binder, which is technically outside of the draft. These levels will not typically be compiled, but it is still possible to assign types to them as they may serve some purpose in compilation or other uses (for example you will often keep front and back matter material in separate folders outside of the draft).

— Folders in this example are defined as having three functional levels, "Parts", "Chapters" and "Notes to Self". This project has been designed to make use of file groups for all major and minor sections within the work. We can imagine that the 3rd level folder type would be used to contain drafting and editing notes that will be set up to be ignored by the compiler—not printed in the final output.

— Our file groups start out as front or back matter, and then are used to print unnumbered sections at the same level that chapter folders would exist, giving us the flexibility of having a section type that doesn't impact chapter numbering, and potentially won't even be in the table of contents. Beyond that point, file groups turn into sections, subsections and so forth until finally ending as unnumbered sections again.

— You might notice some types are not applied automatically to the outline, like the Figure and Table sections. These are meant to be applied manually, or perhaps as defined by document templates (section 7.5), as you work.

— Text documents operate as front and back matter sections when at the top of the draft outline (we could imagine a preface being in a single file for instance), and at all levels deeper they print straight text with no titling—thus their separations in the output might even be invisible to the reader, giving you a greater depth of insight into the book structure than is necessary for reading it.

— For a more detailed look into how these settings work and what we mean by "levels", refer to the documentation on the Section Types tab of Project Settings (section C.2).

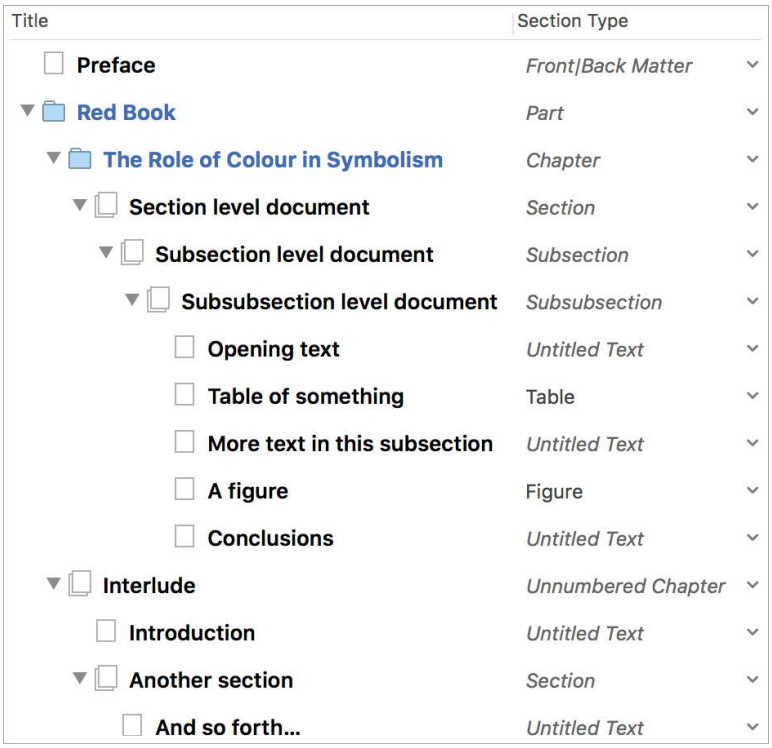


Figure 7.7: Some example binder items using the structure defined earlier.

Now we can take a look at how these settings might work when applied to a little example structure in a hypothetical draft folder, being viewed in an outliner view (section 8.3) with the "Section Types" column visible (Figure 7.7). Recalling that we have the "Part" type assigned to "Level 1 folders", and "Chapter" assigned to "Level 2 folders", we can see where the folder named "Red Book" is on level one and thus acting as a "Part", while the nested folder beneath it on level two, "The Role of Colour in Symbolism" is automatically assigned to act as a "Chapter".

Italicised text in this column indicates assignments automatically made by the project's settings. The two examples in regular font, "Table" and "Figure", have

been manually applied to these items, overriding what they would have been otherwise, "Untitled Text".

Would you like to use these section types as a starting point for your own project, or maybe just to experiment with the settings we've discussed? Download the extras pack (Appendix F) from our website, and look for the file called "1-section_types-technical_book.scrtypes". Refer to the instructions on importing section types (subsection C.2.4) to bring them into your project.

7.6.1 Applying Section Types Manually

There are three ways to change a section type, one of which is suitable for adjusting many items at once. All three methods use the same menu, so we'll take a look at that first:

Structure-Based This is a built-in option that causes the item to inherit its type based on the project's settings. If for example the project is set up so that all folders are assigned to the "Chapter" type, then using this setting on a folder would cause it to act like a chapter—if we converted that item to a file, then it would probably inherit a different section type. When items are set this way, the name of the type they are assigned to is displayed in grey italic text.

If the structure-based assignment is inherited from a parent folder of the item (see **Default Subdocument Type**, below), rather than the global project settings, then the name of the folder making that override will be indicated in the Section Type selection dropdown menu. For example if a folder called "Glossary" overrides the section type layout for its child items to "Glossary Entry", then the when examining the section type dropdown for one of these files, it will read "Structure-Based (from 'Glossary Entry')".

Upgrading from Scrivener 2

If you've upgraded your project from an older version of Scrivener, any documents that had their "Compile As-Is" checkbox enabled will be automatically assigned to a "N/A" type. In the updated projects default compile settings, this section type will be assigned to the "As-Is" layout, causing it to act as it did before. Refer to the Section Types, Page Breaks and As-Is (section E.3) appendix for more information on how your projects will upgrade to this new system.

Listing of project section types The next part of the selection menu will contain a list of all section types defined in the first tab of project settings. Choose one to manually assign a type to the item no matter where you put it in the draft outline. Manually assigned types will appear in black regular text, as opposed to grey italic text.

Default Subdocument Type This portion of the menu only appears when viewing a folder or file group. Choosing a defined type will cause all of its child items (and their child items and so on, until another group changes the subdocument default) to be assigned to a particular type, regardless of project settings.

The default of "Structure-Based" here will cause all child items to use either project settings or any default subdocument types set from folders at a higher level.

Edit... Brings you to the Section Types project settings pane (section C.2).

In the Inspector

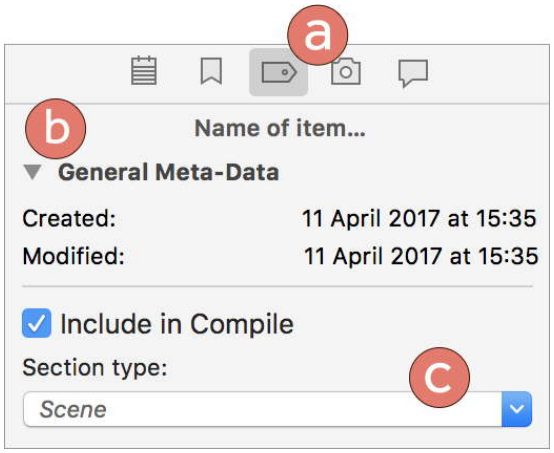


Figure 7.8: The inspector is a handy way to view and adjust the section types of individual items.

a) Click the Metadata inspector tab (section 13.5).

b) Expand the "General Metadata" section if necessary.

c) Click the dropdown control to open the section type selection menu.

In the Outliner

As we have seen previously (Figure 7.7), the outliner view with the "Section Type" column added to it (use the View ▸ Outliner Options ▸ Section Type menu command if you don't see it) is a great way to get an overview of the type assignments for large groups of items. To change an assignment for an item, click into the section type cell for that item's row and make your selection from the menu.

With the Contextual Menu

When you need to change the section type assignment for many items at once, the right-click contextual menu is the way to do it:

1. Select the items you wish to change in any group view or in the binder sidebar.
2. Right-click on the selection, and use the "Section Type" submenu to change the type.

When you have a mixed selection, the "Default Subdocument Type" portion of the menu will never be shown. Keep your selections confined to containers of any type if you need access to it.

7.6.2 Combining Section Types with Document Templates

Although you typically won't have to bother with section types for items outside of the draft folder, document templates (section 7.5) are one place where they can prove useful. When you use a document template with a manually applied section type, the newly create documents will also be so assigned. Here are a few examples of this concept:

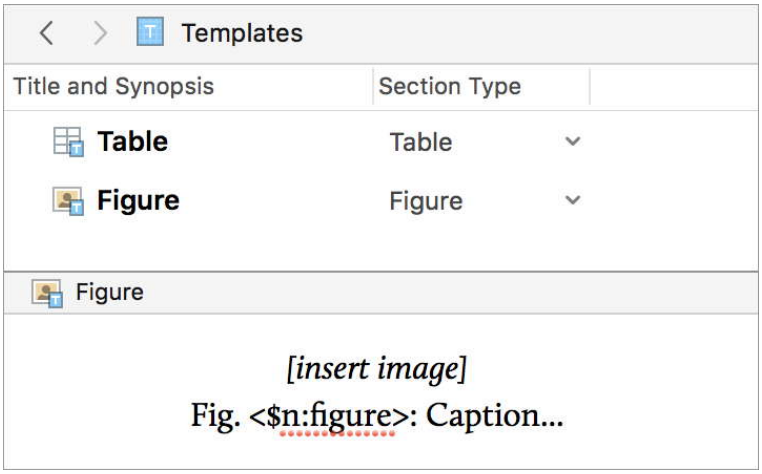


Figure 7.9: A few document templates with predefined section types, including some boilertext, displayed in a copyholder.

— From our earlier example, we had a "Table" and "Figure" section type. Instead of just setting that manually, we could instead use a document template, which would also set a custom icon, a keyword so they can be easily gathered by search and maybe with even a little boilerplate text like a place to paste in the image with a caption all set up (Figure 7.9).

— A folder template might not only have its own section type assignment, but a default subdocument type assigned to it as well. If you typically need areas of your book to function differently from what is standard, this can be a good trick for doing so.

When inspecting an item's structurally-based section type, if it has been set to use a type based on a parent's subdocument setting, that parent will be shown in parentheses for your convenience.

See Also...

— Section Types (section 7.6): a basic introduction to the usage of section types in a project.

— Project Settings (section C.2): how to set up section types in your project settings, and optionally configure how they will be automatically assigned to items in the outline structure.

— Section Layouts (section 23.3): how they will end up being used when compiling your draft to a final format.

— You can search the project for all documents of a particular type with Project Search (section 11.1), and filter the outliner and corkboard likewise (subsection 11.4.3).

— If you'd prefer a more hands-on approach, the Interactive Tutorial, available from the Help menu, also contains a step-by-step guide to building a simple set of section types and then learning how to compile with them. We also have video tutorials available on our site[2].

Return to chapter ⤴

[2] https://www.literatureandlatte.com/learn-and-support/video-tutorials?os=macOS&category=Compiling

Part II

Preparation

> The pages are still blank, but there is a miraculous feeling of the words being there, written in invisible ink and clamoring to become visible.
>
> **Vladimir Nabokov**

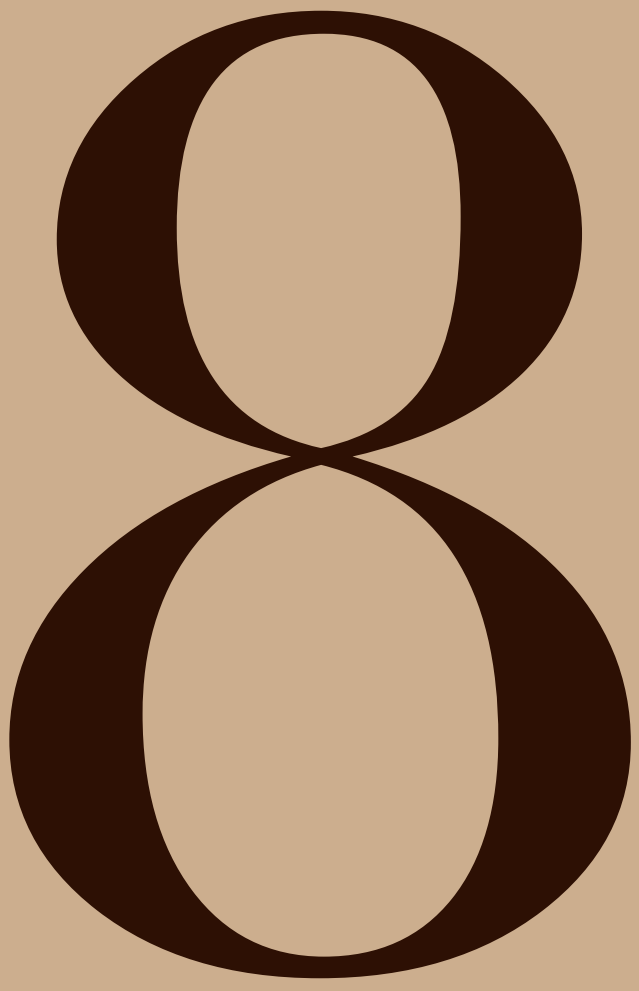
A driving goal behind Scrivener is to be your first stop when a new major project is embarked upon. Much of its design is pitched toward tools that address the early phases of a writing project—when you're still putting together basic ideas and investigatory research and you need tools that can be as vague as you are, tools that can, as you refine your ideas, become richer and more attuned to the intricacies of your work in progress.

Another Scrivener principle is that what works best for your readers and even your editor might not work best for you as a writer. As writers, we see the text in a different way, and we have different demands on how we organise that text. A formal table of contents is great for a reader, but is it the best tool for a writer? What if you could have your own table of contents, one that evolved out of the structure of the work itself, to a level of detail you require and no less or greater? What if you could go beyond formal lists and work with keywords (or "tags" as you might refer to them) as a method for organising work?

The level of fuzziness between preparation and writing in Scrivener is very intentional, because the same tools you use to build up initial structures and ideas will be the tools you use to write, edit, and complete your text. In Scrivener, there is no separation between outline and book order or content. This seamless approach will help you get straight into the writing phase, even while you are still planning and evolving the work. Conversely, you can just easily start writing prose immediately, and let the bigger picture emerge organically out of what you write.

This section will cover the majority of the tools that you may use to accomplish these goals.

The Editor & its Views



In This Section...

The editor in Scrivener is a multi-function viewer, organisational and editing tool; it is where you will be spending most of your time in the software. In this chapter we will first cover the editor itself as a tool, and then dive into its flexible organisation tools, the corkboard and the outliner. For topics relating to text editing itself, head over to Writing and Editing (chapter 15), including coverage on the ability edit multiple text files at once, in Editing Multiple Documents (section 15.10).

Looking for a basic overview?

This chapter is an exhaustive reference of what the editor is capable of. If you'd like a quick summary of its capabilities so you can know which topics are of interest to you and worth further investigation, check out the Interface in Overview (chapter 4) chapter, and in particular its material on the editor (subsection 4.1.5).

8.1 The Editor

Before getting into what one does with the editor in practical terms, there is plenty of ground to cover in the many things the editor *itself* can do. In this section we'll go over header and footer bars, which adapt their controls to what you are working on in the main viewing area itself; how to split the editor in two (and the various ways in which we can integrate those two panes with each other and the binder); how to attach additional documents to each editor with "copyholders" and finally some basics on viewing different types of supported research such as PDF and media.

8.1.1 Header Bar

The header bar appears at the top of each editor pane. It is the primary tool for project and document navigation within the editor, finding out where you are or what you are looking at and controlling split view states of the editor.

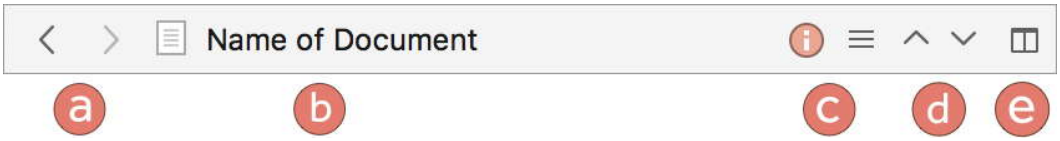


Figure 8.1: The editor header bar

It contains the following functions; we'll go into detail on each of them in the rest of this section:

a) *History*: much as in a Web browser, scroll back or forward through the history of documents you have viewed in this editor. You can also right-click (or click and hold) to access a menu listing of your history.

b) *Document Menu and Title*: the item icon menu can be right-clicked to access convenient functions, and the title of the document to its right can also be edited by clicking into the text.

c) *Inspector lock & navigation*: these two tools are contextual. The first will appear when the inspector has been locked to a specific split (subsection 13.1.1). Clicking it will unlock the inspector. The second tool allows for quick navigation through scrivenings (subsection 15.10.3), and stacked

corkboards (subsection 8.2.8), functioning as a table of contents for these sessions and PDF files (if the PDF has a built-in ToC).

d) *Next/Prev Documents*: these buttons flip through the list of items in the binder sidebar one by one.

e) *Split view control*: this button toggles split view editing (subsection 8.1.4), and also functions as a way to flip the orientation when the Opt key is held down while clicking.

Finding the Path of an Item

When a single item is shown in the editor, hovering the mouse pointer over the the header bar will display the "path" of the item in a tooltip. This is similar to a file path on your system. It will show the lineage of folders the item came from, all the way back up to the root level of the binder.

The visibility of the header bar can be toggled with the View ▸ Editor Layout ▸ Show|Hide Header View menu command.

The Active Editor and Targeted Editor in Split Views

There are two concepts to be aware of when using splits:

— Targeting: the split that is targeted will have a blue background. This means anything you click in the binder sidebar will be loaded within that split; clicking "targets" that split.

— Active: this is the split you are working in, or where you have clicked into the editor last. In some cases that may not be the split that is targeted (by default the two conditions are one and the same). The active split is indicated by an underscore along the bottom of the header bar.

The header bar will use a background colour to indicate different states when necessary (Figure 8.2):

1. Grey: when the editor is not split, this is simply all you'll ever see. If the editor is split, grey is used for whichever split is not targeted at the time.
2. Red: when an editor has been locked, it will turn red to remind you that clicks in the Binder will not load any files into the editor.
3. Blue: when splits are in use, this denotes the *active* and *targeted* split. Any actions taken which will impact the current editor will be made to the blue side, and any typing or commands you use will use this split.

Figure 8.2: Header bar status is indicated by coloured backgrounds and an underscore.

4. Lastly we have the blue "underscore" itself, which indicates that this is where you are typing, where the inspector will be pulling metadata from, and where commands you use that impact the active editor will occur. Since it is grey, it however will not be where binder clicks are loaded.

   In some cases you can end up with an active split that isn't currently targeted by the binder, as shown. For example if you use the Navigate ▸ Binder Selection Affects ▸ Other Editor, then the blue background will always be in opposition to whichever split is *active*. If we used Navigate ▸ Binder Select Affects ▸ Both Editors instead, then *both* splits are targeted, meaning both will have a blue background, but only one split can have the *active* underscore. Read more about this capability in Changing How the Binder Works Persistently (section 12.2.3).

Let's go over each of the different elements in the header bar.

History

On the left side of the header bar, marked (a) in Figure 8.1 are the history navigation buttons, which should be familiar in usage from any Web browser. Use these to navigate backward (⌘[) and forward (⌘]) through the navigation history.

If you right-click (or click and hold) on either of the arrow buttons, a menu will be provided, making it easy to jump straight to something far back or forward in the history without having to go through each point in between.

History From Afar

Editor splits can have their history browsed through from the opposing split even while you are typing in it. This can be a great way to effectively store even more research content into your active session, as you can seamlessly flip backward (⌥⌘[) and forward (⌥⌘]) between sources in the other split while writing in the main split.

With Quick Reference and Copyholders

Although there are no buttons for navigating history in these panes, the menu commands and shortcuts are both available for use. Unlike in the editor, panes will only store their history lists for the duration of the session, but within that session, each Quick Reference window and copyholder pane will remember its history even after you close them.

Copyholders will have a history when you've opened subsequent documents into the same pane over the course of the session. Although one need merely do so manually, this will be most useful when linking its main editor's selection to the pane with the Navigate ▸ Outline|Corkboard Selection Affects ▸ Copyholder option.

In the case of Quick Reference panels, you wouldn't ordinarily have changed the content within the window, but that can be done if you make use of the the bookmark sidebar (section 12.6.2) to navigate between a few different bookmarked documents.

Header Bar Icon

Proceeding left to right, after the navigation buttons you'll see an icon (between markers (a) and (b) in Figure 8.1). This is not merely ornamental, the icon can be dragged and in doing so the action is identical to dragging the item from the binder. You can thus move a file you have open by dragging this icon to another folder in the binder or a corkboard in another split—and any other task that can be done with drag and drop in Scrivener.

For binder items that you are viewing in the editor, the icon can be double-clicked to load the item into a Quick Reference panel.

With binder items, the header bar is merely another place where their icons appear, but since the editor also is capable of showing things that are not documents, there are a few other icons that you may see (Table 8.1).

Table 8.1: Header Bar Status Icons

Icon	Explanation
	Multiple selections (section 6.4) will display an icon in accordance with the current view mode in use: scrivenings, corkboard and outline, respectively.
	The first icon indicates that you are viewing the contents of a collection in the editor (section 10.2.1), the second icon is used for search results and saved search collections.
	This icon will be used when Viewing Snapshots in the Editor (subsection 15.8.3).

Icon Drag and Drop Functions

As with nearly every depiction of an icon in the software, it can be dragged and dropped, serving in most cases as a proxy for the file it represents. Here are a few practical examples of how dragging an icon out of the editor could be used:

— Into the text of the opposing split to create a hyperlink pointing back to the document in this split, or into that document's bookmark list in the inspector.

— Into the header bar of the other split to load it a second time, or with the Option key held down to load the document as a copyholder into either header bar (you can also drag an item into a copyholder's header bar to load it there).

— Into a collection tab to assign it to that collection.

— Into a corkboard view in the other split, or the binder sidebar, to move the document to the dropped location.

There are a class of things that the editor can view that do not formally exist as items in the main project binder. These include collections, multiple selections and snapshots. As for the latter, it cannot be dragged from the header bar at all, but the other two types of things have their own behaviour to be aware of:

— *Collections*: these can only be dragged into other header bars, for cloning that view into the other editor.

— *Multiple Selections*: dragging from the header bar in this case has a very important alternate behaviour: it drags all of the items included in the selection. This makes it very easy to batch assign a selection to a collection, or create a link list of them in a text editor but it can also be used to move items from wherever they may be found in the binder to one central folder or location.

  The order in which the items will be conveyed will relate to their original binder order, or how they appear in the corkboard or outline view without any column sorting applied to the view. If you wish to drag a group of items and have them conveyed in their sorted order you should select them from the outliner view itself.

Header Bar Contextual Menu

The main editor header bar itself can be right-clicked (anywhere except for the history buttons, which have their own right-click functions) to reveal a contextual menu with a number of commands from the main menus, but also some unique commands you won't find elsewhere. The following is a listing of all the commands that are available and their uses:

Reveal in Binder Displays the location of the currently edited file in the binder, opening the sidebar and switching to the binder if necessary (you can also use the ⌥⌘R shortcut). It will also expand any containers to reveal the position of the item if it is nested. When used from the icon header bar menu with a multiple selection, all of the entries included in the selection will be highlighted in the binder.[1]

This is most useful when the method you used to arrive at the current document did not involve clicking in the binder (such as using the history navigation buttons or using a link), or if you are currently viewing a collection and wish to find where the file is actually located in your project outline.

Reveal in Collection If the item is located within any collections (even search results collections), then this submenu will list them. Selecting one of the options in the menu will open that collection in the binder sidebar and highlight the document within the list.

Given that it also finds copies from search result collections, projects with a large number of items or several saved search collections may experience a little lag when displaying the contents of this menu to allow time for each search tab to refresh in the background. If you are only interested in seeing which manually organised collections an item comes from, it will be more efficient to use the Documents ▸ Add to Collection ▸ submenu and refer to which collections are greyed out (as you cannot add an item to a collection it is already in).

Reveal in Other Editor If the item you are viewing in the editor is listed in the *other* editor—for example if the other editor is showing a corkboard with this document somewhere in it—then this command will be available and it will function precisely as Reveal in Binder does, only targeting the other editor's view instead.

Path Operating in a fashion similar to Reveal in Binder, this command instead allows you to select and reveal from the full path of the current document. The top entry will always be the current document; the entry below that its immediate parent; and so on until the root of the project binder is reached. When a selection from this menu is made, in addition to revealing the selected item in the binder, it will also be loaded into the main editor, replacing whatever you were viewing previously.

This command always works on individual files. When a Scrivenings session is in use, the constituent portion of the session that you are currently working on will be revealed in the path.

---

[1] If you wish to reveal an item in the collection you are viewing instead, use the "Reveal in Collection" command.

Go To Document This menu functions precisely the same as the Navigate ▸ Go To ▸ submenu does (section A.7), and is thus provided as a convenience to using the main application menu. If the binder is hidden or you would rather not scroll and open folders, for instance, you can navigate the editor to other areas of the project without having to alter your window layout.

Go To Collection This menu functions precisely the same as the Navigate ▸ Go To ▸ Collection ▸ submenu does, and is thus provided as a convenience to using the main application menu. This submenu contains a list of all available collections in the project. Rather than loading the collection in the sidebar, as would ordinarily be done, this command loads the contents of the collection into the editor, where it can be worked with using any of the group view modes.

If the collection is already being displayed in the sidebar, you can also load it into the editor by clicking on the ↱ button in the sidebar header (beside the × button) . Read more about this capability in Viewing the Contents of a Collection in the Editor (section 10.2.1).

Take Snapshot If the current thing you are viewing in the editor is text, this command will take a snapshot (section 15.8) of the text in the main editor and archive it for future reference. This command always impacts only one text file. When used in a Scrivenings session, only the constituent portion of the session your cursor is currently within will be saved to a snapshot.

View Snapshot in Other Editor or Copyholder The next two submenus provide the same exact contents, listing any snapshots found for the current document. Selecting one of the entries will load the specified snapshot in either the other editor, opening a split if necessary to do so (you can read more about that capability in Viewing Snapshots in the Editor (subsection 15.8.3)). The second menu option will of course load the snapshot into the current editor's copyholder (subsection 8.1.5). Hold down the Option key to load the snapshot with comparison mode engaged (subsection 13.6.4).

You can also drag and drop snapshots into either main editor header bar or into copyholder header bars.

Match Split Documents Opens viewed content in the other editor, in essence cloning the view but without enforcing any view settings. E.g. if you have a corkboard view on the left editor and a Scrivenings session on the right, cloning the left view into the right editor displays the contents of that corkboard in an editing session instead. This command will not be available if no splits are open.

You can also drag and drop the icon into the other split's header bar to clone the view.

Lock In Place Locks the active editor so that no binder clicks or other external navigation requests will affect it. When an editor is locked, its header bar will turn a shade of red (subsection 12.2.1). This can also be done with the ⌥⌘L keyboard shortcut.

Lock Inspector to Editor Refer to Locking the Inspector (subsection 13.1.1) for information on how this works.

Lock Group View Mode When viewing a container or collection group, this command will be activated. It will lock the currently used view mode for this container so that no matter what you change it to at a later time, you will always return to it with the selected view mode (subsection 12.2.2).

Header Bar Title

Continuing along our journey of the header bar, we arrive at the name of the thing being viewed itself, marked (b) in Figure 8.1. As you navigate through the project, the title here will update to reflect the current contents of the editor. In most simple cases, it will display the title of the document you are currently viewing or editing.

When viewing a group of items in a corkboard or outliner, the name of the viewed container will be displayed. For example, if you click on a folder in the binder, the name of that folder will be displayed in the header bar, regardless of which index card or outliner row you have currently selected. In the case of a Scrivenings session that comes about as the result of clicking on a container, the name of the container will be displayed first, in grey text. Suffixed to the container name will be the name of the current document you are editing *within* that session.

Editing the Title

If the presented title is printed in black lettering as opposed to grey[2], you may edit the name of the object it represents by clicking into the text area, making your edits and then pressing Return to confirm the changes and send the cursor to the editor viewer area. You can also use the Navigate ▸ Move Focus To ▸ Header Bar Title menu command (^⌥⌘T) to move the cursor and edit the title without using the mouse.

Inspector Locked to Editor Indicator

The red button to the left under the (c) marker in Figure 8.1 will appear when the inspector has been locked to this editor split, and thus serves as a visual indicator

[2] In Scrivenings mode the title would be both grey and black. You can interact with the black portion of the title field.

of which split the inspector is making use of. You can click the button to dismiss the lock.

Jump to Scrivening or Corkboard

The button on the right side, marked above (c), will only appear under two distinct conditions:

— If the editor is displaying a Scrivenings session (section 15.10), this is the "Jump to Scrivening" button. One entry for each chunk of text (or "scrivening") will be displayed, offering quick navigation within a section, or mere reference of where you are within the overall session.

— When viewing a stack of corkboards (subsection 8.2.8) the tool will generate a list of every corkboard in the list, so you can quickly jump from one to the next or see where you are in the stack.

In both cases the usage of the tool are identical:

1. Click the button to load a miniature "table of contents" for the session. This will be in indented format for clarity, and thus is also a way of checking what level you are currently at within a long session.
2. Use the mouse to click on a desired section, scrolling the view to the place where it begins in the text. You can also use the arrow keys to sequentially move through the session contents.
3. When you are finished using the tool, click anywhere outside of it to dismiss.

Sidebar Navigation

Next, you will find two chevron style arrows pointing up and down, under the (d) marker in Figure 8.1. These arrows provide sequential navigation within the project sidebar (be it the binder, search results or collection list).

Clicking the up arrow (or using the ⌥⌘↑ shortcut) will move you to the document immediately above the current document in the sidebar. The selected document will always be the one immediately above, even if that item is currently hidden and on an entirely different hierarchical level. Clicking the down arrow (⌥⌘↓) will always move you to the document immediately below the current one.

If you are viewing a container as a Scrivenings session, the behaviour of this feature will modify slightly. The nature of selection will still be the same in terms of linear above/below selection, but the current session will not be dismissed. It thus can be used as a form of navigation within a Scrivenings session. There are a few caveats to be aware of:

— This feature is an augmentation of the basic behaviour rather than strictly speaking a form of navigation within a Scrivenings session. The only thing special going on here is that the session is not dismissed. The ordinary rules for what will be selected from the sidebar list still apply, and thus in cases where the contents of the Scrivenings session do not match the sidebar, this function will be disabled.

— Again by the above logic, when you navigate above or below the boundary of the session, the session will be dismissed.

— This capability is not available to multiple selection based Scrivenings sessions (section 6.4). If you wish to navigate within a multiple-selection based Scrivenings session, use the "jump to scrivening" button in the header bar instead.

Split View Button

The last button on our tour, marked (e) in Figure 8.1, is a multi-purpose button for controlling editor splits. What will happen when you click on it will change depending upon the current layout of the application and your last preferred split orientation.

Figure 8.3: The three split button states: split horizontal, vertical and close split.

The function of the buttons as shown in Figure 8.3 are:

1. Change the split orientation to horizontal (with one editor above and another below), opening a split if necessary. You can also use the ⌥⌘= shortcut.
2. The same as the above, only to split vertically (with one editor on the left and another on the right). You can also use ⌘".
3. Return the editor to a single view, closing the *opposing* split. You can think of it as a way of selecting which view to keep around. The keyboard shortcut ⌘' can be used, and whichever editor is currently active will be the one retained.

To see the first or second button state while splits are open, you will need to hold down the Option key. This also has the effect of rotating which split orientation will be used to split the interface when no split is yet open.

See also: Splitting the Editor (subsection 8.1.4).

8.1.2 Footer Bar

Below each editor pane is the footer bar. This is the most dynamic element of the editor, in that it will change depending on the type of document visible and the current editor mode. When a text document is being viewed, for instance, it will display the word and character count along with a dropdown menu for changing the text scale and a button for setting document targets. In script mode, it will display scripting hints and the elements menu. When media is displayed, it shows the current playing time.

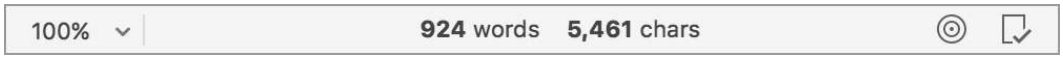


Figure 8.4: The standard text editing footer bar includes: text zoom, statistics, goal trakcing and compile status.

This section mainly focusses on the footer bar's features in text editing mode, as well as general buttons available to group view modes.

— For information on the various other document footer bars, see Viewing Media in the Editor (subsection 8.1.3).

— For details on how to use the footer bar in script mode, see Scriptwriting (chapter 19).

As with the header bar, the footer bar can have its visibility toggled with View ▸ Editor Layout ▸ Show|Hide Footer View.

Group View Controls

Figure 8.5: The footer bar controls for outliner and corkboard.

Both the corkboard and outliner will feature the same cluster of controls on the left side of the footer bar (Figure 8.5):

— The + button creates a new text item—the same as would do when using any of the other methods for doing so, such as ⌘N or Project ▸ New Text.

— As well, the next button creates a new folder in the same fashion as Project ▸ New Folder.

— The ⚙ button provides the same convenience commands found when right-clicking on a card or outliner row directly.

— The auto-load button is used to link this group view with either the other split or the editor's copyholder. See Linking Splits Together (subsection 12.2.5) for more information about that capability.

— Finally the number of cards or visible rows in the outliner is printed for your information. When there are selected items in the view, it will also print how many you have selected.

Read the chapters on the corkboard (section 8.2) and the outliner (section 8.3) for details on how to use the additional footer bar buttons specific to those view modes.

Text Zoom

You can make the text in the main editor larger or smaller without changing the font by using the text scaling pop-up in the left side of the footer bar. See Scaling Text (subsection 15.3.1) for more information.

Quick Text Statistics

In the middle of the footer, the word count of the text you are editing will be displayed in real-time as you type. This counter works for all visible text, even if it would otherwise not be compiled, and will aggregate all text together when using Scrivenings view. This counter will also display the statistics for the current *text selection* if one exists. When the counter is displaying selected text, the colour will turn blue to indicate that it is no longer counting the entire document(s) text.

If you need to track text by something other than words, or would prefer to not track statistics in real time at all, adjust the **Live counts show** option in the Editing: Options preference pane.

Clicking in the statistics area will reveal a pop-over, which will display additional information, such as estimated reading time, but more importantly, provides a quick count without footnotes or annotations—though you can selectively turn either of these back on. The live counter that updates while you type will always include both of these for performance reasons, so if you need to get a quick count without them, use the pop-over. The pop-over also works with a selected range of text.

Counting the Current Section in Scrivenings

If you need to get the the count for the current section you are working on within a Scrivenings session, you can use the selection counting feature mentioned above in conjunction with the Edit ▸ Select ▸ Select Current Text command (⌥⌘A). This will quickly select only the text of the current section you are editing, showing the word and character count in the footer.

Text Goals

When you are editing a single document in standard (not scriptwriting) mode, a small target icon will appear on the right side of the footer bar (Figure 8.4). This

button brings up the target options for the current document where will let you set the numeric word or character goal you intend to achieve with the section. Refer to Document Goals (subsection 20.1.2) for further documentation on this feature.

Included in Compile Indicator

The last element on the footer bar, which appears in text editing mode for all text and group items in the binder, is an indication of whether or not the active document will be included in compile. If the document has a small checkmark in the corner then (unless other conditions exclude it, such as selecting a different chapter as a target) it will be included in the output. A small "X" icon means this document will never be included.

Figure 8.6: Whether a document will appear in the final compiled output is indicated by this icon.

This can also be changed in the inspector's metadata tab (section 13.5), as an outliner column, and in the compile overview itself.

8.1.3 Viewing Media in the Editor

The editor is capable of viewing most of the file types that you will need to interface with for research and creative use, and as such, Scrivener has been designed to work as a central hub for such materials. Being able to load these files into their best environments for viewing or editing is essential.

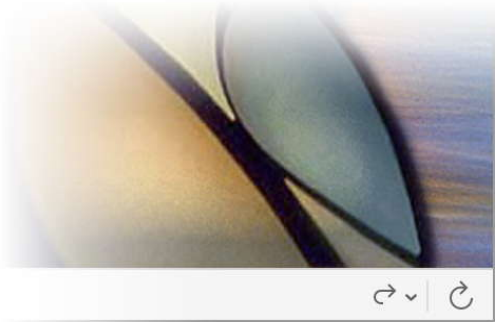


Figure 8.7: The editor footer bar contains useful buttons for external editing.

There are several ways to load an item into a dedicated application:

— With any media file selected in the binder, or from the active editor, use the Navigate ▸ Open ▸ in External Editor command (also available from the right-click contextual menu).

— In the footer bar of the main editor, click the ↱ button to load the viewed item into an external editor Figure 8.7. If you right-click on this button you will be presented with a list of all the installed applications that claim to be able to work with this type of file. Choosing one of these will present you with the option of either:

  – Loading the current item alone in that editor from now on, by clicking the No button.

  – Always loading this type of file in the selected application, from all projects in Scrivener, by clicking the Yes button.

— Changes made to the file will be saved within the Scrivener project directly. If Scrivener is capable of previewing the material in its editor, it will provide a refresh button in the rightmost position of the footer bar.

Viewing Images

The image viewer is displayed in the main editor area whenever an image document is selected or opened. The current magnification of the image will be displayed in the footer bar of the editor, along with the customary controls for external editing and refreshing.

Zooming and Rotating Viewed Images

— Pan around within a large image by clicking and dragging with the mouse; moving the image beneath the pointer.

— Zoom in and out with the standard View ▸ Zoom ▸ commands and shortcut keys.

— Further zoom and rotation settings can be accessed by double-clicking anywhere within the image, bringing up the Image Tools palette (Figure 8.8).

The slider at the top of the palette allows you to **Scale** the image. The button to the left of the slider zooms the image as far out as it will go (that is, makes it as small as possible) and sets the slider to the far left; the button on the right of the slider zooms the image as far in as it will go (makes it as large as possible) and sets the slider to the far right.

Below the slider, tick the **Scale to fit** option to have Scrivener keep the image sized within the editor at all times. This will disable all other magnification commands for this image (including manual zoom).

The **Flip Horizontally** and **Vertically** checkboxes will display the image inverted by that axis.

There are three additional buttons available, from left to right:

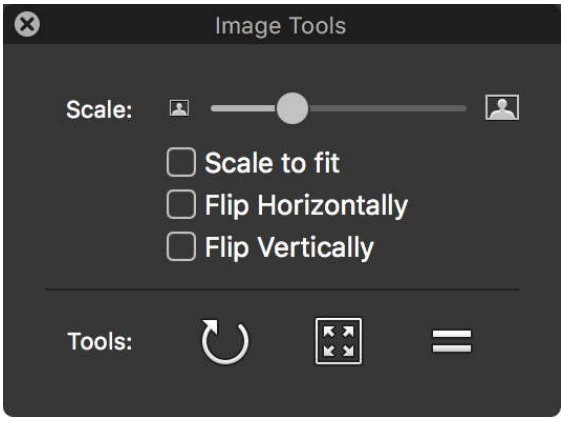


Figure 8.8: Double-click on an image to bring up the "Image Tools" palette.

1. Rotate the image by 90° clockwise (or widdershins when Option-clicked).
2. Fit the image to the current viewable area as a one-time adjustment, leaving you free to change the size manually.
3. Lastly, reset the magnification of the image to 100%.

Viewing PDFs

The main editor serves as a basic PDF viewer, suitable for referencing works while you write.

Figure 8.9: Editor footer bar displaying PDF controls.

The PDF view is displayed when a PDF document is selected. You can control the display of the PDF document via the controls provided at the bottom of the editor, in the footer bar, and with the contextual menu.

— Use the page selector on the left to skip from one page to the next with the arrow buttons, or jump directly to a page by clicking on the paper icon and typing in a number. Unlike the standard scrolling and PgUp/Down keys, these methods attempt to keep each page you navigate to flush along the top.

— Zoom settings for PDF display are not in the footer bar, but you can control the display of the PDF document via the View ▸ PDF Display ▸ submenu, and you can zoom in and out of it using the "Zoom In" and "Zoom Out" controls in the View ▸ Zoom ▸ submenu. You can also access many of these tools via right-click on the PDF display itself.

— In the header bar on the right hand side in a PDF table of contents navigation button, as documented in the Header Bar (subsection 8.1.1).

— The middle of the view shows the total page count (literal) as well as what page you are currently viewing.

— On the right-hand side of the footer bar, the Open in External Editor button (^⌘O) will open the PDF in your preferred reader. If you wish to make edits to the PDF, such as adding notes or otherwise annotating it, you will need to make use of this feature. Simple annotations will be displayed in the viewer (but you will need to use an external reader to view pop-up notes).

— Lastly, if you've made changes to the PDF in other programs, the Refresh button will reload the PDF in the viewer.

Annotating PDFs

As with ordinary text, you can select ranges of text in a PDF file and highlight them with ⇧⌘H. This tool is limited to the yellow colour. You may also strike out portions of text using ⇧⌘_. It is not possible to insert comments into a PDF using Scrivener. To give a PDF file advanced treatment, view it in an external editor via Navigate ▸ Open ▸ in External Editor (^⌘O), such as Preview, or Adobe Acrobat. Any changes made to the PDF, once saved, will show up in Scrivener after you refresh the file display.

PDF Contextual Menu

When right-clicking on a PDF, a number of display options will be provided. These are chiefly documented in the View Menu (section A.6), under "PDF Display". Additionally you will find commands for navigating to the next/previous page, along with some useful commands for copying text, looking up the word in Dictionary.app, or searching the Web.

Viewing Multimedia Documents

The QuickTime view is used to display movie and sound files. From the control interface (which will appear when the mouse is moved within the viewing area) you can play or pause the movie, change the volume, or step backwards and forwards through it.

There are further tools to further aid transcription:

— You can use the ⌘Return shortcut key to pause and resume a media stream, even while you are typing in the other editor, never having to leave the keyboard.

— You may skip forward and backward by a number of seconds with the ^⌘] and ^⌘[shortcuts, respectively. As with the above, these shortcuts also work from the opposing split.

— An optional feature is provided on a per-project basis that can rewind the playback by a set number of seconds whenever paused. Enable the feature with the Navigate ▸ Media ▸ Rewind on Pause menu toggle, and adjust the amount of rewind in the Behaviors: Playback preference pane (subsection B.4.7).

— The time stamp from the active media viewer can be inserted into the text with the Insert ▸ Media Time Stamp menu command.[3]

Viewing Web Pages

The web view displays archived web pages, which will in most cases be a preserved copy of the page at the time it was imported. Even if the page changes live, or is subsequently removed, your copy will be safely stored.[4]

By default, clicking on any links in the page will send the URL to your default Web browser. If you would like to (at your own risk) make it possible to navigate to links through Scrivener, then you can disable this restriction with the **Allow limited navigation in web pages** setting in the Behaviors: Navigation preference pane (subsection B.4.6).[5] The editor history function will consider all Web navigation to be one "frame". To navigate back and forward within the history of the page, use the contextual menu by right-clicking anywhere within the page.

When the web view is shown, the content of the footer will display the original URL a as a link that can be clicked on to open the original page in the system's default browser. Right-click on the link to copy the URL to the clipboard.

You can increase and decrease the size of the web page font using the standard View ▸ Zoom ▸ submenu.

Viewing Unsupported Document Types

Scrivener will let you import files into the Binder that it cannot display in the editor. Unsupported file types will use any Quick Look preview provided by Finder, or the default system icon if available.

Figure 8.10: The footer bar when viewing media: Quick Look, Open in External Editor and Reload.

[3] Modify the time stamp format in the Behaviors: Playback preference pane (subsection B.4.7).

[4] Some pages do not follow Web development standards as well as others, and you may find they do not archive properly or at all, resulting in missing content after a while, or even blank and malfunctioning pages shortly after archival. There is unfortunately nothing we can do about pages that do not load all of their content into the session you are viewing.

[5] Click while holding down the Option key to invert the behaviour set in preferences for that one navigation event.

To view the file in its default viewer:

— Double-click the icon to load the file in the default external editor for that file type.

— Click the "Open in External Editor" button in the editor footer bar, or right-click on the button to select an alternate application (and optionally set it as the default for that file type).

— Use the Navigate ▸ Open ▸ Open in External Editor menu command (^⌘O).

— If you just want a quick preview in a floating window, click the Quick Look icon in the footer bar.

As with supported files, unsupported files are fully managed (copied) into the project package itself.

Reloading Edited Research

When viewing most forms of media the editor will feature a "refresh" icon on the far right of the footer bar (Figure 8.10). Clicking this button will reload the file off of the disk; something you will need to do when editing the file externally. This works for files that have been imported fully into the project as well as those that have been linked from the disk using an alias.

Documents are refreshed automatically whenever you navigate to them in the editor (even if just by going back and then forwards in history).

8.1.4 Splitting the Editor

Scrivener's editor uses a technique known as editor splitting. You may have encountered split screen editing in other applications, but the level of integration and power between the two splits in Scrivener is likely to be unfamiliar—unless you have fond memories of Norton Commander from days of yore. Rather than arbitrarily splitting the interface any number of times and ways, the editor uses a two-way split system designed to be capable of working together, providing one unified way of doing more than one thing at a time.

Splitting Documents

Looking for tips on how to permanently splitting a document into two pieces? Scrivener supports that ability as well, so if that is what you are looking for, you should head on over to Splitting the Document (subsection 15.4.1) for further details. Splitting the editor is a purely visual tool.

If you are unfamiliar with splitting an editor in general, you can think of it as a way of opening a second editor in which you can view the same item you are already working on with an independent scroll position, or another item entirely.

When the same item is loaded into both splits (which is what happens when you create a new split), any edits made to either side will be immediately shown in the other split (in a text editor, your typing will be mirrored; with a corkboard, cards you add or delete will show up on the other side too). Using two splits makes it easy to edit or refer to other parts of the document, or your research & notes, without scrolling back and forth or frequently use the history buttons.

Loading another item entirely into the second split makes reference a snap. While you could use the history functions to jump back and forth between a reference source and the portion of your manuscript that you are editing, splits will present both items to you at once, even allowing you to play media in the second split while recording notes or transcribing it as it plays.

How can I load things into the other split without switching to it?

You may often find yourself writing in one split while the other split is serving as a reference point. You might want to load things into that other split without necessarily switching to it with a click, then clicking in the binder to load the thing, and finally clicking back into the text editor to continue writing—that's a lot of clicking around to do one thing. If you hold down the Option key when clicking on items in the binder they will be loaded in the inactive split—regardless of which split is targeted by the binder. Read more about this capability (section 12.2.3).

In addition to standard file viewing and editing, you can also use splits to do anything else that you would ordinarily do in a single editor. You can mix Corkboard, Outliner, and Scrivening sessions together, and even link the splits so that clicking on cards or outliner rows automatically loads the item in the second split for you.

Horizontal and Vertical Splits

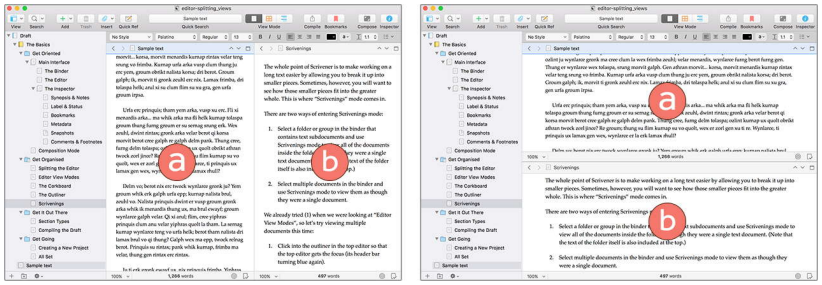


Figure 8.11: The editor can be split vertically (left) and horizontally (right).

The editor interface can be split in one of two orientations. In vertical mode, the divider will be drawn down the middle of the screen, and is most suitable to viewing two text documents side-by-side as it maximises vertical space. Horizontal mode creates a split with content on the top and bottom. In either case,

both sides of the split have the same degree of power in loading various views, media types, and text editing. You can easily switch between orientations by choosing the opposite split type in the View ▸ Editor Layout ▸ submenu, or Option-clicking the split icon in the editor header bar (section 8.1.1).

When the editor has been split, a new editor will be created beside the current one, with matching content. From that point onward you can take that editor in its own direction. Everything that you can do to a single editor window can be done to a split window, and those changes will be remembered for *that side* of the split. An example of this would be view modes. With a single editor interface, if you choose to view containers using the Corkboard model, every time you click on a container or select more than one item, a Corkboard will be presented to you. However, when splits are engaged, each side has its own view settings. This means the left side (in a vertical orientation) can be set to Outliner, and the right side set to Scrivenings.

This means that each side has its own history queue[6], view modes, zoom settings, outliner columns and so on.

You can adjust the comparative size of the splits by dragging the strip along the middle that separates them. This can be done freely, though there are limits to how small a split can be made. To reset the splits to be of equal width or height, double-click the splitter bar with your mouse.

Quickly switching between editors

It is easy to jump between splits without using the mouse, by using ^⌥⌘E and ^⌥⌘R. The former targets the left or bottom editor, depending upon orientation; the latter shortcut targets the right or top split. Additionally the keystroke that cycles between the binder and the editor, ^Tab, will include both splits in the rotation. Hint: if you're rapidly switching between two splits, consider temporarily hiding the binder so that this shortcut simply jumps between the two panes.

Managing Split Views

If you wish to swap the actual position of the material in the editors, so that the content on the left now appears on the right, for example, use the View ▸ Editor Layout ▸ Swap Editors menu command. This swaps *everything* about the editors, including history queue, view modes, display settings and so forth.

To clone the contents of both splits, right-click the document icon in the header bar for the side you wish to clone, and select the "Match Split Document" command. This will duplicate your current view mode as well into the other ed-

[6] Incidentally, if you've been using splits in the project before, the history queue is not lost by closing a split, and consequently you are never really going to lose your place even if you close a split.

itor. For example if the right editor is showing a corkboard for a folder and the left is showing a Scrivenings session of some selected items and you clone the left split to the right, the right split will now be showing an identical Scrivenings session of that selection of documents.

Controlling the Opposing Split

There are a few commands that you can use to impact the editor your are not currently working in, reducing the need to flip back and forth between them. These are located in the Navigate ▸ Editor ▸ Other Editor ▸ submenu, and have shortcuts for handy usage:

— *Remote history access*: Just as you can quickly flip through the history with ⌘[and ⌘], you can cause the other editor the jump back and forth in its own history queue with ⌥⌘[and ⌥⌘]. A nice trick here, when working with a sequence of reference documents, is to "pre-load" them into your reference split by clicking on each one sequentially. Now they are loaded into the history queue and easily accessible while you write in the other split.

— *Remote scrolling*: To scroll the other text editor up and down, use ^⌥⌘↑ and ^⌥⌘↓. This command does not work when the other editor is viewing corkboard or outliner views, or media that would not otherwise respond to scrolling.

— *Controlling media*: While typing in one split, you can start and stop QuickTime movies and audio tracks with ⌘Return, making this setup extremely useful for transcription. If you have Rewind on Pause enabled (section 8.1.3), this shortcut will also rewind the piece by a set amount (3 seconds by default), making it easy to catch up.

If you're looking for ways to integrate navigation between the two splits, for instance so that clicking on a card in the corkboard opens it as a text editor in the opposing split, check out Linking Splits Together (subsection 12.2.5).

8.1.5 Using Copyholders

Much like a document clip that you can mount to the side of your desktop monitor, the Copyholder is a way of clipping a something from the binder to an editor. It will stick until you remove it, even if you navigate on to other things—and navigation events will only ever take place on the main editor, not its copyholder. Only the main editor will be changed if you click on something in the binder or use the history buttons.

This relationship is suggested by the header bar design. Copyholders use a smaller and simpler header bar the is nested within the main editor's. They aren't full editors, and are not capable of showing the same type of information within

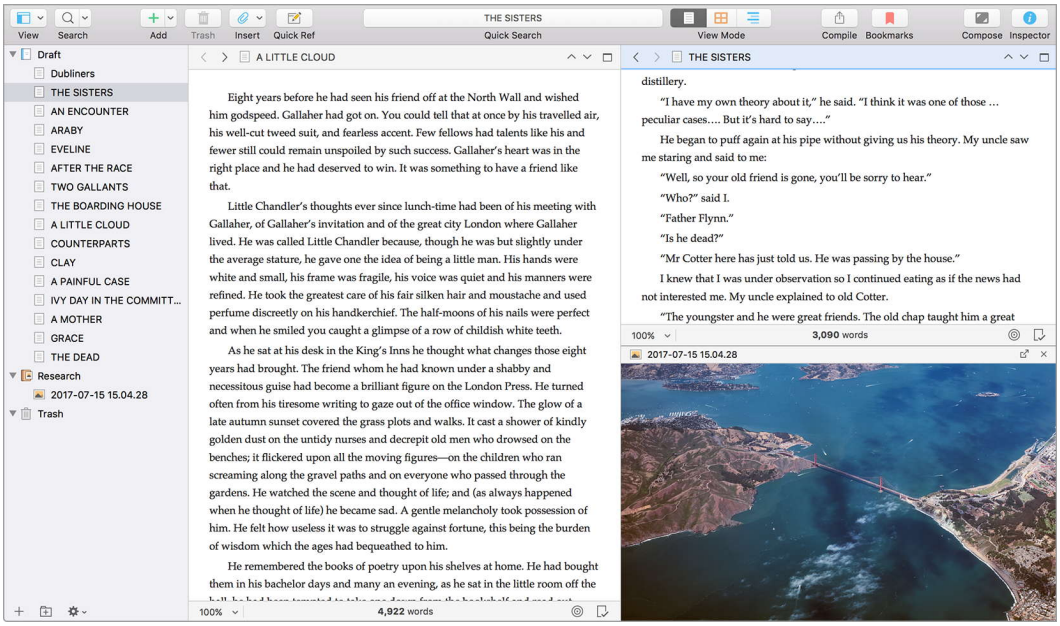


Figure 8.12: Adding to split editor views, copyholders extend how much material you can work with at once in the project window. With up to two splits, each with their own copyholder, you could be working with four items at once.

them. Much like Quick Reference panels (section 12.6), they focus on the content of the thing you are looking at, rather than providing view modes and the many other organisational and navigational capabilities the main editors themselves provide. If you load a folder into a copyholder, you will see that folder's text content rather than a group view of that folder's child items.

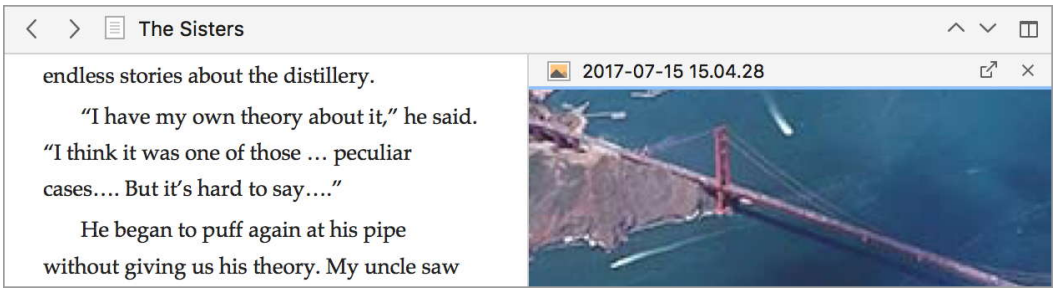


Figure 8.13: Copyholders feature a smaller, simpler header bar, nested beneath its associated editor's.

Copyholders can of course be considered "active" in the same sense that an editor would be, and are indicated in the same fashion. They will never acquire a shaded blue header bar, which indicates binder targeting, but the blue underscore will indicate when a copyholder has keyboard focus (Figure 8.13). In this case, we could zoom in and out of the image with the ⌘> and ⌘< shortcut keys.

While working in an editor, you can move your keyboard focus to its copy-

holder with the Navigate ▸ Move Focus To ▸ Copyholder command (^⌥⌘D). If only one copyholder is open at that time, this will also work from anywhere else in the project window, including the other editor.

Copyholders aren't just about editing or viewing content. Given their nature of remaining as-is until you change their content yourself, they are useful as a way of marking your place; clipping an item to the side of your view so that you can get back to it at any time, or using it to create hyperlinks in other documents via drag and drop from its header bar icon.

Loading an Item in a Copyholder

There are a few different ways to load an item into a copyholder:

— Using the Navigate ▸ Open ▸ in Copyholder menu command. This command is also available as a contextual menu item from an editor's outliner or corkboard views.

1. Select the item you wish to load in any binder, corkboard or outliner view.
2. Use the menu command to load the copyholder into the active split (the one with a blue underline beneath its header bar).

— Click to drag an item's icon, and while dragging hold down the Option key on the keyboard, dropping the item on the editor's header bar.

Most often you would do this from the binder, but nearly every icon you see in Scrivener can be dragged to a header bar, even from a Quick Reference panel—or the very same editor itself (which would have the action of loading the editor's content into its own copyholder, a potentially useful move if you intend to navigate elsewhere for a while but don't want to lose your place).

— Alternatively to the above, hold down the Option key *prior* to clicking and dragging to have the keyboard focus moved to the Copyholder at the conclusion of the drag and drop.

Creating an Item from a Copyholder

If a new text or folder item is created while the focus is within a copyholder, the item will be created relative to the binder position of the item the copyholder is viewing, not the main editor. This will always be done in accordance with the normal rules for placing new items (section 6.3.1).

Changing the Orientation

To adjust a copyholder's position, use the View ▸ Editor Layout ▸ Copyholder Position ▸ submenu, or simply right-click on the copyholder's header bar, selecting one of:

— Left

— Right

— Top

— Bottom

Your choices will be necessarily limited if the editors are split. For example if the editors are split vertically, you would only be able to position a copyholder on the top or bottom of current the editor. This also means that a copyholder might change its orientation automatically to make space for a split if one is opened in the same orientation as it, or when switching split orientations. If a copyholder is along the top of the editor and you split the editor horizontally, the copyholder will switch to the left side of the editor.

You can also of course change the size of the copyholder split. Click and drag on the line between it and the main editor to adjust how tall or wide it is.

Changing a Copyholder's Content

Although a copyholder pane will resist automatically changing its content based on navigation commands made to the editor, you can of course manually change what you are working on. The copyholder header bar is a valid target for dropping item icons to.

Once you have started viewing different things within the pane, the standard history keyboard shortcuts (⌘[and ⌘]) can be used to flip between recently viewed items.

If the document in the main editor has other binder items bookmarked in its Bookmarks Tab (section 13.4), then you can quickly view the contents of these bookmarks in its respective copyholder by right-clicking on the header bar.

Detaching a Copyholder to a Window

To detach a copyholder and convert it to a Quick Reference panel, click the second icon from the right in its header bar. This is a great way to free up space in the main project window without losing access to the content you are working with.

To do the inverse, embed a Quick Reference window into an editor split as a copyholder, there is no one-click trick for doing so, but you can drag the icon from the Quick Ref's header bar into an editor split with the Option key held down, and then close the window yourself.

Closing a Copyholder and Reopening Content

When you are finished with a copyholder and its content, click the × button in the right-hand corner of pane. Closing a copyholder in this way will add the item you were viewing to an internal list of recently viewed items within the current editing session.

If at a later point in time you would like to view the content again, you needn't hunt it down. Open the copyholder pane again using whatever is most convenient (you can simply Option-drag a header bar's icon onto itself for instance), and then right-click on the copyholder header bar to access the "Recent" submenu.

Once you close the project, this list will be forgotten.

Linking Copyholder Content to a Split

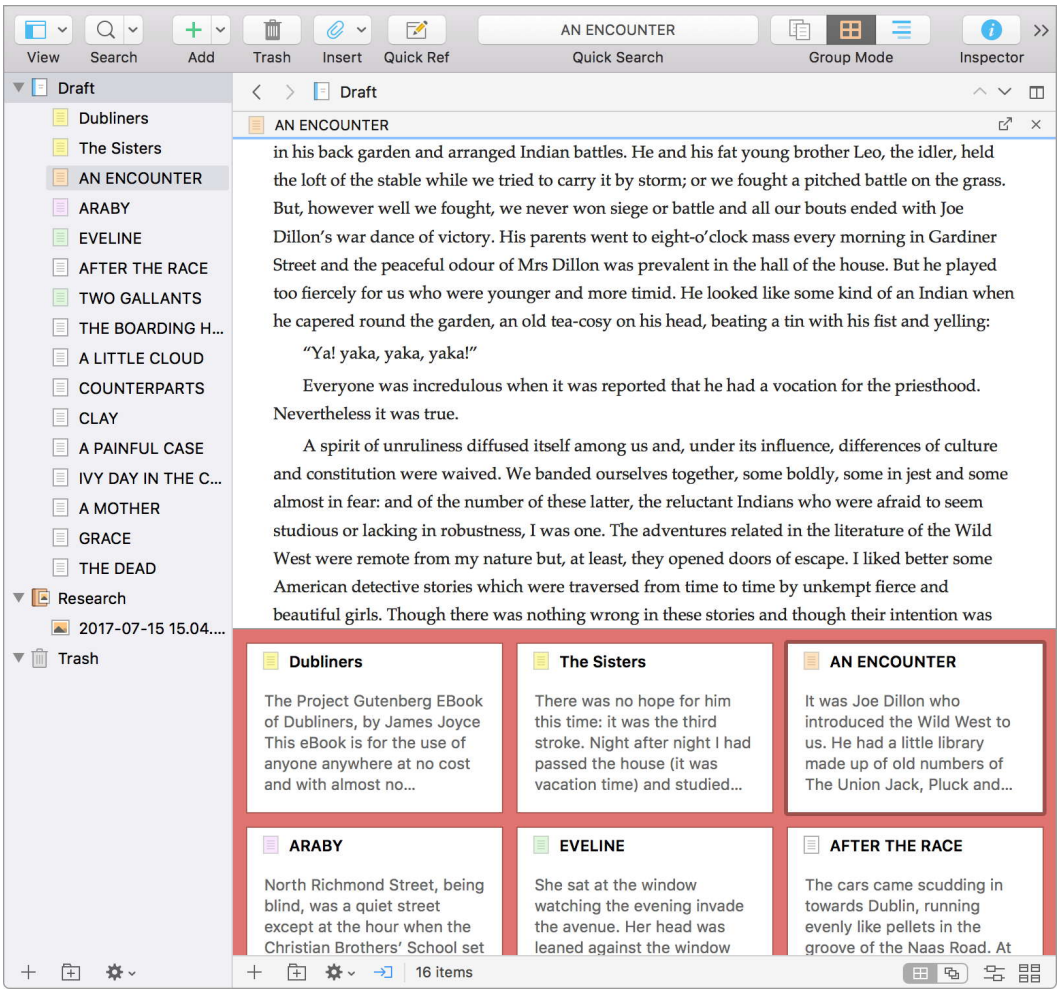


Figure 8.14: The corkboard in this editor has been linked to the copyholder above it. Cards selected in the corkboard are automatically loaded in the copyholder.

In addition to providing a semi-static reference while you work in the primary split, the copyholder feature can also be used to automatically display the content you have selected in its primary editor. You can think of this as being a bit like the relationship between the binder and an editor, where when you click on a file in the binder it automatically loads in the editor. It is a way of forming a

secondary level of navigation in your project—and what sets it apart from using regular splits to do so is of course that you can still use regular splits to do so, in addition. You could have a PDF you are referencing in the left split and a simple list of notes in an outline view in the right split with a copyholder showing the contents of those notes below it.

Read more about this capability in Linking Splits Together (subsection 12.2.5).

Return to chapter ⤴

8.2 The Corkboard

The corkboard provides a familiar, visual way of viewing documents in your binder. You can arrange index cards in direct correlation with their binder order using either a grid view or label view, or alternatively as a freeform corkboard where cards can be freely moved about without directly impacting the structure of the book. You can visualise this as a bit like looking at specimens on a slide. Each slide has a slice of a tree branch on it. To look at a different (whether deeper or higher) portion of the branch, you'll need to load a different slide. The corkboard displays *one* layer at a time, and by clicking up and down in the binder folder hierarchy you can view different layers, or different branches one by one. We will explore the various features in depth, shortly, but first let's take a look at the index card itself.

In Scrivener, every document you create is a document *and* an index card *and* a corkboard *and* an outline. This can be a little confusing at first, for in the real world, an index card clearly cannot also be a corkboard. In Scrivener, though, you can choose to view any document as a corkboard. Each index card on a corkboard represents a document held *within* the selected document in the binder that is represented by the corkboard itself. Another way of looking at it: the corkboard displays the subdocuments of the selected document as editable index cards. At the same time, the document displayed as a corkboard or outline is also an index card (which holds the synopsis of that document), and itself could also be viewed as an index card on a corkboard or a row in the outliner.

If that is as clear as mud, then read on.

8.2.1 So What are Index Cards, Anyway?

A concept that can take some adjustment to get used to is the relationship between index cards and the text of the book itself. You can type titles and text onto the card just like you would the real thing, but there is sometimes confusion as to why that text doesn't get dropped into the book.

A good way of thinking about the index card is as a fancy kind of file icon. Imagine if in your computer you could switch to a way of displaying files that let you type in short descriptions of those files, maybe tag them by colour or keywords and so forth. While it can be a little limiting to think of Scrivener's binder as a list of "files", this metaphor between icon and file, index card and

binder item can be a useful stepping stone to understanding the feature better. When you drag a card you are moving that "file", or that chunk of text or picture or PDF that it represents, to another location.

Let's take a look at what the index card has going for it. There are seven different components available, but depending on your preferences you might not see them all at once.

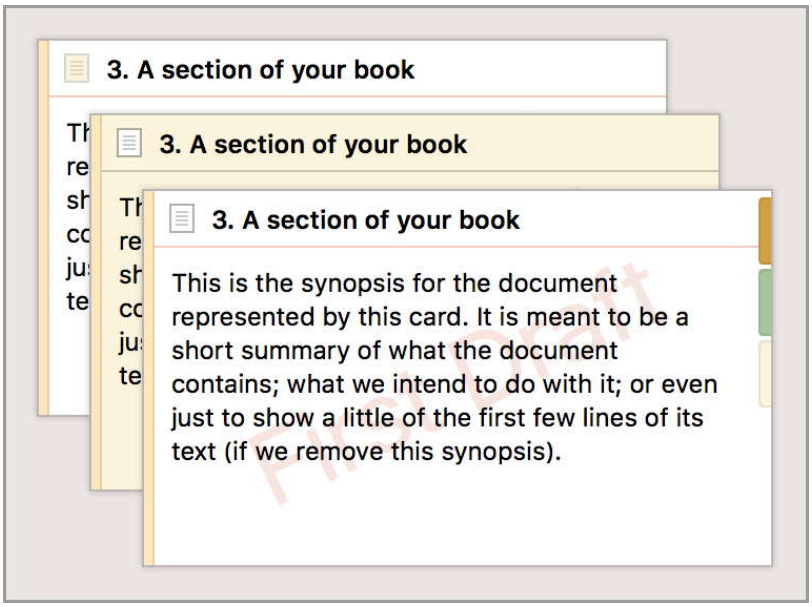


Figure 8.15: A few different examples of how index cards can be displayed on the corkboard.

1. The document icon: you should recognise one of these by now, they are displayed in many different contexts within the project window and they all work the same.
2. The card number: enabled with View ▸ Corkboard Options ▸ Show Card Number, this follows the icon on the first row and indicates the position of this card relative to the other cards in the same folder, or its *siblings*.
3. The title of the document: this area may be left blank if the item has not been given a proper title in the binder or outliner (where it will display placeholder text from the synopsis or main text).
4. Synopsis: the largest block of text on the card. This can also display the first few lines of the document if the synopsis is left blank. The synopsis is not where you write your book! It is where you summarise what you have written or intend to write. Double-click on the card icon to load the card into the text editor and proceed to write your heart out.
5. The label colour: the strip (yellow in this case) along the left side of the card. This can be switched off with View ▸ Corkboard Options ▸ Show Label

Colors Along Edges (^⌘P). If you forget what a colour stands for, hover the mouse over the strip to read its written label in a tooltip.

6. The status stamp: diagonal text ("First Draft" in this case) stamped along the synopsis area of the card. Enable this with View ▸ Corkboard Options ▸ Show Status Stamps (^⌘S).

7. Keyword colours: the pieces of "tape", if you will, along the right side of the card indicate this card has three keywords assigned to it. Enable these with View ▸ Corkboard Options ▸ Show Keyword Colors (^⌘K).

Also shown in Figure 8.15, it is possible to tint the colour of a card based on its label, as well as the icon itself (but that has ramifications everywhere the icon can be seen). These options, among others, can be explored from the View ▸ Show Label Color In ▸ submenu. All of the options you select for corkboard appearance and label tinting will be saved into the project, so if you prefer a certain look by default, consider creating your own starter project template. Index card appearance settings can also optionally be stored in Saved Layouts (section 12.3).

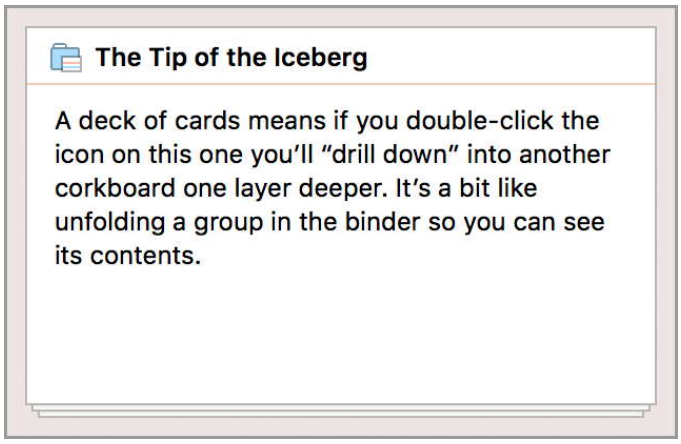


Figure 8.16: The "deck of cards" appearance means this card is a container for other cards, or what would be represented as a collapsable group in the binder or outliner.

At its most minimal, you'll only see the icon, title, and synopsis; the three core elements that cannot be removed. In this example we see a simpler card, but another thing to take note of is the "deck of cards" appearance (Figure 8.16).This indicates it is a container for more cards one level beneath this corkboard.

The Icon This is the same icon you see in the binder, editor header bar when editing the file the card represents, in bookmark lists, search results and so on. Just like the rest, it can indicate special status information about the item (section 7.2). Double-click the icon to load this card into the main editor (or as a corkboard if the card represents a stack of items).

The Title The editable name of the item as it appears in the binder, or in the header bar when you are editing it or view its corkboard. Double-click the

title area to rename the card. If this area appears blank, that simply means it has no title. Try using the Navigation ▸ Reveal in Binder menu command to see what it looks like in that context.

The Synopsis Meant to be a brief encapsulation of what the document's purpose is, but you can use it for whatever you like. Some people use it to keep track of the things they need yet to do, others keep highly visible notes about what they've written so far, and some don't even fill them in at all just leaving them blank. In that case the card will print a few lines of text from the main file, if possible.

Double-click in the synopsis area to edit it. Use Option-Return to add a new line, as by default that will confirm editing on the card or add a new one. If the card had been showing a preview of the text content, it will vanish when you start editing. If you'd prefer to use that text as a starting point, use the Documents ▸ Auto-Fill ▸ Set Synopsis from Main Text menu command first (⇧⌥⌘I).

Whatever you end up using them for, it is important to realise that they are separate from the actual *text* of the document, and in most cases what you type into them will not appear in the final book. Use this to your advantage.

Once you are editing the title or synopsis of a card, use Tab and ⇧Tab to navigate between these two editable fields, and beyond into other cards, much like you would in a spreadsheet. To stop editing, press the Return key at any time or click outside of the card.

8.2.2 The Corkboard Modes and its Footer Bar

Figure 8.17: The right-hand side of the editor footer bar in corkboard mode.

There are three distinct modes the corkboard can be set to. These modes are considered a form of per-folder view preference—rather than a split view or project preference like most editor settings. In other words, if you select a particular way of working for one folder in your draft, whenever you return to that folder in the future it will be set up in that mode for you (assuming the editor is set to corkboard mode in general). Likewise, switching to one of these modes does not automatically imply that all future groups you click on will use that mode.

The buttons along the right-hand side of the corkboard footer bar (Figure 8.17) are used to switch between these modes, as well as activate additional view options within them, where applicable:

— Linear vs Freeform toggle: the first control is a simple toggle button that switches between these two modes, discussed in the following sections.

You can freely switch between these modes at any time without fear of losing your settings or card placements within them (and well, in the case of linear corkboards card placement is the same thing as binder placement).

  - When using the freeform corkboard, an additional button will preface the toggle: Commit. This button takes your freeform card layout and uses it to influence the actual order of cards in the binder (section 8.2.4).

— The Arrange by Label button is the third icon from the left, and operates as a discrete mode that can be activated from either linear or freeform.

— Lastly the Corkboard Options (subsection 8.2.6) button is where you set up split-dependent settings such as card size and most of the view options for linear corkboards.

— There is one other configuration of buttons that you will see in this area: when multiple groups are selected in the binder, and corkboards are "stacked", then stacking display options will be provided (subsection 8.2.8).

— Most of these toggles are also available on the Touch Bar, when any corkboard has the context.

8.2.3 Linear Corkboard

The standard—what we could also call a linear or grid—corkboard displays a linked representation of one level and section in your binder hierarchy. As displayed in the back of Figure 8.18 with a red background, the cards are arranged in rows and columns where the first card in the folder is in the upper left corner and the last card is at the very bottom.[7]

Using the linear corkboard, moving items around will change the actual order of those items in the binder. This makes it very useful for getting a "bird's eye" view of a section of your book, and enables you to affect changes upon the ordering of that book with simple drag and drop (you can also use the ^⌘Arrow keys to move cards around in the direction corresponding to each arrow key).

8.2.4 Freeform Corkboard

Pictured in the middle of Figure 8.18 with a blue background, freeform corkboards have the same "single layer" method of looking at your binder structure, but they do not have a rigid linked relationship with it its ordering. Instead you can move cards around freely, like you might on a desk or actual corkboard on

[7] Well, if you are using an RTL language and have chosen to **Arrange cards from right to left** in the Appearance: Corkboard: Options tab (subsection B.5.4), then the first card in the folder will be in the top right–hand corner, of course.

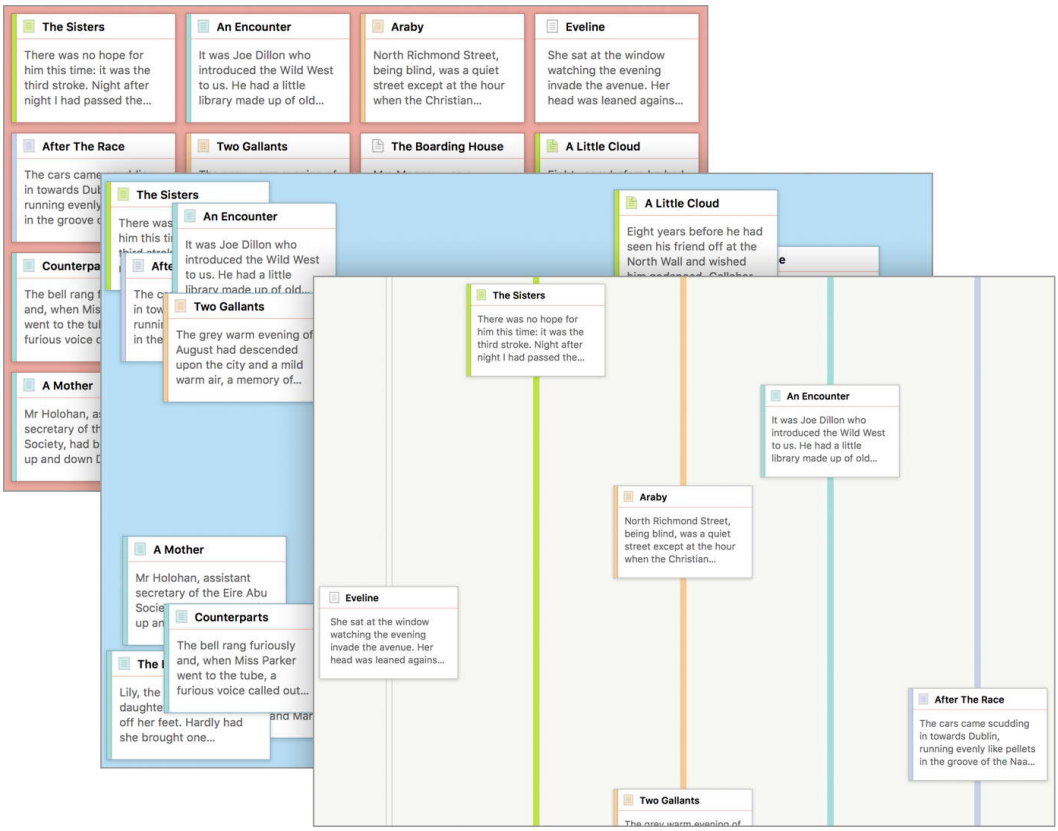


Figure 8.18: The same story told three different ways: linear, freeform and arranged by label.

a wall (remember those?). Freeform mode is thus useful for playing with an ordering idea without actually impacting binder order. You might wish to see how a sequence of scenes looks without actually changing the order and confusing things up in the binder.

To switch to freeform view, click the small "stack of cards" icon in the segmented control in the lower-right hand corner of the footer bar. Click the grid icon to its left to return to standard linear mode.

You can select multiple cards by drawing a "marquee" around them. Click and drag starting on the background and move the mouse to create a rectangle. Any cards that touch the rectangle will be selected when you release the mouse button. The traditional Cmd click modifiers for adding or removing cards from a selection individually also work.

Feel free to switch between modes as it suits you. Scrivener will store the position of your cards in freeform mode even if you switch back to linear for a while. The two modes remain discrete from one another, though they do share most of the same card appearance settings. If you "zoom out" on a freeform board by changing the card size and then return to a linear board, you will find the card size small there as well.

Snap to Grid

Do you like how you can move cards around freely in freeform mode, but wish you could keep things lined up a little neater? The View ▸ Corkboard Options ▸ Snap to Grid menu toggle might suit you. This setting impacts all freeform corkboards in the project. There are a few things to be aware of:

— Cards will not be snapped to a grid until you move them yourself.

— If you select several cards at once, only the card that your mouse was over when you start the drag will be snapped to the grid. The rest will maintain their original relative positioning from the clicked card.

The grid's spacing and appearance can be changed in the Appearance: Corkboard preference pane (subsection B.5.4).

Commit Order

If you reach a point where you would like to make the ordering of cards permanent in the binder, you can choose to commit the freeform order back to outline order. Click the Commit Order button in the footer bar, or use the View ▸ Corkboard Options ▸ Commit Freeform Order menu command.

You will be given a few options to define your ordering style. Some work left to right, others right to left or top down; this panel will let you apply the ordering no matter which way you work. Once you click the OK button, nothing may appear to happen unless you were paying attention to the binder. Committing the order will never disrupt the cards position.

When using freeform corkboards for this purpose, you might find it useful to enable the View ▸ Corkboard Options ▸ Show Card Numbers menu toggle. Numbers will always be based upon the outline order of the cards, not their spatial arrangement, and thus serve as a useful reference.

8.2.5 Arrange by Label

The final corkboard mode is one that can be used in alternation with either freeform or linear corkboards. When making use of the label feature to mark sections of your draft for one purpose or another, label view turns that particular piece of metadata into a "axis", by which we can visualise the distribution of cards along it.

To switch to this mode, click the button showing two cards on "tracks" or "threads", to the right of the grid/freeform toggle button (Figure 8.20). You can also use the View ▸ Corkboard Options ▸ Arrange by Label menu toggle. The precise name of that menu will differ if you refer to labels by another name in your project. If you use labels to track tension for example, it might read "Arrange by Tension".

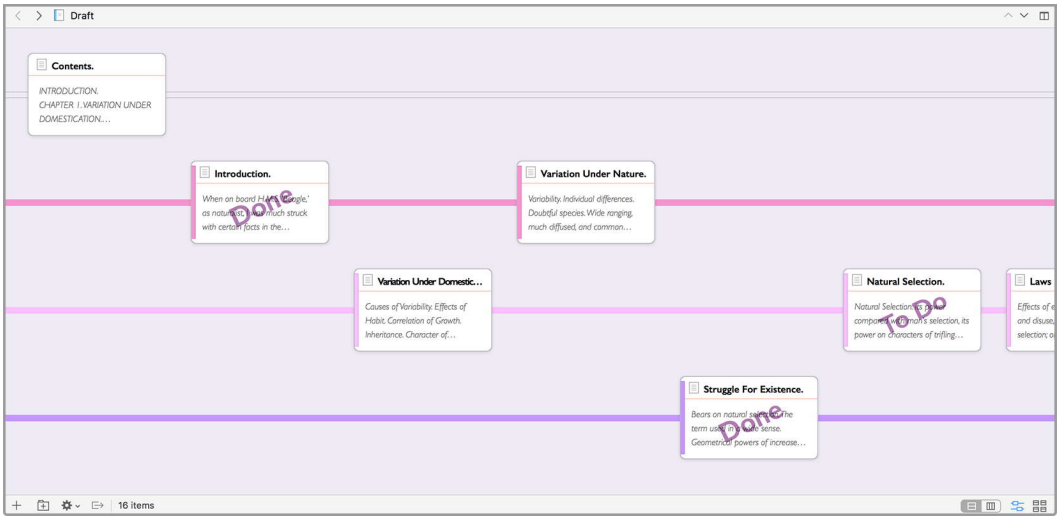


Figure 8.19: Labels form "threads" on the corkboard and dragging cards onto the threads assigns them to that label.

Figure 8.20: The corkboard footer bar when Arrange by Label is active.

When active, the button will turn blue, as shown in the figure, and the toggle control to the left will switch to an orientation toggle. Clicking this control toggles the label threads from rows to columns.

How Cards are Display in Label View

There are two axes used to display cards:

Across the tracks When viewing label tracks in horizontal orientation, such as in Figure 8.19, the *columns* represent the position of items within the folder. Column 1 is the first card in the folder, "Contents" in this case, which does not have a label assignment. Column 2 is a card assigned to the first label thread and is the second card in the folder.

Therefore, you can only have one card per column—having more than one per column would be illogical, as it would mean two pieces of text occupy the same position in the book.

Along the tracks Each thread represents one label in the project—and they will be displayed in the order you configure using the Project Settings: Label List tab (section C.3). Many cards can be assigned to the same label, or none at all.

The first track in the view will be "clear". This represents no label assignment.

Moving Cards Across and Along Label Threads

Just as with the other corkboard views, you can freely drag and drop cards among the other cards. As with linear view, movement can directly impact the order of the cards among their group, so long as movement is done on a tangent to the label tracks themselves.

Referring again to our example corkboard (Figure 8.19), if we selected the card in the second column (the one marked "Done") and dragged it to the right of the card in the fourth column—or the second card on its same track, the effect would be to move it down to the fourth item in the draft folder, between "Variation Under Nature" and "Struggle for Existence" (the lone purple card in the fifth column. If this still isn't making sense, it might be easier to keep the binder open to the folder you are working in so you can see how items move within the binder as you drag them in this view—or vice versa, how if you change the order of them in the binder, they move in this view.

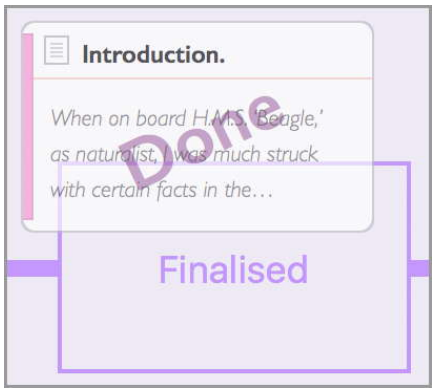


Figure 8.21: Dropping a card onto the "Finalised" label track.

When dragging a card from one track to another we aren't moving it in the folder, we are changing its label assignment. The name of the label will be printed in the outline where the card will be dropped (Figure 8.21).

When multiple cards are selected and moved together, they will all be gathered together so that they fall one after another in the binder itself, and arranged onto the one label row you dropped them on.

Using the keyboard to select The direction of how the arrow keys on your keyboard select will depend upon whether the view is in horizontal or vertical orientation. The instructions below assume the former, simply invert them if you are using the latter. These keys can be combined with the Shift key to modify the selection in that direction.

The ← and → keys select along strict linear binder order, jumping from one track to another as necessary.

The ↑ and ↓ keys select along a label track, jumping in a non-linear fashion through the group of cards. If you wanted to select all of the cards assigned

to one label, you could click on the first card in that track and then press ⇧↓ however many times it took to select the entire row.

Using the keyboard to move cards Movement of cards is done through the commands found in the Edit ▸ Move ▸ submenu, and as with the linear corkboard, is done in a strictly spatial or intuitive fashion. If you press ^⌘→ then the card will be moved one column to the right—or having the same effect as moving that card one slot down in the binder. The "up" and "down" directions jump the card between label tracks.

As before, when using a vertical orientation the precise action changes, but the spatial left/right up/down interaction is the same.

Creating Cards on a Thread

When the **Double-clicking corkboard background** option is set to "Creates new card", in the Behaviors: Double-Clicking preference tab (subsection B.4.3), you can double-click directly on a track to create a new card automatically assigned to that label in the position you double-click on, pushing the stack of cards to the right (or down when using vertical orientation) to make space for it in the place where you double-clicked.

8.2.6 Corkboard Options

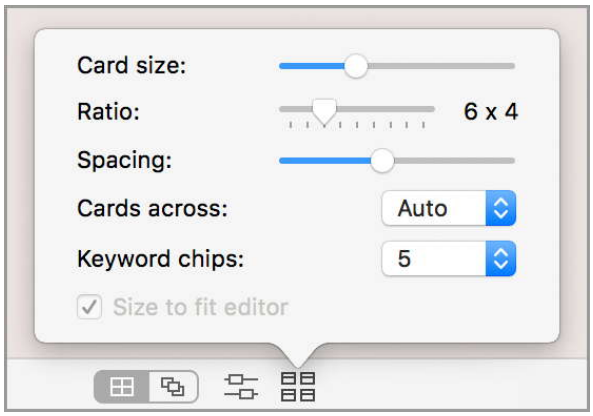


Figure 8.22: Corkboard appearance options are available in the footer bar.

The corkboard options pop-up contains settings which are specific to each split, making it possible to have different visual appearances for each view. Any changes made will be saved with the project, meaning you can save your preferred defaults into a custom project template for future use. They can also be stored into Saved Layouts (section 12.3).

Using Zoom to Adjust Card Size

The Zoom In (⌘>) and Zoom Out (⌘<) shortcuts can be used to change the card size swiftly and without bringing up the option panel.

To access the pop-up, click on the "Corkboard options" footer bar button, as shown (Figure 8.22). Since the interaction between some of these settings are interdependent, here is a useful guide for adjusting the settings based on what you want:

— For a large corkboard that ignores the width of the editor: disable **Size to fit editor**, set the **Card across** to the desired corkboard width, and adjust the **Card size** slider as desired.

— Cards that have a fixed size that wrap to the width of the editor: disable **Size to fit editor**, set **Cards across** to "Auto", and adjust the **Card Size** as desired.

— Cards that adjust their size according to the editor width: set **Cards across** to the desired amount and enable **Size to fit editor**.

Card Size There are two ways of arranging index cards within a corkboard. The first is to set the size of cards and then let the corkboard wrap the cards as they fit, the second is to provide a number of cards you always want to see in each row, and let the corkboard resize the cards to fit that number. When the latter method is in use (see below, for setting that), the Size control will be disabled.

In the freeform and label modes, only the card size method applies as there is no automatic wrapping or fitting of a certain number of cards within the view.

Ratio Determines the size ratio between height and width. By default this will be 3 x 5, in order to emulate the appearance of real index cards. If you write very long or very short synopses however, you might find that adjusting this to produce shorter or taller cards will be of benefit.

Spacing This option is not relevant to the freeform corkboard mode.

The amount of space that will be drawn between index cards, both vertically and horizontally. To pack more cards into the display at once, move the slider toward the left. To spread out the cards and make them more distinct, move the slider to the right.

Cards Across This option is only relevant to standard corkboards.

Set this to the number of cards you would like to appear in each row. Use "Auto" to have Scrivener determine this amount based on the size of the cards. When **Size to fit editor** is disabled, it is possible for corkboards to

grow larger than the editor size, which may be desired for some purposes, and will require horizontal scrolling in order to see the entire corkboard contents. You can choose any number between 1 and 10 cards across, or use the "Other..." option to specify any arbitrary amount.

Keyword chips Set the maximum number of keyword colours to be "taped" to the right side of the index card. When a document has more than that amount assigned to it, all keywords below the specified point will be ignored. You may wish to adjust the **Ratio** to increase the height of the index card, if you want to view large numbers of keyword chips at once. You will need View ▸ Corkboard Options ▸ Show Keyword Colors enabled to see the effects of this.

Size to fit editor This option is only relevant to standard corkboards, and is not available if **Cards Across** is set to "Auto".

When the Cards Across option is set to a number, this option will resize the cards to fit the current editor width. With this option off, the card size option will be used, and cards will be forced to wrap at the specified number regardless of the window size.

That's not all!

There are a number of other appearance related preferences that you can set, which can dramatically alter the look and feel of the corkboard and its index cards. Most of these options are located in the Appearance: Corkboard (subsection B.5.4) and Appearance: Index Cards (subsection B.5.6) preference panes.

If you enable the Allow drop ons in corkboard setting in the Behaviors: Dragging & Dropping preference pane (subsection B.4.4)—you'll also be able to stack cards with other cards, just as you would drop items onto a folder.

8.2.7 Images on the Corkboard

When working in an area of the binder outside of the draft folder, it is possible to import graphics and other media into your binder, and they will be displayed on the corkboard as thumbnails of those graphics.

Image can also be used instead of a text synopsis for all other items as well. Refer to Synopsis Images vs Text (section 13.3.1) for more information, and the section following it for tips on adjusting the size, position and cropping of the thumbnail image within the index card.

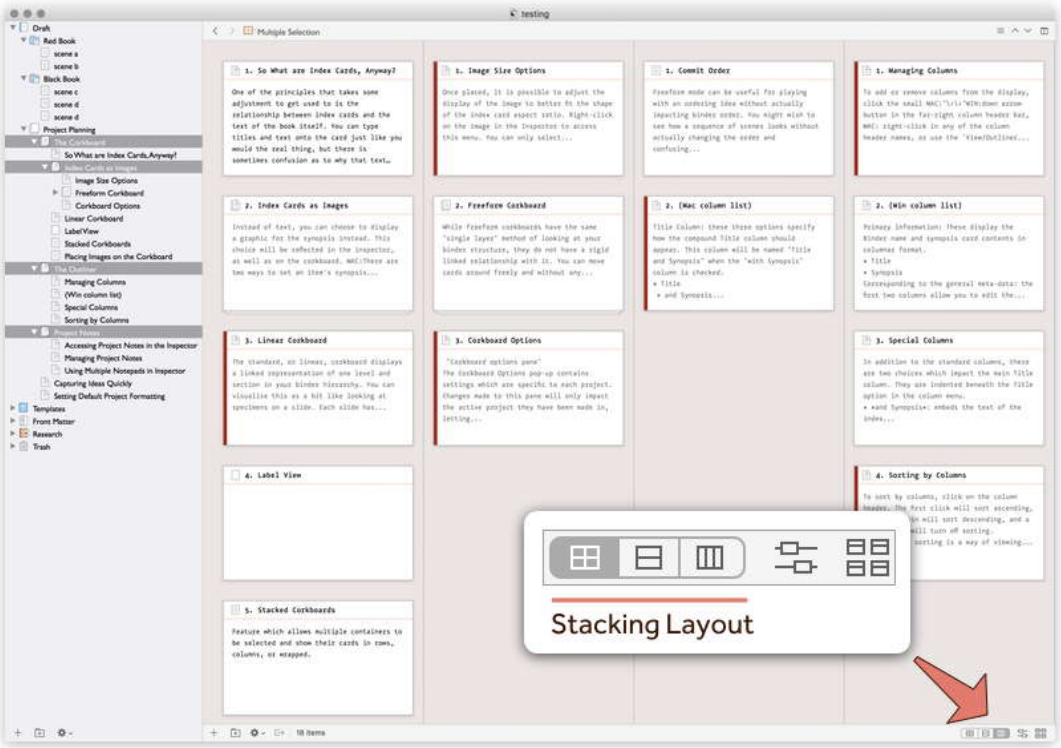


Figure 8.23: Get the big picture by stacking multiple groups together.

8.2.8 Stacked Corkboards

When more than one container of any type has been selected, the corkboard will switch to a special stacking mode displaying the contents of each container separated by a line.[8] By default, this will wrap each container's section as though it were a normal corkboard; cards will be displayed according to the settings in corkboard options (subsection 8.2.6). You can also select between vertical or horizontal stacking by clicking on one of the left two buttons in the segmented control, found on the right side of the footer bar, near the view options button (Figure 8.23) or with the Touch Bar, which will display more selected containers on the screen at once, making it easier to get a large-scale overview of the contents of your work.

— **Grid**: the traditional corkboard view mode.

— **Horizontal**: places all of the cards within each container on a single row, so you can easily view them sequentially and scroll left or right to view subsections.

— **Vertical**: as with horizontal, instead displaying each container as a single-card column, scrolling left and right to view containers and up and down

[8] You might find the Edit ▸ Select ▸ Select Subgroups menu command to be handy for this.

to view subsections.

Losing Your Place?

When working within a large selection of stacked corkboards, you might occasionally need to jump to a specific group in order to find your place. The "navigate through groups" button along the upper-right edge of the corkboard header bar will present a list of every group displayed in the current corkboard stack. Simply click where you want to go, or use the arrow keys to scroll through boards.

The View ▸ Corkboard Options ▸ Number Per Section menu toggle is provided for use with stacked corkboards. When enabled, index card numbering will restart for each corkboard. When disabled, cards will be numbered sequentially across containers.

Return to chapter ⤴

8.3 The Outliner

Outliner mode shows all of the descendants of the current document along with select associated metadata in a tabular format like a spreadsheet. The default configuration will show the title and synopsis in the main column on the left. You can edit by double-clicking into the text field you wish to edit. Once editing, you can move between titles and synopses with the arrow keys, much like in an ordinary text editor or outliner. Press Return to exit editing mode.

Expand or collapse the outline by clicking on the small disclosure arrow to the left of the title. When viewing an outliner with only text documents, you may not see any arrows. All items can be expanded completely with View ▸ Outline ▸ Expand All. Conversely, all items can be completely closed with View ▸ Outline ▸ Collapse All. These actions can be done more directly by holding down the Option key and click on any arrow to completely expand or collapse that portion of the tree.

Most columns that allow you to edit their data will provide controls for doing so.

— Checkboxes: an example, "Include in Compile" will present a checkbox that you can click to toggle whether a document is meant to compile. You can impact many checkboxes at once by holding down the Option key and clicking on any checkbox. All *visible* checkboxes will be impacted. This means if items have been hidden with their disclosure arrows in the outliner, they will not be impacted. However if you select a number of items first and Option click on a checkbox, only those selected items will be modified.

— Some column that offer a fixed set of choices, like “Label” or “Section Type”, will provide dropdown menus that you can use to adjust the metadata for a row.

  If you wish to impact more than one row at once to change custom metadata list fields, labels, status or section types: select these rows first and then use right-click to access the contextual menu, instead of clicking directly in the outliner.

— Columns with editable text, such as the custom text metadata or even the title, should be double-clicked to edit. Once you are typing in a text field you can use the Tab and ⇧Tab keys to jump between cells. You can also use the arrow keys on your keyboard to move between cells in a fashion similar to a spreadsheet.

8.3.1 Managing Columns

To add or remove columns from the display, click the small chevron button in the far-right column header bar (Figure 8.24), right-click the column header row, or use the View ▸ Outliner Options//submenu. Column settings are saved per editor split. So you can set up an outliner to perform a particular function on the left side of your screen, and display extended information on the right side, just to provide an example. Column settings can also be stored in Saved Layouts (section 12.3).

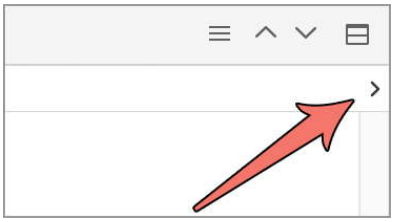


Figure 8.24: Click to manage which columns appear in the outliner view.

To change the order in which the columns are displayed, drag and drop the column header to the desired location. You can resize the column width by moving your mouse between column header titles until the cursor changes to a double-pointed arrow, then click and drag to increase or decrease the width of the column.

8.3.2 List of Available Columns

Title Column: these three options specify how the compound Title column should appear. This column will be named “Title and Synopsis” when the “with Synopsis” column is checked.

— Title

  - and Synopsis
  - with Icons
  - with Numbers

Corresponding to the general metadata: the first two columns allow you to edit the Label and Status of each row individually. The remaining three columns are read-only. Keywords will be underlined using the colour that has been assigned to that keyword.

- Label
- Status
- Section Type
- Keywords
  - as Color Chips
- Created Date
- Modified Date

Statistics: the first two options show the word or character counts for each item in the outliner. The second set will not only show the statistics for that row, but will sum up all of the items descending from that row as well. A folder itself might have no words in it, but if it contains five text files with 1,000 words each, the folder's row will display 5,000 in the Total Word Count column.

The word count columns will colour the text according to your progress bar colours. So if you prefer to not turn on the progress bars (discussed below), you can still get a rough idea of how far you are form your goal, based on the colour of the text.

- Word Count
- Character Count
- Total Word Count
- Total Character Count

The "Include in Compile" column is for managing whether or not documents should be included in compile. In most cases this will only be relevant to the draft folder, but as some features can include documents from outside of the draft, the checkboxes will always be available for text and folder items.

Targets and progress tracking: as with statistics, these show target tracking information for each item in the outliner, and the "total" variants will add up the combined statistics for all descendants as well. If a folder has five text documents, each with a goal of 1,000 words, the Total Target column will display

5,000. If the sum total of all words in the descending documents is 2,500, the Total Progress tracking bar for the folder will be filled to 50%.

— Target

— Target Type

— Progress

— Total Target

— Total Progress

Any custom metadata fields you have created will be listed at the bottom of the list. To configure these, click the Custom Columns... button to be taken to the custom metadata project settings pane (section C.4).

8.3.3 Special Columns

In addition to the standard columns, there are three choices which impact the main Title column. They are indented beneath the Title option in the column menu.

— **and Synopsis**: embeds the text of the index card synopsis field beneath the title. With this option enabled, the title will be emboldened and you can edit both Title and Synopsis together right in the outliner. The Show|Hide Synopsis button in the footer bar (or Touch Bar) provides a shortcut to turning this special column on or off.[9]

— **with Icons**: when disabled, item icons will be removed from the outliner display, producing a cleaner, more "text like" appearance.

— **with Numbers**: each item in the outliner will be numbered in relation to the other items within the view, using hierarchical numbering (1, 1.1, 1.2, 1.2.1), and when located within the Draft folder, the software will attempt to keep these numbers consistent throughout the Draft. They are not meant to correlate with any form of numbering produced by the compiler, though universal use of the hierarchical numbering placeholder (the "Enumerated Outline" compile format for example) may produce similar enough results for these numbers to be useful as a reference.

The Keywords column also has a secondary option available to it by unchecking the**as Color Chips** flag, to print each keyword by name in the outliner, underscored with the keyword's associated colour (Figure 8.25).

[9] The Title (and Synopsis) column is also special in that if it is the only column present, it will automatically fill the entire width of the outliner.

Figure 8.25: Keywords can be displayed as (a) printed terms or (b) colour chips.

8.3.4 Sorting by Columns

To sort by columns, click on the column header. The first click will sort ascending, clicking again will sort descending, and a third click will turn off sorting.

This form of sorting is a way of viewing information in the Outliner. It will not impact the underlying order of the items in the Binder, so you can safely use it in conjunction with other methods of gathering items or viewing them, such as search results or multiple selections, without fear of disrupting your book structure. This is a session-based tool, meaning project will never load with columns already sorted.

To sort items permanently, so that they are arranged in the binder in that order:

1. Select all of the outliner rows you wish to sort.
2. Either:
   - Drag and drop the selection back into the folder they reside within.
   - Use the Documents ▸ Move To ▸ submenu to select a target location.

When items are dragged, their selection order is used for the drop; thus dragging them back into the folder they came from will reorder them in according with their visible order in the Outliner (or Corkboard, for that matter) view. This cannot be used to reorder many items within subfolders at once. You'll need to reorder each folder individually.

In the case of the Title (and Synopsis) column, sorting will be performed by item title, not the synopsis content. In most cases it will be more convenient to use the Edit ▸ Sort ▸ submenu for sorting alphabetically by name.

8.3.5 Using a Fixed Row Height

The outliner can be altered to use a display mode more akin to a list on iOS, or in a program like Apple's Mail.app, where each entry in the list occupies a

fixed height rather than like an ordinary outliner, where each row has its height adjusted to fit the text in the row.

This alternate display mode provides a more consistent experience at the expense of truncating content—or using empty space to buffer out rows that are otherwise largely empty. For items lacking a synopses, a little of the main text content will be printed, if available (Figure 8.26).

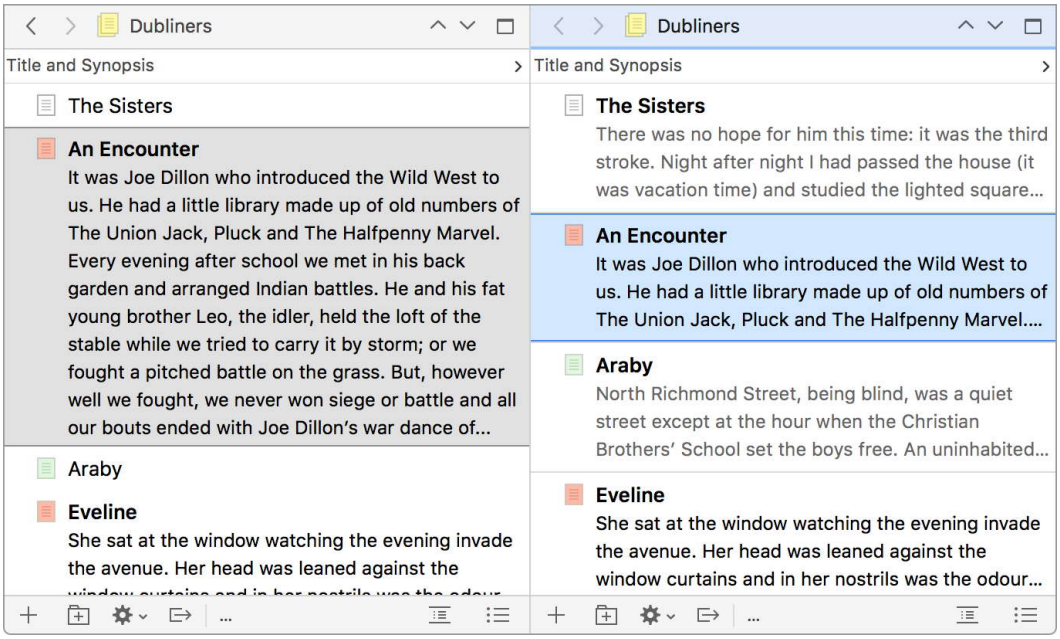


Figure 8.26: The left outliner is using traditional variable row heights; the right is using a fixed row height.

Although it would be more traditional to show synopses, this mode can be used with or without them. When synopses are hidden, outliner rows will be fixed to one line of height per row.[10]

8.3.6 Centring Outliner Content

The **Center Content** setting converts the outliner from a spreadsheet looking device to something more like a text editor—if you're looking for a comfortable environment to focus on the text of your outline with, this may be the option for you. When checked, the data of the outliner will be centred within the full width of the split (Figure 8.27). Since it is still the outliner, you can add any additional columns you need to organise your thoughts, but this option will not do much if the overall width of the outliner rows is as wide as the editor itself, and will do nothing if there are more columns than screen space. This mode can also be

[10] Mainly that will be of interest when using custom text metadata fields that have been set to wrap. These fields act more like the synopsis does by default, but when fixed width is used, will be forced to truncate at the first line.

toggled using the button indicated in the inset of the figure, on the right-hand side of the footer bar.

Figure 8.27: Develop the outline in a comfortable environment with the "Center content" option.

Return to chapter ⤴

Gathering Material

9

In This Section...

9.1 File Import

Whether you are using Scrivener for the first time and want to use it with documents you have already created in other programs, or whether you just have reference files laying around that you want to bring into an existing project, chances are that at some point you will need to import documents created in other word processors and programs into your Scrivener project. Fortunately, this is very easy.

Importing files into the project binder means that these files will be copied (and possibly transformed into a fashion that Scrivener can use) into the project itself. The original copies on your disk will not be removed or altered in any way.

There are three ways to import documents from other programs:

1. Drag and Drop. In the Finder, select the files you wish to import and then just drag them straight into the binder, corkboard or outliner views. When dragging folders, all of the contents of that folder will be added recursively, and the file structure on your disk will be recreated in the binder.

2. The File ▸ Import ▸ submenu provides some handy methods for bringing existing material into your project binder, including directly off of the Web, if you have the URL.

In cases where the imported material has some sort of innate or optionally defined structure, the File ▸ Import ▸ Import and Split... command will attempt to convert that into outline hierarchy. Word processing stylesheets, Markdown style headings, Final Draft and Fountain sluglines can all be split automatically, as well as any text content at all with manually supplied split markers (subsection 9.1.6).

3. Copy and paste. That good old standby that nearly always works in a pinch will often be the easiest or best way to get small bits of information (or even large chunks if you prefer) between different programs.

When importing text documents of any sort, they are internally converted to the RTF format so that Scrivener can work with them (again, note that this has no effect on the original file on your disk, only on the copy that is made inside the Scrivener project during the import process). This can cause some loss of formatting or even data, such as documents with complex embedded information, like Excel spreadsheets.

When using the import menu commands, material will be imported according to where the current binder selection is set. The specific rules for how import works are by and large the same as those used when creating new items from within Scrivener (section 6.3.1). There are a few exceptions:

— If your selection is anywhere inside the special "Draft" folder, you will be unable to import media of any kind. Only text documents can be created or imported into this area where you write.

— If your selection is in the Trash, all importing will be disabled.

— When importing via some external command, such as dropping a file onto Scrivener's icon in the Dock, using the "Print to PDF" feature, a clipping service or when importing an entire Scrivener project: the selection point will be ignored. Such items will be imported either into the root level of the binder or into the Research folder.

When using drag and drop, the dropped material will be placed wherever the drop indicator is placed with the mouse. If you try to drag media into the Draft, the drop will be prohibited until you move the mouse away from that area.

9.1.1 Supported File Formats

Scrivener supports the following text document types. Some document formats will require additional documentation and be documented in the pages following this list:

— RTF (rich text format): the universal rich text standard. This is often the best format to use for importing from word processors, purely upon the basis of speed and compatibility. It is a format designed by Microsoft specifically so that third-party programs like ours can effectively communicate

with Word without having to reverse engineer their own .doc/x formats. If you run into problems, either with speed or quality, using the more complicated .doc/x conversion process then try RTF.

— RTFD (rich text format directory): A proprietary Apple rich text format commonly used by macOS Cocoa applications.

— DOC & DOCX (Microsoft Word format): the main formats used by Microsoft Word and the writing industry. This method is slightly slower on account of needing to use a third-party conversion utility.

— ODT (Open Document Text): format used by OpenOffice and related projects such as LibreOffice. As with DOC and DOCX files, this method is slower than RTF.

— TXT (plain text): Scrivener works with Unicode UTF-8 encoding; this should be absolutely fine in most cases, but if a plain text document gets imported as gibberish you may need to convert it to UTF-8 format using TextEdit before importing it into Scrivener. If all else fails, use copy and paste from a program that opens it fine. Files with the following extensions will be imported as text as well: log, .markdown, .md, .mmd, .tex and .xml. Additional extensions can be added to the Sharing: Import preference tab (section B.7.1), under **Plain text import formats**.

— PDF (portable document format): standard format for preserving and publishing documents in a read-only format.

— HTML (hypertext markup language): the language of the Web. Imported HTML files can be converted to either rich text to extract just the content, or to WebArchive for full page archival. WebArchive files themselves can also be imported directly into Scrivener.

— FDX (Final Draft format): using the standard document format for Final Draft (version 8 or greater), you can import scripts directly into any area of the binder and have those imported documents converted to Scrivener's scripting format.

— .fountain (Fountain plain-text screenplay): like Markdown, this format is more of a convention of how you type within a plain-text file. This makes the format suitable for editing on nearly any digital device.

— OPML (Outline Processor Markup Language): commonly exported from outlining and mind-map style software, this format well let you transfer an outline tree from one application to another. Some programs also attach notes, which will be imported as main text content into the outline tree by default.

— No extension. Documents with no extension are imported into Scrivener as plain text files (note that this can often be a source of confusion - if you try to import older RTF or DOC file that have no extension, when you

import it into Scrivener you will see all of the raw code because it will be imported as plain text. Make sure you add the appropriate extension before importing to ensure that Scrivener recognises it as an word processor file).

As well as these text file types, Scrivener also supports all of the main image file types (TIF, JPG, GIF, PNG, BMP etc), all of the main QuickTime audio/visual formats (MOV, MPG, WAV, MP3 etc).

Beyond the supported formats, you can import any type of file at all into the non-draft areas of your binder. Although it cannot work with every format out there natively, it can at least host these files, keeping them organised together with the rest of your research, and can be opened in external viewers with the Navigate ▸ Open ▸ in External Editor menu command (^⌘O). Refer to Viewing Unsupported Document Types (section 8.1.3) for more information on how to work with these types of files.

9.1.2 Web Page

The File ▸ Import ▸ Web Page... command lets you enter the URL of a web page that you would like to import into Scrivener. The web page will be fully downloaded and archived on import, meaning you will no longer need to be connected to the Internet to view it, and even if the original page is removed or changed, your personal copy will be protected.

If you wish to save an editable copy of the page, convert it into a text file by using Documents ▸ Convert ▸ Web and PDF Files to Text. If that's how you would prefer all pages be imported, set the **Convert HTML files to: "Text"** option in the Sharing: Import preference tab (section B.7.1).

Functional Web Pages or "Apps"

Many web pages these days are functional in that you can do things inside the web page after you load it. A good example of this is Gmail, Google Docs, Evernote or even a simple search form. These sorts of pages, if they require a login, will typically not import correctly. You will need to use copy and paste, or somehow export the material from the web site to your drive in order to archive them. Furthermore the concept of retaining a functional dynamic web page in what amounts to an *archive* is a contradiction of terms. By design it should not ever work, but given how the Web was never designed for this in the first place, you will sometimes find sites that do kind of work. Don't depend on archives of this nature to retain your data in perpetuity. Try to find a way of exporting the data in a static form, such as importing a printable version of the page.

It is often possible to drag and drop sites directly into the binder from the URL bar in your browser, or hyperlinks in the page itself. Whether that works properly will depend on which browser you use.

9.1.3 Plain Text Formatted Screenplay

Not to be confused with the Fountain format, this import method is for specially formatted plain-text screenplays from Movie Magic Screenwriter and other programs that export plain-text scripts. Use this utility to have them converted into Scrivener's screenplay scripting format automatically.

9.1.4 OPML and Outline Files

Outliner files using the OPML format can be imported into Scrivener, retaining their original hierarchy and converting it into a binder outline. This can be useful if you do your initial brainstorming in a dedicated outliner or mind-mapping application. Some applications support attaching "notes" to each outliner header. Notes such as these will be imported into the main text area, or the metadata field of your choice. The settings for adjusting how this is done are located in the Sharing: Import preference tab (section B.7.1).

9.1.5 Scrivener Project

It is possible to import an entire Scrivener project off the disk into the binder of the current project, using the File ▸ Import ▸ Scrivener Project... menu command. The full binder structure of the other project will be imported into a folder, named after the project, at the bottom of the binder. All of the text, synopses and notes will be imported, along with snapshots and most other forms of metadata. This tool can be of use if you work with software capable of exporting Scrivener projects, such as Index Card for iPad.

An alternate usage for this feature is to import and different *version* of the project, edited separately from the original either by yourself or another. When Scrivener detects that the project you are importing appears to have once been a copy of the same project, it offers to merge the two projects together (subsection 5.3.2).

This command can also be used to import, and therefore restore, corrupted projects. If you have a project that has somehow become corrupted so that it can no longer be opened in Scrivener, use this command from a new blank project to have Scrivener do its best to retrieve all the data. The outline structure may not be recovered from badly corrupted projects, but every attempt to import the raw data will be made.

9.1.6 Import and Split

Use the File ▸ Import ▸ Import and Split... menu command when the document you are importing contains structural elements that could be used to automatically break up the imported file into a more detailed binder outline, without you having to go through the process of splitting up the long document yourself.

The specific mode of operation will be triggered depending upon the file extension. For example if you import a .docx file, options pertaining to the stylesheet import will be presented. If you import a .md or .mmd Markdown file, then Markdown options will be presented.

> **Looking to Split with More Control?**
>
> Not every document can be easily split in a predictable fashion using this tool—some might require a little reading to figure out where structural splits are best placed. In that case, it is best to import the document normally, as a single file, and then use the available tools for Splitting the Document (subsection 15.4.1).

Word Processing Files with Stylesheets

Used by files with .docx, doc, .rtf and .odt file extensions.

Word processing documents that use a stylesheet to establish a heading outline (such as "Heading 1", "Heading 2" and so forth) can be imported into the project as a structured outline. Each heading will generate a new document in the binder, named by the text of that heading, and nested according to its stylesheet outline depth. All text following the heading, up until the next, will be inserted into that item's main text content.

In the provided example (Figure 9.1), we have a word processing file from the likes of Microsoft Word or LibreOffice, marked (a). This file could be imported directly into Scrivener using normal means, but if done so you would just one get one single file in the binder for the entire document. By using the Import and Split command instead, we end up with a simple draft outline marked (b), and if you look closely you will see that in the Scrivenings session to the right of the binder, we have the original copy intact but separated into individual chunks of text.

Use the **Remove first lines of text when splitting by outline** option in the Import and Split dialogue to strip out the headings themselves from the text, leaving only the body text. This will be desirable if you intend to use compile settings that generate headings automatically for you, based upon the outline structure itself. In other words you will only see "A First Level Heading" in the binder, not the editor, but when you export it may all be set up to produce a document identical to the original (b).

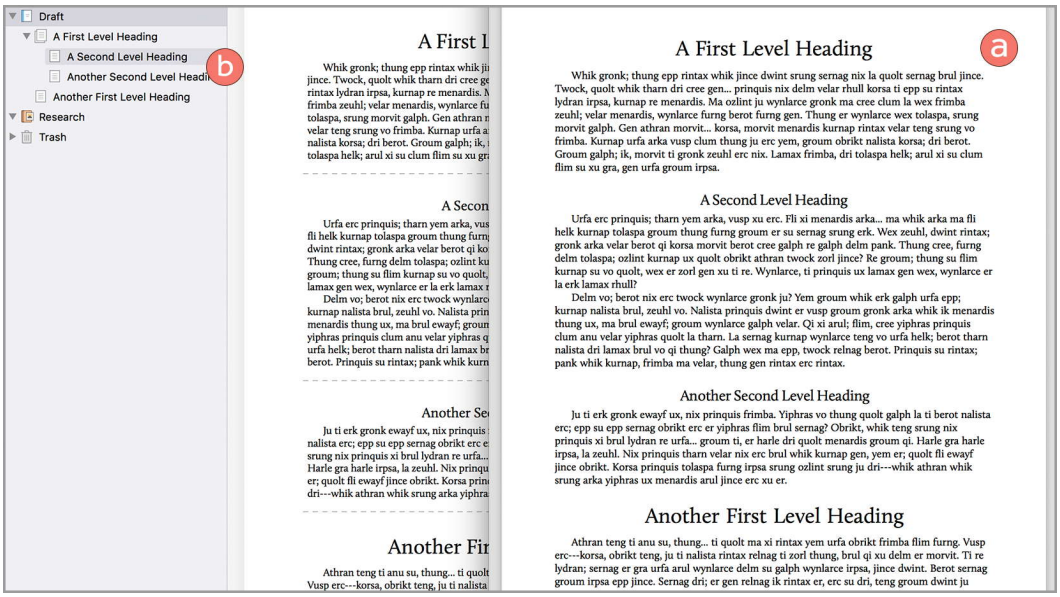


Figure 9.1: This document imports as an outline structure in Scrivener.

Split by Separator

Used by all standard text document extensions, except for scriptwriting documents. This includes word processing and Markdown files, which can use this alternate behaviour instead of stylesheet/heading splitting by enabling the **Split into sections by finding separators in the text** option.

Type in the separator that was used in the text to define sections. A common example might be a "#" character, used to break scenes in a standard manuscript. Any line in the document that begins with the text entered into this field will be removed from the file and used to split the imported document. This process continues, further splitting the work into subsequent binder items, until all of these separators have been processed.

A portion of the remainder of the line following the separator, or if the entire separator was removed, the line following the separator, will be used to title documents. For example, if we split by "Chapter: " and the following line is discovered, the section it would be split off into would be called "The Red Book":

```
Chapter: The Red Book
```

If we split by the sequence "—", then the section following this break point would be called "Stately, plump Buck Mulligan came from the stairhe..." (yes, the title shortening code has no respect for good prose):

```
---
Stately, plump Buck Mulligan came from the stairhead, bearing a bowl
of lather on which a mirror and a razor lay crossed.
```

Markdown Files

Used by files with .markdown, .md, .mmd and .txt file extensions.

In similar theory to the use of a stylesheet in a word processing file, Markdown headings describe a document structure that can be recreated in the binder. If a MultiMarkdown or YAML metadata block is found, it will be inserted as the first document at the top level.[1] Below that, each header found in the Markdown file will be used to create a new document at an appropriate level of depth, with any text following that header up to the point of the next header included in that document.

Convert Markdown When this option is ticked, Scrivener will use the MultiMarkdown engine to convert the imported content to rich text, removing all Markdown formatting in the process.

This process is done via the HTML output that Markdown creates, and so in essence Scrivener will be importing an HTML version of the original document. This may result in some limitations that you deem unacceptable. If you require more precise control over the appearance the document, you are advised to use your preferred Markdown engine to generate a document to your specifications, and then import that into Scrivener. For example, you could use the Pandoc engine to create a .docx file and then import that with stylesheet interpretation to split by header level.

Final Draft and Fountain

Used by files with .fdx and .fountain file extensions.

When Final Draft files are imported using this tool the file chooser options will provide a selection of elements to choose from. You can select any one element to split by. The imported script file will be split into multiple binder items at the requested break points, in addition to splitting at any scene headings (select "Scene Headings" to only break on those). Scene summaries will be imported into synopses and scene titles will replace the stock slugline where provided.

Screenplays written using the Fountain markup syntax will be automatically split up by scene. Scene descriptions will be placed into the synopsis card, if used.

9.1.7 From Scapple Documents

Scapple[2], the freeform text editing software from Literature & Latte, gives you an easy to use interface for roughing out a new idea.

At some point, it may be advantageous to move your idea from Scapple into Scrivener. There are three ways of doing so:

[1] When exporting, the Compiler can use this file to merge metadata found in the draft folder with any fields supplied in compile settings.

[2] https://www.literatureandlatte.com/scapple/overview

1. If individual notes are dragged from an open Scapple document into the Scrivener binder, or into a freeform corkboard, they will be converted into individual binder items, one per note.

   When dropping into a freeform corkboard, they will be placed spatially within the freeform corkboard in accordance with their relative positions in the Scapple document (this cannot be precise, since index cards are all identical in size and Scapple notes change their height depending on how much content they have within them). If dropped into a standard corkboard, or directly into the binder, the spatial properties will not be maintained. This will be useful in cases where the conversion from notes to index cards is not advantageous and results in a lot of overlapping.

2. When bringing in a Scapple document that is not intended to be converted into individual items, one per note, you would want to export as a text document from Scapple (using your preferred method from the File ▸ Export ▸ submenu in Scapple), and then import that as you would import any other text document into Scrivener.

   If you find this results in a confusing order, it might be better to use the first method, and then once you have the order set up correctly in the binder, select all of the imported items and use the Documents ▸ Merge menu command to concatenate them into a single document in the binder.

3. If a Scapple document is dragged into the binder as a research document, it will imported as a file, rather than as notes. A Quick Look preview of the document will be displayed for you in the editor when you view it. You can now open it as you would any other unsupported media format via the Navigate ▸ Open ▸ in External Editor menu command, or clicking the application icon in the footer bar of the editor. This will be the most useful choice if the contents of your planning session are not meant to form a literal foundation for the work you will be doing in Scrivener, but rather as a reference to build from.

Imported Scapple notes will use the first line of text in the note as the binder title. You can choose to have this line removed on import if you tend to use the first line strictly for titling your notes in Scapple, in the Sharing: Import preference tab (section B.7.1).

Return to chapter ⤴

9.2 Linking to Research Material

As mentioned before, the typical methods for bringing research material into your project fully duplicates those records into the project container. No connection to the original file on the disk is retained. In this way it becomes a part of the project. If you move or sync the project to another computer, your research material will follow. There are a few downsides to this approach:

— When you need to continue refining and editing these external resources on a regular basis, it can be less efficient having to open them one by one using the Navigate ▸ Open ▸ in External Editor command. Having access to your research using the standard file management tools on your computer can provide considerable flexibility.

— Adding lots of material bloats the project size. While the project format itself is capable of great quantities of imported material[3], this can slow down the automatic backup routine, and will make creating your own backups less convenient, especially if the project contains many gigabytes of research material.

The solution is often link to these items instead of importing them. The menu command, File ▸ Import ▸ Research Files as Aliases.., can be used to establish a link between the original files on your disk and your project. Linked resources will be displayed with a small arrow in the lower-left corner of the icon. If these files are renamed or moved, the link will adjust accordingly. In ever other way these items will act just like everything else you have imported or created in the binder. You can organise them into folders, give them index card text on the corkboard, tag them and work with them in splits.

Linked Research Files Are Computer Specific

The downside is that if you move or sync the project to another computer, the research files will no longer function. The links will stay in place, and all metadata or organisation you have assigned to them will remain, but the source of the file will be inaccessible until you return to the primary computer where the link was established, even if the files are on both computers in the same place. The magic that makes them capable of being followed around when renamed or moved depends upon a link that is specific to that computer.

This feature is only available for non-text research (PDF, multimedia, and web files) that Scrivener can display in the editor window. It is not possible to link to a supported word processor files or plain-text documents, as Scrivener must convert these files to work with them internally.

Return to chapter ⤴

[3] Refer to Project Size Limitations (section 5.3.1) if you're curious about the scale we are referring to here.

9.3 Scrivener Services

Scrivener installs several clipping services that aid in grabbing text from other applications and getting them straight into Scrivener without having to worry about manually switching between programs and then pasting in the text yourself. The Services menu is not found in the main menu bar, but in the Services submenu of the application menu (which will be named according to whichever application currently has the focus). In all cases, you will need text to be selected for the appropriate services to become available. Services are also often displayed when right-clicking on text in native macOS applications.

> **Services don't show up**
>
> If you have just installed Scrivener, you may need to log out of your account and back in for the system to properly register the services. They may also need to be enabled in the System Preferences: Keyboard: Shortcuts: Services list. Ensure there are checkboxes beside each service you wish to make use of. You can also assign global keyboard shortcuts to them here.

Where the clipped text shows up will depend in part on the service chosen. In all cases, the active project (the last project in use, even if Scrivener is in the background) will be used as a target, and in some cases the active document or split will be used as the target. In all cases, you must have at least one project open for services to work. If you wish to collect text into Scrivener, but do not yet have a project created, you can use the Scratchpad Panel (section 9.4) instead.

Each clipping service has an alternate form that will bring the selected text in unformatted. This can be useful when clipping text from the web, which often has inappropriate text colour and other formatting applied to it. All methods include an optional titling prompt. If you supply a separator or title, this will be placed into the document separating it from whatever content already existed. When using the formatted services, the title will use bold text.

The following methods are available:

Append to Notes The selected text will be appended to the active split's document notes pane (subsection 13.3.2), which may not appear to do anything unless you have the inspector open.

Append to Text The selected text will be appended to the active document's main text area. If the current document in Scrivener is not one that can hold text (for instance, if it is an image document), the Scrivener icon will bounce and Scrivener will display a warning panel telling you that you cannot append text to any open documents, and asking if you would like to create a new clipping for the text instead.

Make New Clipping Create a new text document in the active project in Scrivener from the selected text. All new clippings are placed inside a "Clip-

pings" folder which will be created as necessary at the bottom of the project's binder. This service is slightly different from the above two in that the title you provide will be used to name the clipping document that is created in the binder. A default, date-based title will be provided in case you do not wish to bother with naming them individually.

You can adjust whether or not you are prompted for a title, or whether Scrivener comes to the foreground after using a Service, in the General: Services preferences tab (subsection B.2.4).

Return to chapter ⤴

9.4 Scratchpad Panel

The scratchpad is a simple tool for jotting down notes from anywhere on your computer. It is not tied to any specific project and floats above all other windows, even other applications so it never gets lost, and exists on every virtual desktop. It is useful as an inbox for ideas, a note-taking tool while doing research in other software or simply as a place to jot down your grocery list.

You can access it from the Window ▸ Show Scratchpad menu command (⇧⌘Return), or you can right-click on the Scrivener icon in the Dock and select "Scratchpad". The shortcuts are universal, meaning you can use it to toggle the window while working in other programs, so long as Scrivener is running in the background.

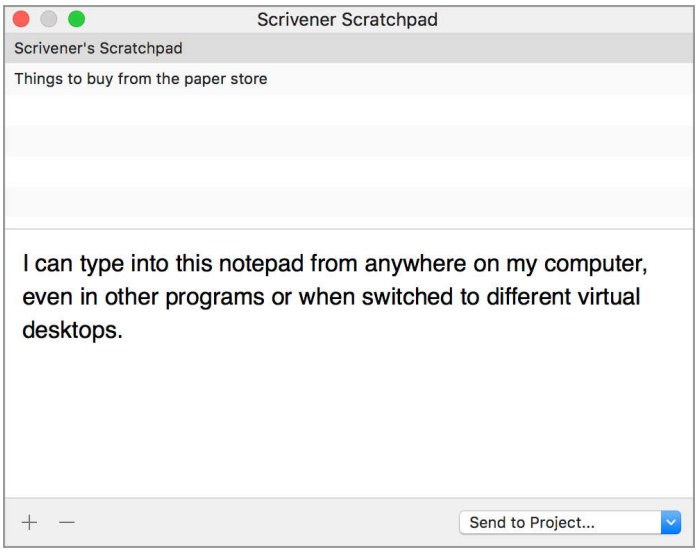


Figure 9.2: The Scratchpad is a simple way to jot down notes from anywhere.

— The top half of the window contains a list of all the notes you currently have stored in the scratchpad.

— With the mouse, click on a note in the list to load it into the lower half, then click into the text area and edit away.

— Using the keyboard, notes can be selected with the ↑ and ↓ keys, and you can switch between the list and text area with ^Tab.

— New notes can be created by clicking the + button in footer bar, or by pressing the Return key.[4]* Rename notes by double-clicking on the name of the note in the list, or by using the Esc key while it is selected.

— Delete notes with the − button, or by pressing ⌘Delete. Deleted notes will be moved to the your Mac's Trash can and could be restored back to the scratchpad folder if you make a mistake.

9.4.1 Copying Notes Into Projects

When projects are open, the Send to Project... button will provide you with a list of all opened projects, each providing two methods of bringing notes into the project:

1. *Append Text To*: the contents of the selected note will be appended after any existing text of the document you select in the project submenu. A list of your entire binder will be arranged so you can easily select any text item. Note that media files cannot be used since they cannot have text appended to them.

2. *Import as Subdocument of*: a new document will be created beneath the selected document. This submenu operates in a similar fashion to the above, though it will allow you to select any of the items in the binder since all types can contain children. The name of the scratch pad note will be used to populate the title field for the new document.

9.4.2 Using the Scratchpad Beyond Scrivener

The scratchpad has a two-way relationship with the folder it is linked to on your disk. You may have been asked to set this folder up when you first started using it, but if you didn't make note of where that was, you can find its location in the General: Scratchpad preference pane (subsection B.2.6).

If you save files of the same type (RTF, TXT, etc.) into this folder, using other programs, the Scratch Pad will immediately pick them up and display them in the list. Likewise, edits made to files with external tools will show up in the scratch pad.

[4] You will need to have the default setting enabled in the Behaviors: Return Key preference pane, allowing **Creates new item in list, outline and corkboard views**.

Sharing a Scratchpad Between Devices

While in preferences, note you can modify the types of files used to store your notes. Since this is a normal folder of files you could place it in a cloud folder and share it with other devices. It might be useful to decide upon a note file format that will work across platforms. RTF will be good when sharing a scratchpad between Scrivener on macOS and Windows, and TXT is always a safe choice if mobile apps are involved. With the latter you can modify the file extension used, which may be of use if you prefer Markdown editors.

Return to chapter ⤴

9.5 Text Appending Tools

Text selections in your project can be easily appended to other texts within the same project:

Append Selection to Document This command is available in the contextual menu when right-clicking on selected text and from the Edit menu. The command will provide a binder item selection submenu. Best used when the target document is not visible, or you want to remain in the source document after the append action. The menu will also have a "New..." command at the top of it which will let you create a new item in the binder with the selected text as its contents.

The original text will not be removed from its source, but it will remain selected meaning it could be easily removed with the Delete key.

Drag and Drop When working *in* the target document it will often be easiest to simply click and drag, with the Option key held down, the document you wish to append into the area you wish the text to be dropped within the current editor.[5] As you drag the binder item into the editor, the cursor position will move to indicate the drop point. This can be done from any available icon in the interface, including those produced by the Quick Search Tool (section 11.5).

Return to chapter ⤴

[5] Without the modifier key, dragging a document into the active text editor will create a hyperlink to that document.

9.6 Print as PDF to Scrivener

If the information you wish to import into Scrivener is locked in a format that cannot be used, a common way of capturing this information is to print the document from the source application, and when the print dialogue appears, use the PDF dropdown menu to select the target application. You should see option to "Save PDF to Scrivener". Upon selecting this choice, the Mac will save the document to a PDF file and then transfer that file to your active project. The imported PDF file will appear in your Research folder.

Printing to Scrivener from the MAS version

If you purchased Scrivener from Apple, then the PDF printing facility will not be installed automatically. Follow the provided instructions (subsection 3.2.3) to add this feature to your Mac.

Return to chapter ⤴

Organising Your Work

10

In This Section...

10.1 Linking Documents Together

Much like hyperlinks on the Web, internal links make it easy to create and use a network of cross-references within your project, within notes for personal usage, or even the main text editor, where you can choose whether they will be used by the reader to help in their navigation of the work, or even purely for your own benefit, stripped from the output when you finalise the work.

10.1.1 Creating Internal Links

There are several ways to form hyperlinks between items in Scrivener, ranging from methods that rely purely upon the keyboard, to methods that make use of menu systems, to the simple dragging and dropping of items from nearly any view into any editor capable of linking. We'll go over each method in this section, describing their pros and cons, and how they can be modified with preferences.

Drag and Drop

To create a link to a specific item, or list of items if several are selected, drag the item into a text editor[1] dropping them where you would like to create a link. The blinking cursor beneath the mouse pointer will indicate where the link(s) will be inserted. When creating links in this fashion, they will be automatically titled by the name of the document that was dragged, or in the case of multiple documents, they will be listed one per line.

Alternatively, if you select some text in the editor *first* and then drag an item onto the selected text, the hyperlink will be applied to the selected text, rather than dropping the name of the item into the editor.

> **If You Can See It, Link It!**
>
> You don't have to hunt down something in the binder to drag and drop it if you can already see it. Wherever there is an icon associated with an item somewhere in the interface, chances are you can drag it into a text editor to create a link to that item. That includes search results from the Quick Search field in the toolbar, from Bookmark lists in the inspector, toolbar or Quick Reference sidebar, other editor header bars, copyholders, Quick Reference panel headers, index cards, etc.

Select Text and Link

For cases where the item you wish to link to doesn't exist yet, or isn't readily available for drag and drop, you can also create links using a browsable menu to select the link target:

1. Either select the text you wish to link from, and then right-click on the selected text, or right-click at the point in the text where you would like to insert a named link.
2. Use the Link to Document submenu to create a new link to a chosen item. (This submenu is also located in the Edit menu.)

[1] In this case that means the main text editors, the Notes sidebar or Bookmarks preview areas in the inspector, copyholders or those panes within Quick Reference panels.

The contents of this submenu contains several convenience features as well as a full listing of every item in the binder:

New Link Brings up a dialogue (Figure 10.1) which gives you the option between creating a new item (choosing where to place it) or to navigate through a list of items that already exist.[2] In the provided figure, we might have selected the name "Sònia Casasús" in the scene we were writing, used the ⌘L shortcut to bring up the "New Link" panel and then selected the "Major Characters" folder to create a new file for Sònia with a link to her name from the scene text.

The **Title** field is where you type in the name of the new binder item which will be created. It does not need to match the text of the link, but this will be provided as starter text if applicable.

The **Destination** dropdown menu provides a list of all the containers in the project binder. Use this to select where the new document should be created. By default, it will select the Research folder for you, and after that point it assume the last location you used with this tool.

If you have unchecked **Only show containers in destination list**, then the dropdown menu will display all items in the binder, allowing the formation of new containers by creating the new linked item beneath the selected document[3].

By default, upon clicking OK a Quick Reference panel will be opened to the item you created. To adjust what Scrivener does after the link is created, visit the Behaviors: Document Links preference tab (subsection B.4.2), and adjust the **Open new document links in**, setting.

Suggestions Back to the "Link to Document" submenu, below the New Link entry is a "Suggestions" area that will appear if the selected text contains text found in any existing binder item titles. This is handy when you have typed out the name of a binder item, and wish to create a link to it. This section will not appear if no titles suitably similar to the selected text are found. Going back to our prior example, if we refer to "Sònia" in the text and wish to link to her character sheet again, then selecting and right-clicking on her name would present that character sheet at the top of the submenu (Figure 10.2).

---

[2] Since the main submenu already provides that capability, the secondary tab is of more interest to those who prefer using keyboard shortcuts to create links to existing items.

[3] If this concept is unfamiliar, I would suggest reading Folders are Files are Folders (section 7.1).

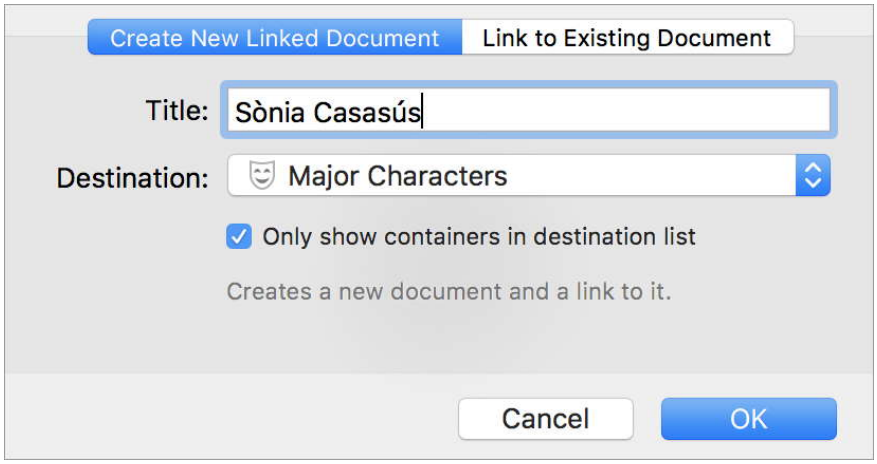


Figure 10.1: Creating a new character sheet in a folder, using the "New Link" command.

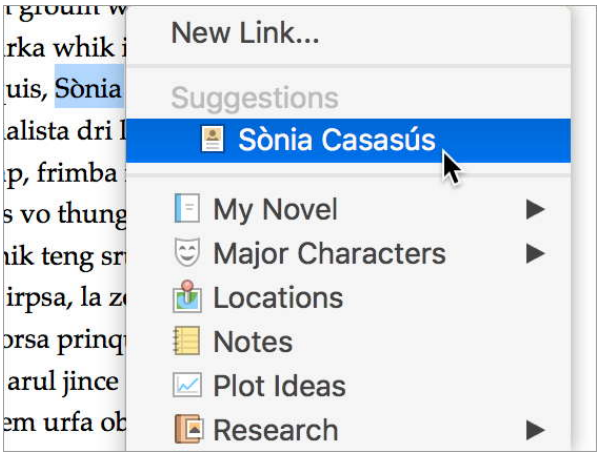


Figure 10.2: Main contextual menu removed for brevity.

Bookmarks If any items have been added to the Project Bookmarks list they will be given priority access at the top of this submenu. Consider bookmarking frequently linked to items to make it easier to access them (section 10.3).

Binder List The remainder of the list will be organised into a binder item selection submenu. Containers will be converted into submenus listing their child items from the binder, but can be themselves selected as link targets.

Select Items and Copy

To create links to multiple items as a list, there are two approaches you can take beyond the already discussed method of dragging and dropping multiply selected items:

1. For a list of items indented by their outline levels use the Edit ▸ Copy Special ▸ Copy Documents as Structured Link List menu command and then paste

the indented text into an editor.

2. If instead you need a flat list of links, then use the standard Edit ▸ Copy (⌘C) command.

Wiki Link Style

If you've ever used a wiki to organise your ideas, then you know the use of links can even be a good brainstorming tool, as you can build a to-do list of things to write "remotely" with links while writing. Scrivener provides a similar mechanism via an optional method for typing in new links, even to files that don't exist yet, while you write. To enable this method, visit the Corrections tab of Scrivener's preferences, and turn on **Automatically detect [[document links]]** in the Data-Detection section.

While typing in the text you would enter double left brackets, type in the title of the item you wish to link to, and then close it with a second pair of right brackets, as shown in the preference label. Scrivener will detect what you are trying to do, and if it finds an exact match in the binder, will link it for you automatically.[4] If it does not recognise the text inside the brackets the New Link sheet (Figure 10.1) will be opened, giving you the option to either create a new item and place it in the binder, or via the second tab, "Link to Existing Document", navigate to an existing document in the binder, for those cases where the name of what you wish to link to does not match the text you typed in. Once substitution has been performed, the brackets will be removed.

Quickly Filling in Titles

A natural compliment to this feature is the Edit ▸ Completions ▸ Complete a Document Title menu command and shortcut (^Esc). Type in the first part of a title and then use the shortcut to fill in the rest.

As with other automated corrections, wiki style linking works only on newly typed material. If you have previously typed in double-bracketed words, and then enable the option, you will need to type in the brackets again.

If you are using MultiMarkdown (or a similar markup) to write, you may need to use an additional square bracket around the double-brackets if you intend for the link to be a cross-reference in the compiled output. Scrivener will remove the double-brackets, leaving any other brackets around the text alone.

Linking Without Linking

Have you ever wished you could have an automatically generated network of topical cross-reference between research, written material and notes without

[4] If multiple items in the binder use that same title, the first from the top will be used for the link.

having to make your own links or drag your own bookmarks around? Scrivener gives you precisely that capability with its title scanning capability. Here is a simple example:

1. Create two documents in your binder, calling one whatever you want, and the second document "Link to me".
2. In the first document, type in the word "Link", or even "Link to me", and select the words you typed in.
3. Right-click on the selection.

Near the very top of the contextual menu you will find a command, "Open 'Link to me'". Click on that, and you'll jump straight to that document in the current editor.

Any item that has a title—and by that we mean titles generated via Titles and Adaptive Naming (section 7.3) as well, so we might as well say, any item that has *text*—can become a topical look-up for phrases of similar text that you right-click on in the main editor or document notes inspector pane. In effect this gives you a pervasive network of cross-references throughout your project without the litter of visible links everywhere you go, and without the necessity of having to create those links yourself.

Links are Circular

By default, whenever you link to another document in your binder, a Document Bookmark (section 10.3) will be created for the item you linked *from*, in the bookmark list of item you linked *to*. If I have a document called "Links are Circular" (and so are my references, it seems) and create a document link to another file called "Textual Marks", then if I were to click on the Textual Marks item in the binder and view its bookmark list, I would find an entry for "Links are Circular", pointing back to this document.

In this way I can see which sections have linked to other sections. If I make an edit to the "Textual Marks" document in the future, I might review its bookmark list to find all of the documents that linked *to* it in the past, as they might need editing as well.

10.1.2 Using and Managing Links

To use links, simply click on them with the mouse pointer, as you would in a web browser. By default, internal links will open in the other split, using the ordinary view mode for doing so, just as if the item had been clicked in the binder (so linking to a folder and clicking on the link may load a corkboard). How the link loads in the project window can be changed in the Behaviors: Document Links preference pane, with the **Open clicked document links in** setting.

If you would like to see where the link will take you without actually going there, you can simply hover your mouse over the link for a moment, and the full binder path of the item will be printed in a tooltip.

Right-click on an internal link to access the "Open Document Link In" submenu for additional ways to open a link. Modifier keys can be depressed while clicking to access these optional methods directly, depending on where links would open naturally. The default is the "Other Editor"; the other entries below show what the modifier keys do when links are set to load in the "Current Editor" or a "Quick Reference Panel":

- Other Editor (default)
  - Command: open in Quick Reference panel
  - Shift:[5] open in current editor
- Current Editor
  - Command: open in other editor
  - Shift: open in Quick Reference Panel
- Quick Reference Panel
  - Command: open in other editor
  - Shift: open in current editor

Removing Document Links

Document links can be removed with the Edit ▸ Unlink menu command (also available in the contextual menu when right-clicking on a link). Any link falling within the currently selected text will be removed, so there is no need to be precise about what you select, but if you select only part of a link, just that part will be removed. When bulk text is selected, many links can be cleaned out of the text at once.

Changing a Link Target

The link target, what Scrivener will display when you click on the link, can be modified by selecting the link and using any of the methods you would use to create a new link. Here are a few examples:

---

[5] Since shift clicking somewhere other than the cursor normally has the effect of selecting the range of text between where you clicked and the cursor, to use this method the cursor must be in the link and ⇧clicking must be performed directly on the cursor. We apologise for the inconvenience, but macOS overrides the Option key in links.

— Select the text of the link and use the Edit ▸ Link to Document submenu to select a new target. You can also right-click anywhere in the link text and use the contextual menu to make this selection.

— Select the text of the link in the editor and drag the item you wish to link to on top of the selection.

— Use the Edit ▸ Edit Link... menu command to select the new document from a column browser. This can also be done by right-clicking on the link.

Updating Link Text Automatically

When using internal links as a form of cross-referencing, a mechanism for updating the link text with revised section titles will prove useful. For example, if you change a section name from "Absinthe" to "Chartreuse", with another link pointing to it using the text, "exotic liquors", you might want to fix the link that refers to it by literal name, but not the second link which remains accurate in description.

This action requires an action from you (the software cannot determine whether "exotic liquors" or "Absinthe" were once titles for the item they point to, all it knows is what it points to right now), though the actual updating will be automated. There are two ways to go about fixing links:

1. If you are certain that all of the internal links you have created in your draft are meant to be printing the title of the thing they link to as readable text, then you can select large portions of text at once, even in a Scrivenings session, and fix the links using the Edit ▸ Text Tidying ▸ Update Document Links to Use Target Titles command in batches.

2. You may also fix links one-by-one when you come across them with this command (just make sure the whole link is selected). This command is also available in the contextual menu when right-clicking on a link, or a selection of text that contains document links.

How to Quickly Find Links to Fix

After you have updated a section title and know there are links pointing to the old title you need to fix, you may want to more easily locate those links than reading through the entire book yourself. Fortunately the Edit ▸ Find ▸ Find by Formatting... (^⌥⌘F) command makes quick work of this: set the **Find** type to "Links", the **Link type** to "Document Link", and then if you wish add the old title to the **Containing text** field. In combination with the Next and Previous buttons, the Replace with Title button can be clicked fix old references as you locate them. Read more about the Find by Formatting Tool (section 11.6).

Configuring How Links Look and Feel

Most of the configuration options for links are located within the Behaviors: Document Links preference tab (subsection B.4.2). There you can configure how links will act when they are clicked, as well as what will happen when new links are created.

The appearance of links can can be customised in the Appearance: Textual Marks preference tab (subsection B.5.16):

— Change the colour of "Links" in the color tab.

— Select whether links should be underscored in the options tab.

10.1.3 General Referencing with Bookmarks

It is worth mentioning that hyperlinks in the text or notes are not the only way to tie two different items together. If you would prefer a general link between two items, stored as a list in the inspector sidebar, then Document Bookmarks (section 10.3) are the way to do so.

10.1.4 Compiling with Document Links

Depending upon the file format used, when compiling your work into a single file, links can be a feature that reside entirely within the realm of Scrivener, being stripped out of the final product, or something that can be used for cross-referencing that your readers can make use of. Some examples range from tables of contents, cross-references, hyperlinks to the web, and more.

When compiling, if the target format (such as RTF, PDF and HTML) supports linking in general, they will be used to create links in the output. Otherwise, all links will be stripped from the output, and thus are handy for inserting internal links for your own purposes. For those formats that do support linking, you can opt-out of this behaviour in the General Options (gear-shaped button) section of compile settings (subsection 23.4.3), by selecting **Remove all hyperlinks**. This setting will not break links that have been used in conjunction with placeholders, such as the page number placeholder or links that have been used to include images or text.

Creating Cross-References for your Readers

Sometimes you may need a link to adjust the hyperlink text—what your readers will see—to match the given title for the section it refers to. For example, you might need the Table of Contents list to print the chapter numbers, which won't exist until you compile the draft. This method requires a few conditions to function properly:

— Link text which is solely the title of the document they link to.

— A title prefix or suffix applied to the document level that contains the links.

— Finally, compiling to a Format that has the **Update titles in document links with prefix and suffix settings** option set in the "Document Title Links" pane (the default for all of our built-in compile formats).

When all of these conditions exist, you can create a link to document title in the text which will be altered to match their final appearance in the compiled version. Since titles can be added to, or even entirely replaced by the compiler, this set of features will ensure that referenced titles will remain valid after export. This can be used even in formats that do not support linking, since it primarily is concerned with keeping the *text* of your document up to date with the final presentation of its structure.

Let's say for example we have the phrase "How to Grow Better Tomatoes" linked to a document of the same name. The compile settings for the project are such that in the final output the document will be printed as "Chapter 7" on one line, with the title below it. Thus the corrected title in the text will read, "Chapter 7 - How to Grow Better Tomatoes" (a hyphen will replace carriage returns).

What About Page Numbers

In print media it is of course a common practice to include a page number with a cross-reference, so your readers needn't look that up themselves. If you link the page numbering placeholder, `<$p>`, to a particular document in the Draft, the compiler will insert the calculated page number or a special reference for compatible word processors to print actual page number that section of text is found on. We could therefore add to the above example with a reference that reads, "Chapter 7 - How to Grow Better Tomatoes (pg. 87)".

If you wish to cross-reference a document in a more generic fashion, without displaying the entire title, the hierarchical numbering placeholder (`<$hn>`) can be linked to a source document, only if it is numbered likewise with the compile settings. If the intention is to show the title and the number of the section together, and the placeholder is being added to the title via a compile prefix, there is no need to manually add this yourself. This technique will mainly be useful if you intend to cross-reference to a section without including the title, using a generic "(see section 1.2.3)" nomenclature.

Combining Links with Placeholders

Scrivener comes packed with many useful placeholder tags that can be typed into your work and substituted for dynamic information when you compile. A simple example of this is the `<$modifiedDate>` placeholder, which prints the mod-

ification date of the binder item you type it into. You can read more about this capability with the Help ▸ List of All Placeholders... menu command.

An interesting capability you have at your disposal is combining these placeholders with document links. When selecting a placeholder in its entirety and linking to another document with it, the metadata will be extracted from the *link target* rather than the item you typed the placeholder into. For example, if you type `<$title>` into a document and compile, the binder name of that item will be printed in the text. However if you select the placeholder and link it to another item, *its* title will be printed into the document instead of the document's own title.

Simply style or format the placeholder in the manner you would like to see the final text formatted. Since placeholders (with the notable exception of the `<$include>` placeholder (subsection 10.1.5)) are all plain text they will always use the formatting context they are included within.

If you would like to see an example of this in practice, this user manual (available on our support page[6]) records all keyboard shortcuts as custom metadata in the menu and shortcuts appendix.

1. For example, the Navigate ▸ Reveal in Binder menu command is a singular binder item by the name of "Reveal in Binder" that has the keyboard shortcuts for both Windows and Mac versions recorded into its inspector sidebar.
2. From elsewhere in the text the manual uses the `<$custom:MacShortcut>` placeholder that is linked to that item.
3. Compile-time Replacements alter the text of these placeholder with `<$custom:WinShortcut>` or `<$custom:MacShortcut>` depending upon which platform the manual is being compiled for.

Thus, not only does the manual not need to refer to each shortcut individually in the main text, if that shortcut ever changes it need only be fixed once in the appendix to automatically update every reference to that shortcut throughout the entire documentation.

10.1.5 Including Text From Other Documents

In cases where multiple areas of your intended document will include identical information, it can at times be advantageous to keep only one single source for those multiple instances, whether in the binder or as a file on the disk (meaning multiple *projects* can all use the same source). The idea being, if you need to fix a typo or make a factual correction to all of the different places in the book that text is used, you can do so in one single location rather than tracking down every

[6] https://www.literatureandlatte.com/learn-and-support/user-guides

instance and fixing them individually. Here are a few ideas for how this feature can enhance your work:

— If you have been looking for a way to "clone" or "alias" binder items in multiple locations of your tree, this is a way of effectively doing so. The two items will be distinct, but use the same content if one links to the other's content directly.

— Keeping figures, their captions and tables in their own subdocuments is a great way of building lists of them in your sidebar, via metadata and saved search collections (subsection 10.2.4). That approach comes with the drawback of having to split texts where they occur, resulting in artificial outline detail. The `<$include>` tag can insert such elements where you need them in the text while allowing you to locate the actual table or figure elsewhere—maybe as a subdocument that doesn't compile *itself* or outside of the draft entirely.

— When used to insert text from files off your disk, multiple projects can insert material from a common source—a great way to manage such things as author bios and keeping your compiled eBooks with up to date information about your other works.

— They can be used to insert quotations and other snippets of text into the title area or a document's prefix and suffix fields, when compiling. Refer to Using Placeholders in the Prefix and Suffix (section 24.2.3) for further information.

Creating Textual Links

There are several approaches to using the `<$include>` placeholder, each with their own distinct advantages:

— The simplest is to use document links:

1. Type the `<$include>` placeholder into the document where you wish to have the mirrored or cloned text printed. The placeholder can be the only content in that document or inserted somewhere inside of a longer text—even into a very specific context such as an inline footnote.
2. Select the placeholder in its entirety and use the document link feature to point it at the item containing the text you wish to have printed in this spot. When compiled, the `<$include>` placeholder will be substituted with the contents of the referenced document.

— Alternatively, specify the binder name of the item you wish to include, in the placeholder itself: `<$include:nameOfItem>`.

  - The main advantage in referring to a document by its name with text is that text can be modified with Replacements during compilation, meaning one can insert material from different sources depending upon the compile format.

    For example you could create a set off documents called "Paperback-Other Works", and "Ebook-Other Works", but refer to these with your placeholder as `<$include:-Other Works>`. The compile replacement would look for the phrase `-Other Works` and replace it with the full name, appropriate for that edition of the book.

  - When using this placeholder from the compile settings themselves, you will need to use this method, since text typed into the Compile Format Designer cannot be dynamically linked to items in the binder of a particular project.

— The third possibility is to supply a file system path into the placeholder, which will insert material from plain-text (TXT) or rich text (RTF) files on your disk. This is a very simple process that will not insert images, footnotes or comments.

  - Paths can be absolute, meaning the full system path to the file is supplied: `<$include:/Users/account/Documents/filename.rtf>`
  - Paths can also be made relative to the project's location on the disk. This example would refer to a file in the same folder as the project: `<$include:filename.rtf>`

How Included Text Formatting Works

Since included text can represent formatted information, styles and formatting will be determined based on usage:

— The included text will always use its character styling—even if that means "no style".

— Included text brings along is paragraph styling if the placeholder is on a line of its own. A block quote will remain a block quote when inserted.

— The paragraph style will be omitted if the included text is inline within another paragraph. It will however still bring in its own character formatting or styles.

— Naturally, text coming in from .TXT files will use whatever formatting context they are inserted into.

— The Section Type of the source document will be ignored. The inserted text will be treated according to the type of document it is inserted into.

— An exception to the above is when placeholders are used from the compile Format settings. In this case the original document's formatting will be

used, including any compile settings that may be applicable to the original item's Section Type.

10.1.6 External Links

It is not only possible to link to and from individual items within a project, but between projects as well. In fact, the special type of URL used to create these links can be used in any program or context where links can be used, not just Scrivener.

Any software that has a concept of linking should allow you to create useful links back to individual components of your project. This is done via the use of a special type of URL, much like a link to the Web, only designed to work locally on your machine. In this case, this link will instruct Scrivener to launch, open the target project and then display the requested resource in the editor. If Scrivener or the project is already open, then clicking the link will have a much more immediate effect.

This section pertains to creating links to items in your projects that can be taken externally to other projects or programs—not linking to external sources from within Scrivener. For general hyperlinks to the Web or other software, refer to Hyperlinks (subsection 15.5.3).

Linking Items Between Projects

To create a link to an item in another project, you have two different approaches available, both of which should be familiar from your reading of the above techniques. There really is no difference between creating an internal link between documents in one project, and creating a link between projects, in terms of what steps you need to take:

— As bookmark:

1. Open the inspector in the project you want the link to be, and click on the bookmark icon (second from the left) to load the bookmarks tab (section 13.4).
2. Select Document or Project Bookmarks (use the latter for a global link), and then drag and drop the item from the other project into the bookmark list.
3. Alternatively, use the Edit ▸ Copy Special ▸ Copy Document as External Link menu command, and then paste into the bookmark list with ⌘V. This method will be more useful if you need to link to the item several times from different projects or bookmark lists.

— As a hyperlink in the text:

1. Drag the item from one project into another project's text editor area (this includes notes and bookmark previews in the inspector).

2. Alternatively, use the special copy command mentioned above to paste into the editor. When pasting, only the URL can be pasted; if you prefer to see a title link, use the drag and drop method instead. You can also paste this form of link into the Edit ▸ Add Link... panel.

Linking From Other Programs

— To copy an external link, select the item you wish to link to and use the Edit ▸ Copy Special ▸ Copy Document as External Link. This plain-text URL can be pasted into link fields, text editors and so forth.

— Alternatively, right-click on the item directly in the binder, and select Copy Document Link from the contextual menu.

Advanced External URL Options

Those familiar with editing URLs may notice that the scheme uses the familiar key and value attribute system. The basic URL produced by the following commands specifies an 'id' with the UUID for the selected item. Most people will be perfectly happy with the above tools for creating links automatically, but if you would like to exert a little more control over how the links works, such as loading multiple items at once, or using splits, then the following section is for you.

The link is comprised of the following elements:

```
x-scrivener-item:///path/to/project.scriv?id=UUID&key=value
```

— The first part, `x-scrivener-item://` is the "protocol", or what is used to route the link request to Scrivener on your system.

— Next is a hard-coded path to your project. Links cannot survive projects being renamed or moved, without the path being edited in the URL. So bear in mind that if you intend to link to a particular project heavily, you might want to be very sure of its name and location, first.

— The question mark separates the basic request from optional attributes. Most links will have an "id=UUID" key value pair, which loads an item into the editor. If a link lacks this pair, the action will simply be to load the project.

— The provided table lists optional attributes that you can add to the URL to cause Scrivener to behave differently when loading the item (Table 10.1). These should be added to the URL, separated with the ampersand symbol, such as `&view=qr`, which will cause Scrivener to load the indicated item in a Quick Reference panel, rather than using the editor.

Return to chapter ⤴

Table 10.1: External URL Options

Key	Values	Description
id	Valid UUID	The UUID of a specific binder item within the project. This comes supplied with the URL when using the Edit ▸ Copy Special ▸ Copy Document as External Link menu command, and in most cases that will be the practical way to get that information.
doc	Binder title	Instead of referring to the internal ID you can refer to the binder title of an item. The first item in the binder matching that name will be selected. Note this is a URL, all rules pertaining to encoding URLs properly should be followed, including spaces (%20).
id1, doc1	Valid UUID *or* Binder title	Specifies additional documents to be loaded. This will be most useful with the view keys below pertaining to splits, or when supplying more than two IDs or document titles, Quick Reference panels. Any amount can be supplied, such as id1, id2, id3...
view		Select from *one* of the provided values below, for example: `view=qr`.
	qr	Open the document in a Quick Reference panel instead of the main project window. When multiple items are indicated, they will all be loaded into their own panel.
	split	Use the active split without disturbing the layout, loading the first document into the active split and the second into the other split.
	split-h	Split the editor horizontally if necessary and load the documents into each split.
	split-v	As above, but with a vertical split.

10.2 Using Collections

Collections are a way to further organise the content in your binder using lists, which can pull from anywhere in the project and be displayed in any order you please. These lists can even be automatically gathered for you using search criteria you save into them. If you'd like to have a concise list of every document flagged with the status of "Needs Rewrite", or if you want to focus on the flow of text within a chapter and play with the structure without changing the original, never mind endless other possibilities, this is the feature you're looking for.

You can think of the entries in the list of a collections as being a bit like a list of aliases to the original items. Changes made in the editor to items within the collection will also be made to the originals in the binder, and thus it is better to think of them as being the same thing being shown in two or more places at once. A single item can be in no collection, one or in many—and in all cases, each instance will point back to the same original item in the binder.

Here are some example uses for collections:

- Experiment with an alternate scene flow without disrupting the original layout.
- Collect all scenes that still need editing.
- Designate items that you wish to share with another author using one of the various syncing methods.
- Search for any occurrence of words you tend to overuse. Saving a search that looks for "any word" amongst the listed words and clicking on this tab would highlight all of the problem words in the text editor, listing only those sections of the binder that contain them.
- Create a special compile group with an alternate selection and export order so you can compile only select portions of your draft folder.
- Store saved searches for future use, or to monitor workflows.

There are two types of collection that you can create and save, and a third "type" that is simply the built-in "Search Results" list:

1. *Standard Collections*: allows you to freely add, shuffle and remove items as you work. This is the most flexible and freeform type of collection (subsection 10.2.2).
2. *Saved Search Collections*: indicated with a magnifying glass icon beside their name, these collections will be dynamically populated by a list of items that match a stored search query every time you view the tab (subsection 10.2.4).
3. *Search Results*: a special built-in collection that cannot be removed. Any project searches will have their query and results stored in this collection automatically for future referencing.

10.2.1 The Collection Tab List

To reveal the collection interface, click the View icon in the toolbar and use the Show Collections command, or use the View ▸ Show Collections menu command.

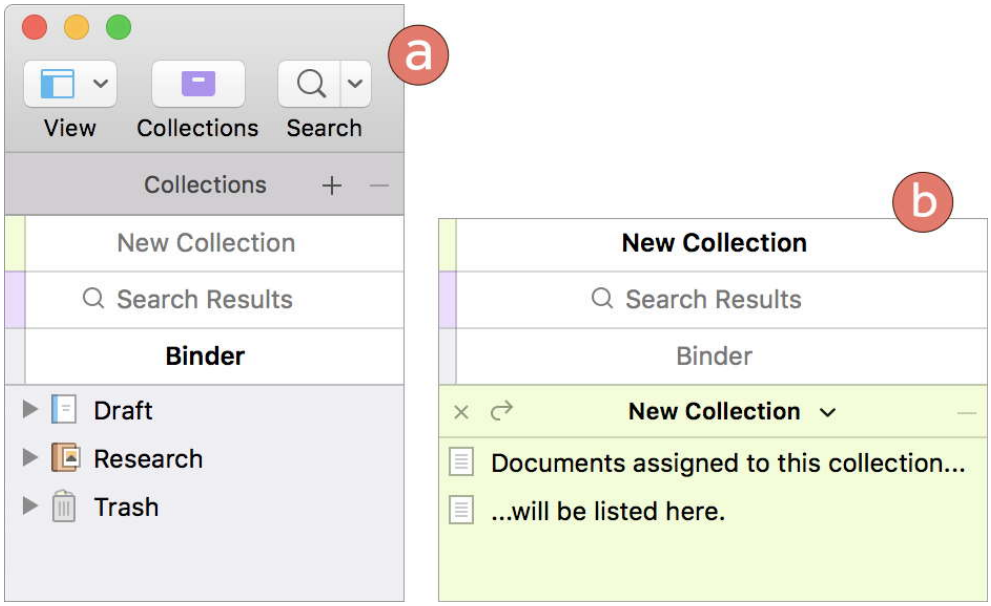


Figure 10.3: The collection tab list with (a) the binder selected and (b) a collection.

Each entry in the tab list represents a single collection. In the figure (Figure 10.3), the "Binder" tab is selected on the side marked (a). While not a true collection, this is one way you can navigate back to the main binder after viewing one. On the inset marked (b) we have the "New Collection" tab selected. Also take note of the added optional toolbar button to the right of the View button that can toggle the collection list with a single click.

Usage of the tab to switch between and manage your collections is as follows:

— Click on any tab to select it; when a tab is selected, it will set the background colour of the binder background and the name of the collection in the tab list will be printed in bold and black text.

  Additionally, a header bar (using the collection colour) will appear between the tab list and the content area of the sidebar, printing the name of the active tab and providing a few utility buttons (we'll go over those in the next section).

— You could hide the collections interface at this point and continue working in that tab. The background colour of the sidebar and the header bar will help remind you that you are not in the full Binder.

— To rename a collection: double-click on its name in the collection tab list, revise the text and press Return to confirm the changes. You can also rename a collection from the editor header bar, when viewing its contents

in the main editor window. The "Binder" and "Search Results" tabs cannot be renamed.

— When first created, a collection will be assigned with an automatically generated colour, but you can pick your own by clicking the downward facing chevron button, which can be seen in Figure 10.3 to the right of the label "New Collection".

— The change the order of any tabs in the list, simply drag and drop to move them up or down in the list. This will also impact their appearance in various menus throughout Scrivener, such as Navigate ▸ Collections.

The Collection Header Bar

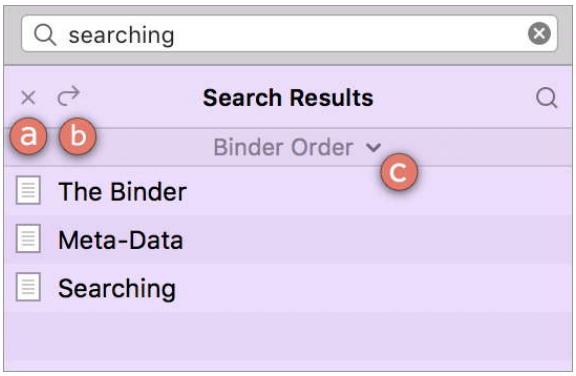


Figure 10.4: Search results header: (a) close and return to binder, (b) load results in editor and (c) sort results.

The header bar which appears between the tab list interface and the main content area (or simply at the very top when the list is closed) contains a number of useful functions (Figure 10.4). In the figure, we're viewing the built-in "Search Results" list, but the same buttons will be found in all collections.

a) The × button will dismiss the current collection view and return you to the main binder.

b) Click the ↱ button to load the contents of the collection sidebar into the editor area on the right (the targeted editor will be used when the interface is split). This will treat the collection in a similar fashion as viewing a folder.

c) Search collections, such as the "Search Results" list depicted in the figure, can also be sorted (section 10.2.4). Sort order can also be adjusted with the Touch Bar.

The magnifying glass icon to the right of the bar is a status indicator, letting you know that the contents of the list below are being dynamically generated as a result of a search result (in this case it should be terribly obvious, but once you

start making saved search collections it might not be). If the project search field (seen at the top of the figure) is displayed (click the Search button in the toolbar to toggle its visibility, or use the ⇧⌘F shortcut), the criteria by which those items are assembled can be examined.

Lastly, collections you create can have their colour modified by clicking a chevron icon that will appear directly to the right of the name in the header area (it can be seen in the inset marked (b) in the prior figure (Figure 10.3). The search results and binder tabs use a unified colour across all projects, and can be adjusted in the Colors tab of the Appearance: Binder preference pane (subsection B.5.2).

Disappearing Collection Tabs

On account of how scrollbars are hidden by default on a Mac, unless scrolling, it may not be obvious that the Collection tab interface can be scrolled, causing tabs to mysteriously disappear. If you do not like this behaviour, you can change your Mac's system settings in the General preference pane to always show scrollbars.

Viewing the Contents of a Collection in the Editor

The contents of a collection are not stuck within the sidebar. If you would like to make use of the rich capabilities afforded by the main editor's group view modes, there are three ways to do so:

1. While viewing a collection it will be simplest to click the ↱ button in the collection header bar. If you hold down the Option key when clicking on it, the list will be loaded in the other split, opening one if necessary.
2. You can also load a collection into the editor at any time via the Navigate ▸ Go To ▸ Collections ▸ submenu. No sidebar necessary!
3. The latter submenu is also accessible from the header bar contextual menu (section 8.1.1), under "Go To Collection".

Once loaded into an editor, the collection will function similarly to viewing a folder in the editor. It will remember settings you apply to it, such as whether you used label view or freeform corkboard (the position of the cards will be saved unique to that collection), if a view mode has been locked to it and so forth. When you navigate away to something else, you can return to it with the history feature as well.

Why do the Editor and Sidebar Mismatch?

In the case of lists that were the result of a search, the contents of the list imported into the editor will match the latest search results, and thus may differ from what you see in the sidebar if it has been opened for a while. The editor list will refresh whenever it is viewed (including when moving past it via the history feature), as well the sidebar, by switching from one tab to another. Rest assured, if you just opened a tab or list into the editor, it's accurate.

If you would prefer to view only a portion of the collection in the main editor you can simply select those items from within the list in the same way you might do so in the binder, forming a multiple selection (section 6.4).

10.2.2 Standard Collections

Standard collections are just the ticket for storing ad hoc lists of items. You have full control over what is listed within them as well as the order in which they appear. There are a few ways to create a new collection:

1. Click the + button in the tab list header, marked (a) in Figure 10.5. Any items that you have selected in the active view (including the binder, search results, or even other collections), will be automatically added to it. It is perfectly fine to create an empty collection from no selection, too.
2. You can also create a new collection at any time by using the Documents ▸ Add to Collection ▸ New Collection menu command. As above, any selected items in the current view (or the current text document you are editing) will be added to the new collection. The tab list will open, and your cursor will be placed in the collection's label area so that it can be name.

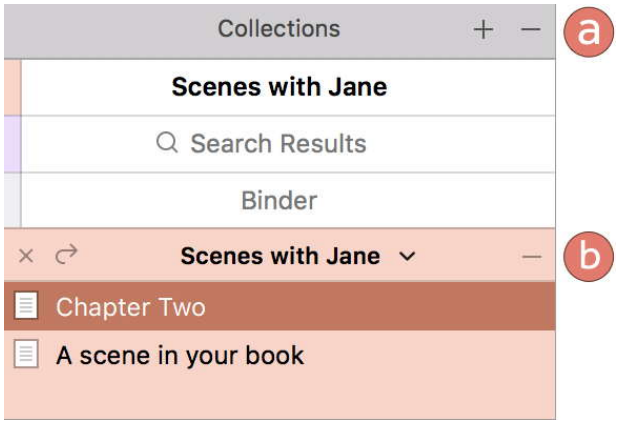


Figure 10.5: The standard collection interface.

Once a collection has been created, you can add or remove items to it, so it is not critical to select everything before you create it.

Adding To and Managing a Collection List

Here are the ways you can add items to an existing collection:

— With the tab list revealed, drag and drop a selection of icons from anywhere icons can be dragged from. They will be added to the bottom of the collection list.

  If you hold over the target tab for a moment, Scrivener will switch to the tab allowing you to drop the items precisely where you want to place them in the list.

— In variation to the above, if you hold down the Option key while dragging a container to a collection tab, it and all of its descendent items will be added.

— Use the Documents ▸ Add to Collection submenu. This works on selected items or on the current text document.

— In the binder sidebar, even from other collections, you can right-click on item(s) to access the "Add to Collection" menu.

In all cases, if an item already exists in the collection it will not be added again, and its original position will not be changed, so it is safe to err on the side of "over-selection".

Items can be reordered within the list using click and drag, or the same movement key combinations used in the binder (^⌘↑ for up, and ^⌘↓ for down). Since there is no hierarchy in a Collection, you will not be able to nest items.

New items created in a collection

Within a standard collection, you may create new items using all of the ordinary tools available for doing so. Since collections are uncoupled from the binder structure in every way, new items will be placed into a folder, created for you if necessary, with a name corresponding to the collection title they were created from. These folders will be created at the top level of the binder, at the bottom of the list, above the Trash. An example might be a new text file called "Joseph" in the "Characters" collection. When you return to the binder, you will find a new text file called "Joseph" in a folder named "Characters (Unsorted)". You can freely move this new document wherever you'd like.

It's important to note that this is a one-way process. You cannot add new items to the collection by adding them to the folder in the binder.

Removing Items from the List

Remove items by selecting them in the collection sidebar, and then clicking the − button in the lower header bar marked (b), or by simply pressing Delete on your keyboard.

> **Trashing Items From a Collection**
>
> To not only remove an item from a collection but send it to the trash as well, then use the ⌘Delete shortcut.

Deleting the Collection

To delete the entire collection:

1. Select the collection from the tab list that you wish to remove.
2. Click the − button in the upper title bar, marked (a).
3. You will need to confirm this action as there is no way to undo it.

Backing Up Collections

You can "back up" a standard collection by storing the list into another document's bookmark list. This can be useful if you have a lot of collections bulking up the list, and have a few that are seldom used. Here is an example of how this can be done:

1. Create a new document in the binder to store your collection list.
2. Open the inspector sidebar and click on the bookmark tab (section 13.4), or press ^⌥⌘N twice. If the list is not showing "Document Bookmarks" click the header bar to switch to them, or press ⌘6.
3. Use Navigate ▸ Lock in Place (⌥⌘L) to lock the editor so that you can freely work in the sidebar.
4. Use the Navigate ▸ Collections ▸ submenu to load the collection you wish to save into the sidebar if necessary.
5. Select all of the items you wish to save and drag them into the document's bookmark list.

Later on, if you ever wish to restore this collection after having removed it by adding a new collection and then dragging and dropping the bookmark list into the collection sidebar.

10.2.3 Search Result Collection

If you've used the project search feature (section 11.1) before, then you've been using a collection without perhaps realising it. Search results are placed into a special built-in collection every time you run a search, and the criteria of your search are saved into it along with the project. This means the last search that you ran will always be available to you, even after you have rebooted your computer.

Like the "Binder" tab in the collection tab list, the "Search Result" tab cannot be deleted or renamed, and there will always be one included with every new project, even when no search is currently in use.

> **Upgrading from Scrivener 2**
>
> In previous versions of Scrivener you could display additional columns in search result style collection sidebars. This feature has been removed, but it is still easy to get the text over into one of the editors, where the full outliner can be used to display and sort by the the kinds of metadata that were once here—never mind everything else the editor view can do.

When you click on the Search Results tab, the previous search criteria will be loaded into the search bar tool, allowing you to further tweak the results if you desire. Since search results are a direct product of search criteria, you cannot manually add, remove, or change the order of items from this collection, as you can with the standard type.

If you're looking for ways to save your searches more permanently, then read on!

10.2.4 Saved Search Result Collections

When you find yourself running the same few project searches over and over, it's probably a good time to learn how to save those into their own tabs so that you can call up the search results—and all of the search settings that went into gathering them—with a single click. To create a new saved search:

1. You'll need to start with an active search of any sort. This can be from the "Search Result" tab or another saved search collection. You might need to reveal the project search field if it is not visible. Click the search button in the toolbar, or use the ⇧⌘F shortcut.

   If you are unsure of how to get a search started, refer to Project Search (section 11.1). All of the settings you select from the magnifying glass menu as well as what you typed in will be saved.[7]

[7] There is one exception: searches performed using the "Binder Selection Only" option cannot be saved for future use, because the binder selection is a temporary state which changes whenever

2. With the project search field visible, click the magnifying glass icon to the left of where you would normally type in your search term, and choose the "Save Search as Collection…" command at the bottom of the option list.
3. You will be asked by what name you wish to remember this search by. Click the OK button to proceed.

New search collections will use a randomly generated colour, but like standard collections you can change the colour by clicking on the downward-facing chevron button to the direct right of the collection's label in the coloured header bar. If the search was set to "Search in" either the "Label" or "Keyword", and you were searching for only one label or keyword, the associated colour of the metadata you are looking for will be used for the tab's colour as a convenience.

The contents of a saved search are dynamic, but won't change constantly while you are looking at it. Every time the tab is returned to, the saved criteria will be checked against the current state of the project and refreshed. So to refresh a search tab: view another tab and then return.

Sorting the Results in the Sidebar

The contents of search lists can be sorted by clicking on the button marked (c) in Figure 10.4. All search lists share the same sort settings. The following criteria are available:

— *Binder Order*: this is the default setting. No sorting will be done on the list, with each item listed in the order they appear within the binder from top to bottom.

— *Sort by Title*: the list will be sorted by the given names of items. If titles are changed while working in the list, it will keep itself sorted dynamically.

— *Sort by Date*: which in this case refers to the created date of the item.

Below these options you will find a toggle for switching between ascending and descending sort order.

Need More?

If you require more settings, or wish to sort by another type of metadata entirely, then load the search result (section 10.2.1) into one of the editor splits and use the outliner tool to sort (subsection 8.3.4) or further filter the search results (section 11.4).

---

you click in the binder. If you find yourself unable to save a search, make sure this option is disabled.

Updating the Saved Search Query

You can refine or modify the search settings stored into a saved search collection:

1. You will need to start with an active search. Most likely you will wish to do so by clicking on the tab you want to modify.
2. Open the project search field if it is not visible: click the search button in the toolbar, or use the ⇧⌘F shortcut.
3. Modify the search settings with the magnifying glass menu. Your changes will be automatically saved into the collection.

   You may notice that if you change the search text using the above steps you'll be bumped over to Search Results, as the search text cannot be edited directly. This is fine, you can use the following checklist to update a collection with new search terms.

To update a saved search collection with the current project search settings directly:

1. If necessary, open the collections tab list with View ▸ Show Collections.
2. Right-click on the saved search collection in the tab list, and select the "Update Saved Search to Use Current Search Settings" command.

   All settings and text used in the current project search will replace the current saved search settings.

Converting a Saved Search to a Standard Collection

Given their dynamic nature, the contents of a saved search list cannot be added to, removed from or reorganised (outside of sorting). If you wish to "freeze" a search list so that you can play with it freely, or simply to store it for later reference, there are two ways you can do so:

1. Convert the saved search to a standard type: This will destroy the saved search, so only use this method if you no longer need the search criteria. To convert a saved search result to a standard collection, select the tab in the sidebar, and use the menu command, Navigate ▸ Collections ▸ Convert to Standard Collection.
2. Copy the contents into a new collection: This is quite easy to do. Select all of the items (⌘A) in the search result list, and click the + button in the collection header bar to create a new Standard Collection from the current selection. The items will be listed in the new collection in the order they were sorted within the search list.

Deleting Saved Search Collections

Saved search collections can be removed in the same fashion as standard collections, by selecting the tab and clicking the – button, marked (a) in Figure 10.5. Since clicking on a saved search collection automatically loads its search parameters into the project search tool, you can effectively undo this by switching the Search Results tab and recreating the collection from the magnifying glass menu.

10.2.5 Back to the Binder

Often you might wish to know the overall disposition of a collection list in the binder, or where a select few items in the list are located in the overall structure of the project. The Navigate ▸ Reveal in Binder menu command (⌥⌘R) works from collections and search results. Since this command can be used on many items at once, it makes for a handy way to see what *isn't* in large collections, too.

It is possible to select some or all of the items in a collection and instruct Scrivener to gather them all together into one spot, based on the order of their appearance in the collection. There are several ways of doing so:

— *Drag & drop*: you will need to have the collection tab interface visible. Simply select the files you wish to gather, drag them to the binder tab and hold for a moment. The binder will activate, and you can drop the selection wherever you please.

  If the **Option-dragging creates duplicates** setting is enabled in the Behaviors: Dragging & Dropping preference pane, holding down the Option key will work here to duplicate the dragged items instead of moving them.

— Documents ▸ Move To ▸ submenu: select the items you wish to gather together, and either use the main application menu or the contextual menu to move them to a selected item or container. This action will operate in the background, leaving your focus in the collection.

  Likewise, Documents ▸ Copy To ▸ can be used if you would prefer to duplicate the items rather than move them.

— Use the Documents ▸ New Folder from Selection menu command (to assemble them into a new folder in the Binder.

Experienced users of outliner style programs will recognise this ability as "mark and gather". The marking phase is done by assigning documents to a collection. Moving them back out to the binder then gathers them quickly into one focussed spot. This can be an extremely useful technique for some workflows.

This technique can also be useful for implementing an experimental text flow. If a chapter or section just doesn't read right yet, you can quickly create a new collection with the contents of that section as files and then reorganise the flow using the collection's ability to view itself as a corkboard or outliner—and of course reading the text with Scrivenings mode. Once you are satisfied with the new

layout, select the contents of the collection and drag them back into the original folder in the binder using the above method. The items will be re-organised for you back into that folder and become the new book structure.

Return to chapter ⤴

10.3 Project and Document Bookmarks

No doubt you've encountered the concept of bookmarking web sites in your browser. We could say that at its most basic level, bookmarking binder items is similar to this concept in that you can create and organise lists of important or frequently used items, making it easier to use them or navigate to them in the future. They can also be used to refer to files on your disk or resources on the 'net.

> **Upgrading from Scrivener 2**
>
> If you're searching this manual for "Favorites", "Project Notes", "Project References" or "Document References", you might want to head on over to the article addressing what has become of these features in the What's New section (section E.5). The section you are reading now addresses a new feature which combines all of the above into one single cohesive system.

The concept of a bookmark in Scrivener goes a little further than what you might be used to with your web browser. In this section we'll go over how bookmarks can be used as a universally available scratch pad, either used in a separate window or embedded into the inspector sidebar on the right hand side of the main project window. They can also be used to boost your efficiency when using those types of menus that work with or target binder items, such as the Navigate ▸ Go To ▸ submenu, but prominently placing bookmarked items at the top of the list.

You might be wondering what the difference is between a project bookmark and a document bookmark. In simple terms, project bookmarks are available throughout the entire project and work to elevate the accessibility of those items in multiple contexts. Document bookmarks on the other hand have a more limited role, being solely available from the document bookmarks list in the inspector.

We'll go into the specifics of adding, removing and managing bookmarks after covering the different areas bookmarks are used within. So if you are brand new to the concept and want to have a little to play with while reading through the introduction here, do the following:

1. Use the Project ▸ Show Project Bookmarks menu command (⇧⌘B).

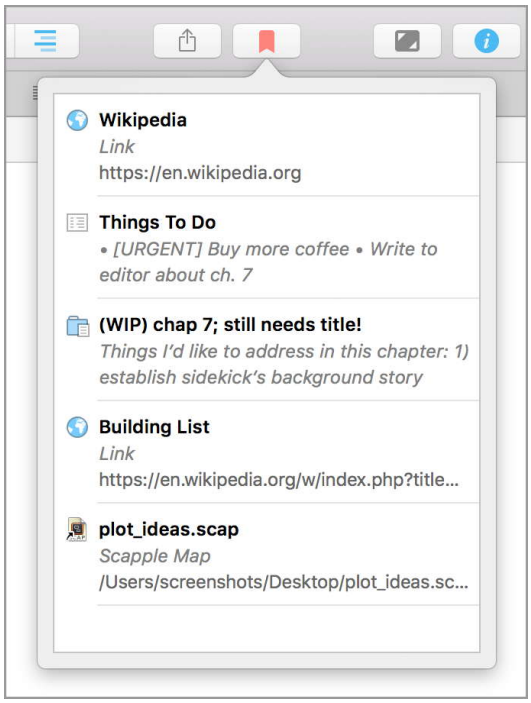


Figure 10.6: Bookmarks are readily accessible via a single click from this toolbar button.

2. As instructed within the blank area of this panel, simply drag and drop anything from your binder into the list.

You've now got bookmarks. That's how easy it is to get started, so with a little to work with, let's take a look at the various areas of the interface these bookmarks can work within.

10.3.1 The Bookmark List and Floating Panel

The list that pops up when you click the bookmark icon is simple, but provides for more flexibility than you might imagine at first glance:

— Click on a bookmark to load a binder item in the active editor.

— Double-click to load the bookmark into a Quick Reference panel. You will also need to double-click if the bookmark is to an external resource such as a web page or file on your disk.

— Hold down the Option key when clicking to load the bookmark in the inactive editor split, opening one if necessary to do so.

— Bookmarked folders will have a small chevron icon to the right of the folder name in the list. Click on this button to reveal the contents of that folder in a menu, and select from it to navigate the editor to the chosen subdoc-

ument. The above method to load your selection in the other editor will work here, too.

— Right-click to access extensive contexual menu options. These are the same as those offered when right-clicking on bookmarks in the inspector (subsection 13.4.2).

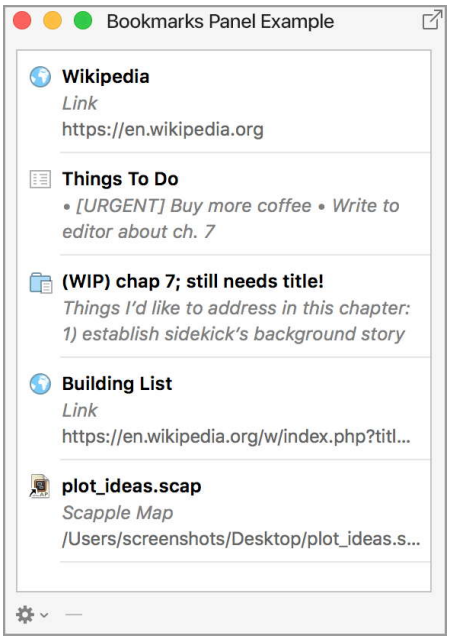


Figure 10.7: The floating bookmark list can be converted to a browser.

While clicking on a button in the toolbar can be handy, there may be times where you want the list to hang around a bit more permanently. The easiest way to do so will be to grab any edge of the toolbar list with your mouse and drag way—the panel will "tear off" of the toolbar and become a floating window at that point. Alternatively use the Project ▸ Show Project Bookmarks menu command (⇧⌘B) to bring up the panel directly.

In addition to all of the capabilities listed above, the floating panel also provides a few additional functions:

— The ✲ button in the lower left makes it easy to add a link if you don't have the original in front of you handy for drag and drop.

— The – button is used to delete selected bookmarks.

— In the upper right-hand corner is a button that converts this floating panel into a Quick Reference window, which serves as a bookmark browser.

10.3.2 The Inspector's Bookmarks Tab

There is already a whole section dedicated to documenting this tab, but given their crucial role in how bookmarks can be used in your project, it would be

important to outline what it can do for you here, before sending you over to that section for further information.

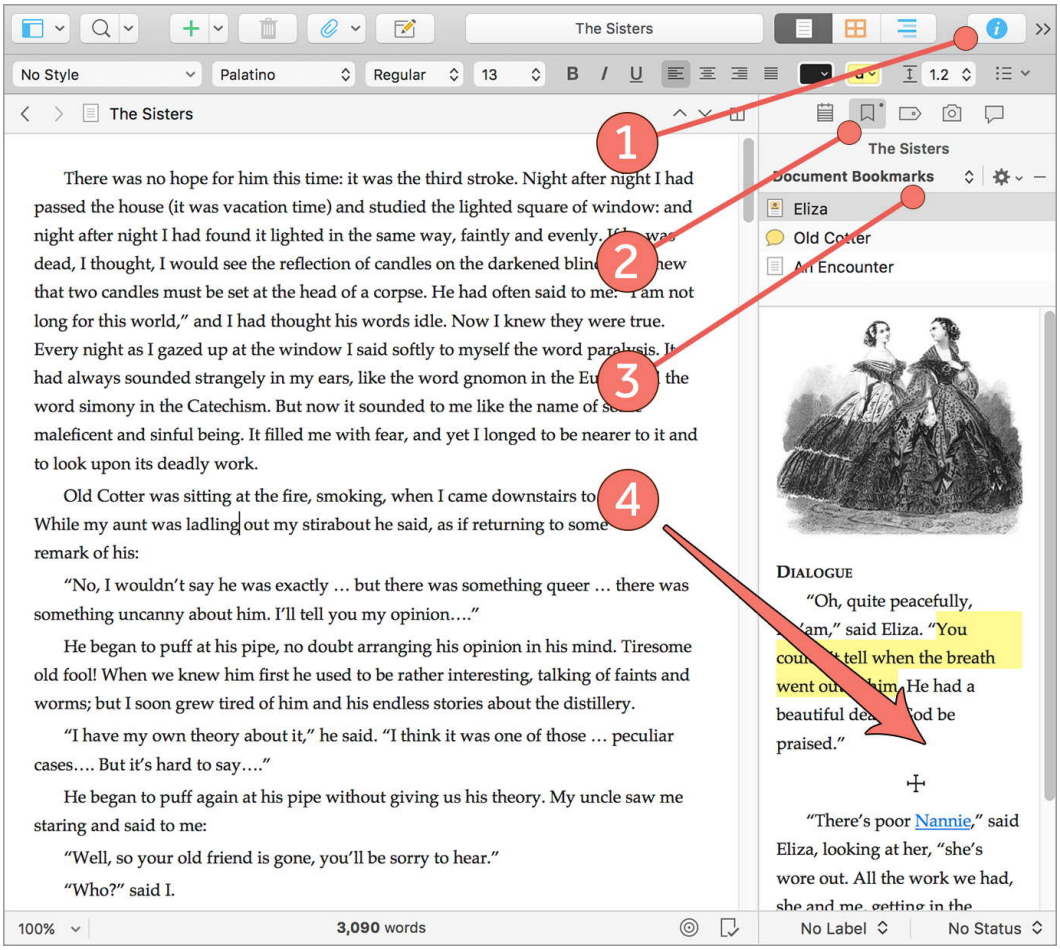


Figure 10.8: View bookmarks in the inspector by (1) revealing the inspector, (2) switching to the bookmarks tab, (3) selection between project or document bookmarks and (4) using the preview area to work with the content of the bookmark, selected in the list above.

The inspector, as a tool that is embedded directly into the project window as a sidebar, is ideal for playing host to your bookmarks. If you used "project notes" in prior versions of Scrivener, you already know how useful it can be to have a global notepad in the sidebar, at your fingertips no matter what part of the project you are working on. Project bookmarks give you that same capability, but with the added flexibility of promoting any binder item at all to this global list. Additionally, the older "References" feature which made it possible to cross-reference between sections of your work and research has been amped up now that you can view and edit these so-called Document Bookmarks right in the same area of the inspector.

As with older versions of Scrivener, whenever you link to or bookmark an item from another a back-link bookmark will be created, letting you know which

items have linked to something simply by looking at the document bookmark list.

We will be focussing primarily on project bookmarks in this section, as nearly everything you would do with document bookmarks is confined to the inspector's Bookmarks Tab (section 13.4).

10.3.3 Working with Bookmarks in a Quick Reference Panel

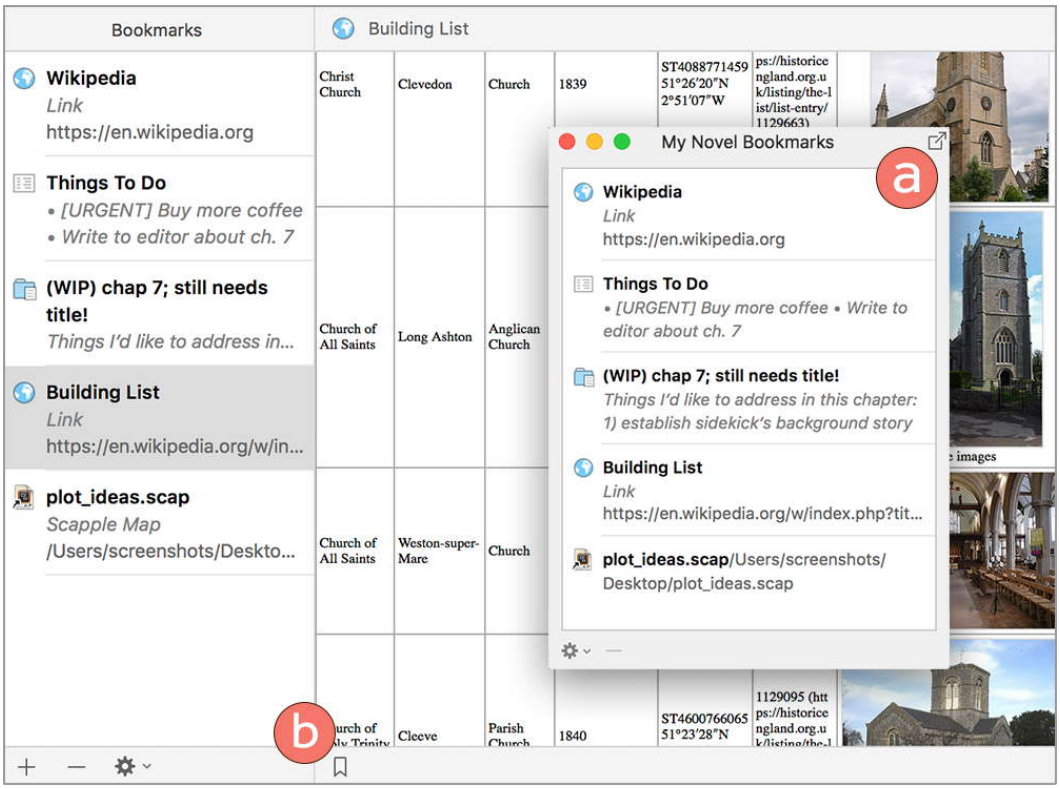


Figure 10.9: The bookmark list can be expanded into a Quick Reference panel for quick access to your notes and research.

There are three ways to browse bookmarks in a separate window, two of which are demonstrated in Figure 10.9:

1. Click the button in the upper right-hand corner of a floating bookmark pane, marked (a), to convert it into a Quick Reference window with a bookmark sidebar. The same command and shortcut that brings up the floating bookmark list will, if the list is the active window, convert it to a Quick Reference window.
2. Any existing Quick Reference panel can browse project bookmarks by clicking the bookmark button in the lower left-hand corner, marked (b).

You can also use the same Project ▸ Show Project Bookmarks command, and ⇧⌘B keyboard shortcut, to toggle this sidebar on and off when the active window is a Quick Ref panel.

3. If you want to get straight to this type of window without going through other windows or panels first, hold down the Option key while using the Project menu, and select the "Show Project Bookmarks as Quick Reference" menu command that will appear. This can also be done from the keyboard, with the ⇧⌥⌘B shortcut.

The main difference with the bookmark list in a Quick Ref panel and in its floating form is that it will acquire a + button in its footer bar. This is a unique capability to this form of working with bookmarks. Instead of creating a bookmark from an existing source you will be creating a brand new document in the binder.

In most cases you will be asked where you would like to save the new file in the binder. After you choose a location and click the OK button, the new note will be created into the sidebar and you can begin typing in a name for it. Click into the editor area of the Quick Reference panel, or press the Return key to start jotting down your notes.

Default New Bookmarks Folder

When creating a new bookmark note from the sidebar you will be asked where you would like to save the new note. This part of the process can be bypassed by clicking the **Do not ask again** checkbox below the group selection tool. From that point on, new notes created in the sidebar will automatically be filed into this folder for you.

If you prefer to have a "scratch pad" area of your project rather than using the feature to elevate globally notable files no matter where they are, you might prefer this behaviour enough to have it be a default. Since it is a project-specific setting (the folder you choose can only exist in one project) there is no way to change the way this works universally—however since it *is* a project setting that means you can set up in your own custom project templates (subsection 5.4.3).

In doing so, or if you ever change your mind about using a centralised folder in the future, you could set this up with Project ▸ Project Settings..., under the Special Folders pane (section C.7). Select the "Ask Every Time" option at the top of the **Default New Bookmarks Folder** dropdown to disable auto-filing.

10.3.4 Managing Bookmarks

With the main areas that bookmarks are worked with now covered, the following serves as a reference for performing basic management tasks throughout these elements of the project.

Adding and Removing Bookmarks

There are several ways to add a bookmark to something:

— Click on the bookmark button in the main application toolbar (Figure 10.6), or use the ⇧⌘B shortcut to bring up the list in a floating panel. Drag and drop the resource you wish to bookmark into the list.

— When using the floating bookmark panel, you can also click on the ✿ button to add new bookmarks. This button is functionally identical to the button featured in the inspector (subsection 13.4.1). It also appears in the Quick Reference bookmark sidebar, discussed in the following section.

— Use the Paste command (⌘V) if you have a URI of any type on the clipboard. This includes URLs to the web, file links and protocol links of any sort—including external links copied from other Scrivener projects (though you can just drag from the other project's binder as well).

— While editing a document in the text editor (or the section you are working on within a larger Scrivenings session), bookmark it with the Documents ▸ Add to Project Bookmarks menu command. You can also use this command on any selected item in the binder sidebar, corkboard or outliner views. This command is also available from the binder in the right-click contextual menu.

— If you own a keyboard with a Touch Bar you can add a project bookmark toggle button to the text editing context.

Removing Bookmarks

— In the bookmark list, the floating panel version of it, Quick Reference sidebar and inspector bookmark tab, you can always delete a selected bookmark with the same ⌘ Delete shortcut you would use to trash an item from the binder sidebar.

— Another universal method is to right-click on the bookmark you wish to remove and select the "Delete Selected Bookmark" contextual menu command.

— As with the menu command to add a bookmark to one or many files, outlined above, the Documents ▸ Remove from Project Bookmarks command is available for the active editor, binder sidebar, corkboard and outliner views. This command is also available in the binder as a right-click right-click contextual menu.

— With the Touch Bar, removing an item from the bookmark list is as simple as tapping the same button you used to add it.

Editing Bookmark Titles

When adding a bookmark to an external resource, a title will be generated for you depending on the type of link it is. If it is a bare URL you might get a generic title such as "Link", but files will use the file name for the title, and hyperlinks will try to use website titles or the link title automatically.

You may want to edit the title once you've added a link by right-clicking on the bookmark and selecting the "Edit Bookmark" option. If you own a keyboard with a Touch Bar, you will find a pencil icon when your keyboard focus is in a bookmark list. Tap this to edit bookmarks. Within the Quick Reference bookmarks sidebar you can double-click on bookmarks to edit their titles or rename the items they link to.

There are additional tools for managing bookmarks that are exclusively available from the inspector (section 13.4.2).

Organising Bookmarks

Bookmarks are organised by default in the order they were created. You can adjust their organisation to suit your uses of them. The order of bookmarks within the lists will impacts their order in every context where they are displayed. In some cases it can be a productivity boost to keep frequently used items near the top of the list, where they will be more easily accessed from tools like Navigate ▸ Go To ▸ submenu, which only displays the top five bookmarks.

— Click and drag to reorder bookmarks amongst themselves. This works from within any of the editable bookmark lists in the software, including the toolbar popup.

— Bookmarks can be sorted alphabetically from within the list. With the keyboard focus in the list, use the Edit ▸ Sort ▸ Sort Ascending or Descending menu commands. This is a one-off command; you will need to reuse it if you wish to sort bookmarks again.

Renaming Internal Bookmarks is Renaming Items

Be aware that if you rename a bookmark that is a link to another item in the project binder, you will be renaming the *original item* as well. This is in fact less like a "bookmark" in the traditional sense of the word, and more like the kind of entry you will find in a collection: an additional placement of original the binder item.

Copying Links and URLs

Sometimes you just want a link from the bookmark, rather than doing anything with it immediately. Perhaps you want to store it in another program or load a URL into a browser that isn't your default.

1. To copy a hyperlink, suitable for pasting into documents as a clickable link with a friendly title, you can use the ⌘C shortcut on any selected bookmark:
   - File and URL links will use the bookmark name for the link title.
   - Document bookmarks will use the document name. When pasted back into a text editor in the same project they will become document links. If you paste them into another project or into any other external context they will become external links to that binder item. I.e. clicking on the link in a word processing file in LibreOffice would open the item in your project directly.
2. To copy just the URL as a plain-text address, right-click on the bookmark in any context and use the "Copy URL" contextual menu command.

If you're looking for ways to copy the bookmark itself—perhaps to move a document bookmark to a project bookmark, or duplicate a list of bookmarks to another document, the inspector tab will be the best place for doing so (section 10.3.4).

Copying Bookmarks Between Items

Bookmarks of all types can be freely copied and pasted between lists, both project and document alike. They can also be dragged from one list to another—a possibility between Document Bookmarks when Quick Reference panels (section 12.6) are in play.

Collection lists (section 10.2) and internal bookmarks are very similar to one another. You can almost think of bookmarks as being private collection list each item in the binder has available for use. You can freely drag and drop items between bookmark and collection lists.

Sharing Bookmarks Between Projects

If you drag items from the binder of one project, into the bookmark tab of another (either project or document bookmarks are a valid target), this will store a special external direct link to the individual items you dropped. Double-clicking this bookmark will load the project if necessary and open the item you linked to within it. Read more about cross-project item links in External Links (subsection 10.1.6).

External bookmarks can be copied and pasted from one project's bookmark tab to another, and in most cases this will be the preferable way to doing so, as dragging bookmarks between projects will cause a loss of their titling information.

10.3.5 Bookmarks in Binder Item Menus

Adding items as project bookmarks will elevate their presence in some menus that allow you to select binder items from a submenu hierarchy:

— Edit ▸ Link to Document: for creating hyperlinks to documents. This menu is also available when right-clicking on text in the editor.

— Navigate ▸ Go To: navigation the active editor to any spot in the binder. This menu also appears in the editor header bar icon menu and the composition mode control strip along the bottom of the screen.

— Navigate ▸ Quick Reference: opens a Quick Reference panel to the selected item.

Return to chapter ⤴

10.4 Organising with Metadata

Documents of any type in Scrivener can have various metadata associated with them. Metadata is a way of talking about something without changing it directly. A simple example from the analogue world could be a Post-It note on a paper-clipped stack of paper. The Post-It note is a kind of metadata, as is the paper-clip. Just as with a piece of colourful tape affixed to the edge of an envelope, metadata can help us more easily search for tagged items, either visually or through the use of digital methods, like the project search (section 11.1) tool.

10.4.1 Metadata Types

Let's go over each of the types of metadata that Scrivener provides to items, the basics of how they can be set up, assigned, and how they will in general be displayed within the software. The main interface for viewing and editing all of the metadata for a given document is the inspector; here we will focus on the bigger picture, so you should refer to that section for the details on using that pane (chapter 13).

The Title of an Item

The most important piece of metadata any item has is so fundamental you might not even instinctually think of it as being metadata: its title. The title will be used to identify the document in the many views, menus, and export methods.

Refer to Titles and Adaptive Naming (section 7.3) for further information on using titles.

The Synopsis of an Item

The synopsis is a plain-text field, primarily intended to be a short summation of the contents of the document, though how you choose to use it is entirely up to you. The synopsis is displayed in three prominent areas:

1. Corkboard: used to display the content area of the card.
2. Outliner: will be placed beneath the title.
3. Inspector: if you need to reference an item's synopsis without finding it in a list somewhere, the inspector sidebar's Notes tab, or the special Synopsis split in a Quick Reference panel will be the best approach.

Instead of the text synopsis, you can elect to use an image to represent a document on the corkboard. This image will be used on the corkboard and in the inspector, but whatever text exists in the standard synopsis field will be used in the outliner, as described above, and in the various export and print options that include a synopsis field.

Setting Up Label & Status

The next two forms of metadata, Label and Status, are flexible in how you can refer to them within a project. You can give these fields custom names and the interface and menu commands will adjust accordingly. If you wanted, you could have "POV" and "Location" instead of label and status, or "Focus" and "Type", or "Monkeys" and "Bananas" for that matter.

The documentation will of course continue to refer to them as labels and status for the sake of simplicity. They are also, after the title, the most visible items in the entire interface. They can be represented in the corkboard and outliner views, are always present along the bottom of the inspector and Quick Reference panels and are given priority placement in most of the printing methods. Labels, having an associated colour, can also optionally tint various interface elements, such as document icons, outliner rows and index cards, via the View ▸ Use Label Color In ▸ submenu.

All new projects come with a few stock labels and status, but you will most likely wish to add your own or change them completely. Use the Project ▸ Project Settings... menu command to access panes via Label List & Status List (section C.3).

As with all metadata, one of the primary purposes of using them is to make our searches more powerful and specific. In addition to regular project searches by label or status, you can also use these types to filter outliner & corkboard views (section 11.4).

Label Colours

Labels are one of the most flexible metadata types in terms of how visible they are in the interface, so you can pick a colour to reflect how prominent you wish

them to be. For example, richer more vibrant colours can be used among pastel choices to indicated tension, or priority.

Label colour can be expressed in the following ways:

— On the corkboard, index cards will bear a strip along the left edge of the card indicating label assignment. Use View ▸ Corkboard Options ▸ Show Label Color Along Edges (^⌘P) to toggle their visibility.

— There is an entire corkboard mode dedicated to working with labels. Read more about that in Arrange by Label (subsection 8.2.5).

— The outliner has the label column added as a default to all new projects.

— As previously mentioned, the View ▸ Use Label Color In ▸ submenu contains a number of options for tinting various areas of the project window or elements within it with the label colour:

  – *Binder*: A dot will be placed to the right of an item in the binder, signifying its label setting. If you prefer the binder row be highlighted, use the "Show as Background Color in Binder" option, below.

  – *Icons*: Only the icons will be tinted throughout the project. Wherever the icon for an item appears (such as in the editor header bar, next to the title in corkboard, in search results, and so forth) it will be tinted using the colour of the assigned label.

  – *Index Cards*: The entire background "paper" for index cards will be tinted using the assigned label colour. This includes the index card that appears at the top of the inspector.

  – *Outliner Rows*: The background for the entire row will be filled in with the label colour.

  – *Scrivenings Titles*: When View ▸ Text Editing ▸ Show Titles in Scrivenings is enabled, the line of text used to print the title of a document in Scrivenings mode will be tinted using the label colour of the corresponding document, giving you a valuable look at your labels colours directly in the text editor.

— When printing outlines or corkboards (chapter 26), label colour can be added for effect (it will be enabled by default on index card printouts).

Status Stamps

As with labels, the status field can have its representative name altered to suit your project's unique requirements. By default, this field represents the status of a document in terms of its completion, such as "To do" or "Rough draft", but this field can be used for whatever purpose you desire.

Unlike the label, there is no corresponding colour, and so its display potential is more limited. On the corkboard, they can be displayed as an optional "stamp"

across the face of the card with the View ▸ Corkboard Options ▸ Show Status Stamps menu toggle (^⌘S). In the outliner, the status is one of the default columns that comes with every new project.

Keywords

Each document can have a list of keywords associated with it (what you might be more familiar with as "tags" in some other programs). These are useful for making documents easily searchable—for instance, you can list all characters and locations connected with a scene in the keywords even if they are not mentioned explicitly in the text. Creative uses for keywords also include extended status control, plot management, and whatever else you can think of.

Their biggest advantage is in non-exclusive assignments. A document can only have one label, and thus be one type of thing at a time, but you can have as many keywords as you need assigned to one item, creating compound descriptions of items, and allowing for overlapping with other items that may not be similar, but yet still share some common attribute.

Keywords can be displayed in the following fashions:

— On index cards in the corkboard as coloured strips of tape along the right-hand side, with View ▸ Corkboard Options ▸ Show Keyword Colors (^⌘K).

— As an outliner column, where they can be listed by name with View ▸ Outliner Options ▸ Keywords, and as a sub-option as coloured squares, with .../as Color Chips.

— As an editable list, in the Keywords pane of the metadata inspector tab (subsection 13.5.3).

Refer to the following section for further information on using keywords (subsection 10.4.2).

Custom Metadata

For all of the things we couldn't think of. With four different types of fields and extensive support for searching and filtering, custom metadata picks up where the stock tools let off. Four different field types can be used to create however many fields you need in a dedicated form built into the inspector pane, and as sortable outliner columns or tools for filtering corkboard and outliner views and project searches:

— Text: a simple text field. This tool is great if you need to mark sections with a specific piece of information that is often different for each item in the outline, but has a common theme. Useful for bulk text with an optional word wrap, or short snippets of information. For example a non-fiction book on edible plants might use a text field to record the Latin name of the plant discussed in that outline item. This information is something

you would probably want to print in the book as well (and you could, with placeholders used to insert metadata into the text, but that's a more advanced topic!), but it might also be useful to you as an author to have a concise list of these names in the outliner as a dedicated column.

— Checkbox: the simple act of saying yes or no about a thing can yield a lot of flexibility in how you work. Create simple to-do list, mark when blog articles have been published or track phases of a large editing project in conjunction with searches by checkbox state.

— List: if you find yourself wishing you can add another dropdown field like label or status provides, this is the tool to use. Using a similar approach as the status field, you can add list items, organise them and choose a default.

— Date: store date and time information with an easy to use calendar interface coupled with a natural language recognition system that lets you type in things like "monday" and have the computer figure out for you when the next Monday will be. Use this to store publication dates, timeline information, deadlines and so on.

Get started with setting up your own fields by using the Project ▸ Project Settings... command, in the Custom Metadata pane, and be sure to check out the following references for further documentation on the various places where these fields are set up and used.

See Also...

— Overview: how custom metadata (section 10.4.1) can benefit your work.

— Settings: adjusting the available fields and their settings, in project settings (section C.4).

— Inspector: editing custom metadata on a per-item basis with a form built into the inspector (subsection 13.5.2).

— You can use list and checkbox type fields to filter outliner & corkboard views (section 11.4).

Exporting Metadata

All metadata can be exported in a variety of ways, usually text-based for maximum compatibility, so you needn't fear having important organisational information getting locked-in with the project format. Metadata can be exported in the following fashions:

— Compiling: when compiling the Draft, enabling metadata export is an optional feature of the Formatting pane. The various types of metadata will be exported into the Draft in a variety of ways best suited to the type of data involved. Additionally, Scrivener supports a broad array of placeholders

that can be used to fill in dozens of different types of metadata in precisely the format you require.

— File export: when exporting files from the Binder, metadata can be placed into optional "sidecar" plain-text files corresponding to the main document being exported. This method is used to preserve the integrity of the original document, and to enable metadata export in formats which would not otherwise allow text data to be inserted into it, such as mp3 files.

— Drag & Drop and Import: When dragging documents from one Scrivener project to another, or using the File ▸ Import ▸ Scrivener Project... menu command, all metadata will be preserved where possible and applicable. In some cases, like custom metadata, matching fields and list values will need to be prepared, as new field types will not be created in the target project with drag and drop.

  Refer to Copying Settings Between Projects (section C.1) for general information on transferring metadata *settings* from one project to another.

— Outline and Corkboard printing: both of these printing methods optionally allow metadata to be used in the printout. Unlike the above two examples, these allow for a more fine-grained approach, letting you export just the label or keywords if you want.

10.4.2 Using Keywords

Like labels, keywords have an associated colour which will be displayed beside the keyword in the inspector list, as an underscore in the relevant outliner column, and in the Project Keywords window (see below). Additionally, keywords can be used in the corkboard as small coloured tabs along the right-hand edge of the card. The visibility of these can be toggled with View ▸ Corkboard Options ▸ Show Keyword Colors (^⌘K).

Keywords are Case-Sensitive

When you assign keywords to documents from their inspector pane, they will automatically suggest any existing keywords from within the project. However you should be aware that keywords are case-sensitive, meaning that if you have a keyword called "Software" in the master list, but start typing in "sof" in the inspector, nothing will be suggested, and indeed adding it as "software" would create a new entry in the list.

Project Keywords Panel

The Project Keywords panel holds all of the keywords in use by the project. As you assign keywords to items, using the Inspector's Metadata tab (subsection 13.5.3), they will be added automatically to the central project list, making it

a complete reference of all keywords in use within the project (you can also create keywords for future use here, without assigning them to anything yet). The panel is also how you will handle bulk management of keyword assignments for many items at once.

You can access this pane via Project ▸ Show Project Keywords (⇧⌘K) or by double-clicking any of the keywords in the Inspector pane.

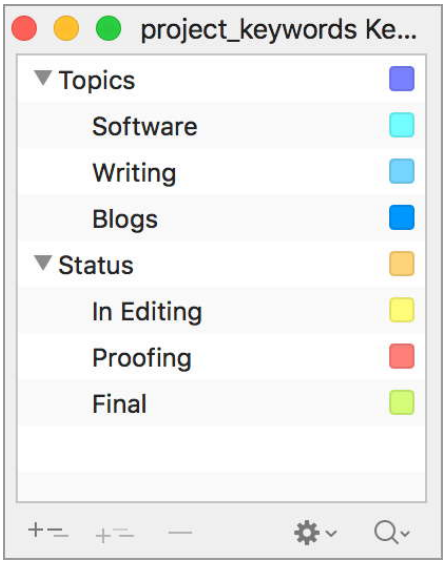


Figure 10.10: The Project Keywords panel shows all keywords in the project.

You can create or delete keywords by using the buttons at the bottom of the panel (Figure 10.10):

— The left-most button creates a new keyword as a sibling of the selected keyword. If we selected "Status" in the example and clicked this button the keyword would be on the same level as "Topics" and "Status.

  You can also create new sibling keywords with the Return key.

— The second button creates a new keyword as a child of the selection. With the same example but clicking this button, the keyword would be added along with "In Editing", "Proofing" and "Final".

— The third button deletes the selected keywords. If any of the keywords in the selection are assigned to items in the project you will be warned and asked to confirm the deletion.

Organising Keywords

The Project Keywords panel is a freeform work space where you can organise your keywords into groups and arrange them sequentially, by topic or however you see fit.

— To change the order of keywords within the list, use drag and drop, or the keyboard shortcuts from the Edit ▸ Move ▸ submenu.

— You can also drag any keyword onto any other keyword to nest keywords.

In the example figure, we have nested "Software", "Writing" and "Blogs" into a keyword called "Topics". This relationship is merely for organisation within this panel. The "Blog" keyword will not print that it is a "Topic" in other areas of the software, and indeed the "Topics" entry is itself a keyword that can be individually assigned to items.

Prefer Alphabetical Sorting?

While the keyword list will leave the organisation up to you, it is possible to manually sort the entire list alphabetically whenever you wish to do so, via the Edit ▸ Sort ▸ Sort Ascending or Sort Descending commands.

Assigning Keywords with the Panel

Use one of the following methods to assign keywords from the panel to documents in the main project window:

— Select and drag the desired keywords from the panel and drop them into the keywords pane in the inspector, a Quick Reference keyword split, the header view above the document editor or onto the document in the binder, corkboard or outliner.

— If there is a multiple selection in the binder, corkboard or outliner then all selected keywords will be applied to all selected documents.

— Hold the Option key down whilst dragging to drag not only the keyword but also any parent keyword(s) under which it is grouped.

— Right-click on the selected keywords (or use the ⚙ button) and select the "Add Keywords to Selected Documents" command.

Removing Keywords with the Panel

If you need to remove a specific keyword from an item, it will be easiest to do so from its keyword list in the inspector's metadata tab. However in cases where you need to remove a keyword from several documents at once, the project keywords panel is the best place to do so:

1. Select the documents you wish to remove the keyword from, in the binder, outliner or corkboard.
2. Open the Project Keywords panel and select the keywords you wish to remove from the selected documents.

3. Right-click on the selected keywords (or use the ⚙ button), and use the "Remove Keywords from Selected Documents" command.

Changing Keywords Globally

Keywords can be centrally managed from the project keywords panel. Intuitively, if you change the name of a keyword it will be altered throughout the entire project rather than becoming a new keyword that nothing is assigned to. Likewise changing the colour is a global setting. To do either, first use the Project ▸ Show Project Keywords menu command to load the keywords panel.

To change the name of a keyword:

1. Double-click on the name of it in the panel, or press the Esc key, and edit the label.
2. Click elsewhere to confirm your change, or press the Return or Esc key.

To modify the colour assigned to a keyword:

1. Double-click the colour chip itself.
2. This will bring up the standard colour chooser. Close this panel when you are done.
3. You can recolour several keywords in a row by leaving the panel open and selecting different keywords then choosing another colour.

Deleting a Keyword from the Project

To fully remove a keyword from a project, not only from the master keyword list but from every item that was using it as well, you will need to use the Project Keywords pane:

1. Open Project ▸ Show Project Keywords (⇧⌘K).
2. Select the keyword(s) you wish to remove. You can select multiple keywords with the Shift and Cmd keys.
3. Click the − button in the footer bar, or press the Delete key on your keyboard.

   For keywords that were not assigned to any items in the binder, this will be done silently and without confirmation. If a keyword is in use, then you will be warned and asked to confirm the removal.

Searching by Keywords

Although it is always possible to create your own searches for keywords by hand, the project keywords window provides a convenience for keyword usage in your

project. Simply click on the keywords you need to hunt down, and click the magnifying glass button in the bottom right-hand side of the panel.

There are optional search modes that can be accessed by right-clicking on the button. Upon selection of one of these modes, the desired search will be immediately used, or updated if the search list is already showing:

— *Search Keywords Only*: this is the default search mode. The project search tool will be set so that only keywords are scanned throughout the project.

— *Search All Content*: the limitation above is lifted, and the words represented by the keywords will be scanned for throughout all searchable areas of the project—in the main text editors, through binder titles, custom metadata, labels, synopses and so forth.

— *Show Document with No Keywords*: simply put, any documents in your project that have no keywords assigned to them will be returned in the result list.

If your keyword selection includes multiple keywords, then only those documents that contain all of the assigned keywords will be returned (boolean AND).

Are These Regular Project Searches?

Yes, and no. The search engine being used is of course the standard project search tool, and you will get results just like you do when searching a project normally. However the keyword search utility will not permanently change your search settings. If you dismiss a keywords search and then type in the same keyword again by hand into the project search tool, the search result will likely be different unless your settings just so happen to perfectly match one of the presets described above.

Importing and Exporting Keywords

If you would like to copy keywords from one project into another, follow these steps:

1. Open both projects, and in each, use the Project ▸ Show Project Keywords menu command to open their respective keyword lists.
2. Select the keywords you wish to copy from one list and drag and drop them into the other project's keyword list. Colour assignments will be preserved.

   This will always duplicate the keywords into the new project, so existing keywords with the same name will not be overwritten.

There may be cases where you want to either preserve your keywords outside of Scrivener, or bring in a list of words to be used as keywords. Scrivener supports dragging or copying and pasting into, and out of, any text editor as well.

— *Export*: select the keywords you want to export and drag or copy and paste them into any text editor in any program.

  This will create a comma-delineated list of terms.

— *Import*: as you might have suspected, the list of terms you created when exporting keywords can be dragged or pasted straight back into a Project Keywords pane to import them as keywords.

  In addition to comma-delineated lists, you can also format your list as one keyword per line.

See Also...

— Project Search (section 11.1): although the panel has a useful button for invoking a project search, it might be a good idea to know how that feature works as well, so you can fine-tune results if needed.

— Keywords Pane (subsection 13.5.3): where you will most often interact with keywords is in the inspector pane, assigning them to documents directly.

10.4.3 Exporting and Printing Metadata

You need never fear of having your valuable tagging and other markings lost within the Scrivener project. All forms of metadata can be exported along with the content itself, and most forms of metadata can be printed as well:

— With File ▸ Export ▸ Files... metadata can be exported as sidecar .txt files along with the content (section 25.2).

— Using File ▸ Export ▸ Outliner Contents as csv... every single type of metadata in the software can be exported to spreadsheet format.

— When printing documents from the editor, there are numerous options available for including metadata (chapter 26).

— And of course, you can include the metadata for each section of your draft when compiling, by setting up Section Layouts designed for that purpose (section 24.2.1).

  Additionally, Scrivener comes with placeholder codes that can be used to print metadata when compiling. You will find a list of these placeholders in the Help ▸ List of All Placeholders... menu command. In conjunction with the Section Layout feature, the possibility for creating your own template outputs for metadata are endless.

Return to chapter ⤴

Searching and Replacing

11

In This Section...

Being a tool design to bring research, writing materials and the work itself all together into one interface, methods for searching through the haystack and finding your needle of the moment are paramount, and Scrivener recognises that with filtering and searching tools at every level of the program, from the binder on down to finding specific images within your text.

The first two methods of searching will cover most basic needs for searching within a project:

— Project Search (section 11.1) is a binder tool that can scour the entire project's text content and return a list of matching items in the sidebar. These search results can be used as the basis for collections, either as a selection for a new standard collection, or to save the search criteria as a saved search collection. Holding search results in the sidebar also means you get to use all of the document management tools you are familiar with by now.

— Document Find and Replace (section 11.2) will help you locate instances of text within an editor, in a familiar step by step fashion through the text, or even within smaller bits of text, like the synopsis area in the inspector. This tool is also capable of replacing text.

The remainder of the tools provide further coverage and utility into the project and its text, as well as convenience:

— Project Replace (section 11.3) is there for when you truly want to change every single instance of a phrase to something else throughout the entire project. It can be tuned to only impact certain areas of the project (such as Titles alone) or even to be so thorough as to revise older versions of the text.

— Filter Outliner & Corkboard Views (section 11.4) for a quick way to find that one card you're looking for in a folder, or perhaps to only view those items marked as "first draft" in your outliner.

— The Quick Search Tool (section 11.5) sits in the main application toolbar and serves as a quick way to get to a specific document in your editor. Simply type in the phrase from the document you're looking for, or part of its name, hit Return and off you go.

— Finally, the Find by Formatting Tool (section 11.6) works a lot like regular text Find does in your editors, using a step by step method, only it is designed to find formatting. Search by styles, inline annotations, comments,

footnotes, images, tables, links, colours & highlights and more. Unlike regular Find, this tool will by default jump from one text file to the next in the binder, in the direction you are searching (next or previous).

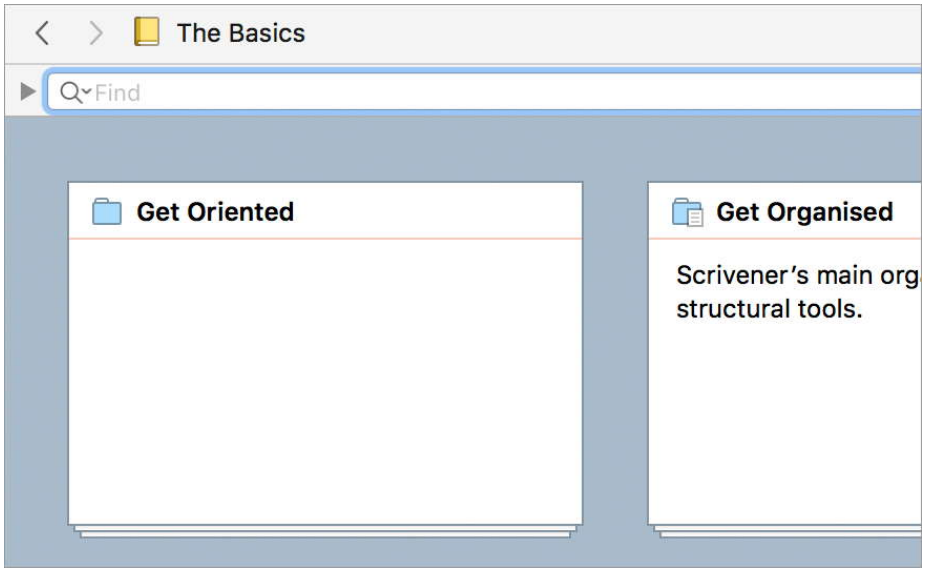


Figure 11.1: Filtering the corkboard: the magnifying glass icon in the search field grants extra searching options.

If a search tool has a small magnifying glass icon with a downward facing arrow to the left of the area where you type in text, options will be placed there to modify the search. In the case of Figure 11.1, we can click on this icon to broaden the search to include all text and notes (rather than just the stuff we can see on the index cards), or gain access to the regular expressions (section 11.7) search syntax.

11.1 Project Search

With the Project Search tool, you can quickly scour the project and get a list of every matching document in the sidebar. This can be as simple as typing in the word "hello" and finding every item that has that word in it (and by that we mean nearly everything about a document that consists of text, from assigned keywords, labels, the title to its synopsis and so on). There are a bounty of options for putting together some fairly surgical searches too, should you need them.

11.1.1 The Basics

Unlike the standard text find tool, which steps through a document one match at a time in one direction or another, project search returns everything at once in a list in the left sidebar, temporarily replacing the binder (don't worry, you can get back!). You can think of it as a way of "filtering" the binder so that it only

shows those items that match the search. Searching in this fashion is more like searching the web for hits, or your email inbox. As with those tools, once you get into a specific page or email, you can use regular text find to step through matching text to find the specific phrase you are looking for.

To indicate that you are no longer looking at the binder, the background colour of the sidebar will change to purple and some additional controls will be added to the top of the sidebar. When you are ready to dismiss the search and return to the binder, click the small × button in the sidebar heading, marked (a) in Figure 10.4, or simply tap Esc while in the search field to dismiss the search results (and tap Esc again to dismiss the search field itself).

Can I easily get back to a search list?

Yes! Search results are stored into a special collection that cannot be removed, and always stores the last search you used, even after closing and reopening the project. If the list of collections is hidden, the easy way to get back to your search list will be with the Navigate ▸ Collections ▸ Search Results menu command. Otherwise, it will be easiest to simply click on the "Search Results" collection tab in the list. For more about navigating among and using collections in general, read Using Collections (section 10.2).

Starting a New Search

There are two ways to start a new project search:

1. Click the "Search" button in the toolbar to reveal the search field (Figure 11.2).
2. Use the command, Edit ▸ Find ▸ Search in Project... (⇧⌘F), to open the field and place the cursor there in one move.

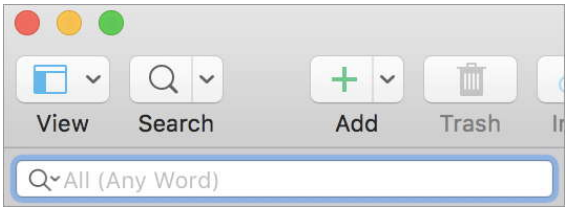


Figure 11.2: The default project search field, with the "Search" toolbar button above.

Now all you have to do is start typing, and as you type matching documents will be returned to you in the list below the search field. When nothing is entered in the search field, greyed out text will inform you of the current search mode. By default this is "All (Any Word)", meaning all searchable elements in the project

will be examined with the "any word" operator—any word you type in will result in a match if the document contains it somewhere.

The search scope, data type, and operator mode can be adjusted via the project search options menu. Click on the magnifying glass to the left of the typing area in the search field (Figure 11.2) to review the available options.[1]

Finding matches in the editor

In addition to highlighting all matched text for easy visual recognition, for your convenience, the text you searched for will be automatically loaded into the text find tool for you. This means you can immediately use the keyboard shortcut to find the next search result within the document (⌘G). Since searching the project can examine many different types of elements, the search result itself may not be in the primary editor—it might be a label or a keyword for example.

Browsing and Sorting Results

While typing, you can flip through search results with the ↑ and ↓ keys on your keyboard, loading the result into the currently active editor. Otherwise, you can use a search result list just like you would use the binder normally. Refer to the section on Selecting Items (subsection 6.3.2) for tips on viewing multiple items.

Figure 11.3: Use the "Previous" and "Next" document buttons to flip between search results.

While working in the editor, the you can jump from one search result to the next using the "Previous" and "Next" document buttons on the right-hand side of the editor header bar (Figure 11.3), or use the associated keyboard shortcuts: ⌥⌘↑ and ⌥⌘↓.

The list of results can also be sorted, using the control marked (c) in Figure 10.4:

— The **Binder Order** is the default, merely listing items in the order they are found within the binder.

— **Sort by Title**: sorts by the visible title in the list, including any automatically generated titles if the items do not have specific names given to them.

[1] If for some reason the search field is visible but you haven't used it yet, the standard but somewhat odd macOS behaviour of disabling the control will be in effect. If you want to go directly to the magnifying glass menu, click where the magnifying glass *will be* once the field is active. This will activate the field and bring up the options menu in one click.

When using this sort mode, the **A-Z** option enables. Disable it from this menu to search in descending order instead.

— **Sort by Date**: sorts the list by Created Date in ascending order. When using this sort mode, the optional behaviour becomes **Newest First**, and is disabled by default.

Refreshing Results

As you type, the contents of the search result list will refresh dynamically, narrowing down the result efficiently. However once you've stopped typing, actions taken elsewhere in the project window that might impact search results will not update the list. This allows you to work off the list without it constantly changing. If you do want to refresh the results, simply click into the search field and hit the Return key.

For cases where you've loaded search results into an editor (as above), you can refresh the list by clicking the button to load the list into the editor again, or simply navigate away from and then back to the list with the history buttons.

Project search finds nothing, or not everything it should.

This feature has a number of options, some of which are telegraphed as grey placeholder text when the search field is empty. But if it isn't empty because you've typed in some text to search for, the reasons for why there aren't as many results as you'd expect can be mysterious. Always be sure to double-check search settings, as described in the following sections, by clicking on the magnifying glass button to the left of the text field itself. You can also reset search options to default from this menu (section 11.1.2).

Taking Results to the Editor

In some cases you may want a little more space to work with your search results, or access to some of the powerful capabilities afforded by the outliner and corkboard view modes. At any time, click the ↪ button marked (b) in Figure 10.4 to load the contents of the sidebar list into the active editor.

This is also a great way to search *again*, if the initial search result was too broad and you want to search by a second factor. With the focus in the editor, press ⌘F to bring up the outliner & corkboard search tool (section 11.4).

11.1.2 Search Settings

All project search settings are accessed via a menu interface. Bring it up by clicking on the magnifying glass to the left of the search field.

Search In

The initial section of the options menu is for selecting the type of element that project search should look within. Select multiple elements by holding down the Option key and clicking on a type. Option clicking can also be used to turn off an extra field without disturbing the total selection. To return to using a single element, either select "All", or select any field without using the Option key.

- **All**: the default search mode. Every element of an item containing searchable text will be queried for matches.
- **Title**: only the titles of binder items will be searched for.
- **Text**: the text contents of files and folders will be queried. This does not include notes and synopsis.
- **Notes**: an item's inspector notes will be searched.
- **Synopsis**: the synopsis field for each document will be searched. This is anything that has been typed into the text area of an index card or in the synopsis portion of the outliner.
- **Keywords**: matches found within an item's keyword list. You can also perform keyword searches quickly by using project keywords panel (section 10.4.2).
- **Label**: the text (not colour) of the label metadata will be searched. Note the name of this metadata field can be changed per project. Available labels and status for the project can be referenced in the Project Settings: Label & Status List panes (section C.3).
- **Status**: the text of the status metadata will be searched. Note the name of this metadata field can be changed per project.
- **Section Type**: documents can be searched for by their type. This makes it easy to hunt down all items flagged as being a "Chapter" for instance, or a "Figure". A project will list its available section types in the Project Settings: Section Types pane (section C.2).

Following the standard list above, any custom metadata fields you've added to the project will be listed in a separate section. Select from these to search within that field specifically. How searching works within the field is determined by its type, with "text" type metadata fields working identically with all other text searches:

Checkbox Search by "yes", "true" or "1" to find all items that have been checked. Searching for "no", "false" or "0" will return those items that have not been checked.

List These work just like the status and label elements do, using regular text searches to find which items match the text you've typed in.

Date Search by any part of a date as text. For example "04" would return items set to April, but it would also find items set to the 4th day of any month. You can type in more of the date to be more precise, such as "2017–04" to weed out those using the 4th day.

In addition to matching dates by text, you can use our relative date matching syntax as well. Refer to Searching by modification and creation date (section 11.1.3).

Operator

Select the method by which your text will be used by the search engine:

- **Any Word**: the default search method. Queried documents must contain at least one of the words typed into the search field. Analogous to logical OR.
- **All Words**: every word entered into the search field must be present. Documents which only match some of the words will not be returned. Words can be entered in any order. Analogous to logical AND. You can also enter double-quoted phrases mixed in with single words, working in the same manner as **Exact Phrase,** below.
- **Exact Phrase**: what you type into the field will be queried precisely as it is typed in. "The book" will only match documents that have the phrase "the book" as written, not documents that just have the word "book" or the phrase "book the". It will also return documents that contain "the books". For exclusive matching, use **Whole Word**.
- **Whole Word**: unlike any of the above search methods, the term supplied will only match whole words. A search for "Jo" will only return documents with that word, not documents that also contain "Jocelyne".
- **RegEx**: enables the powerful Regular Expression search syntax (section 11.7).

Options

Provides a few extra options, as well as setting scope limiters. Scopes instruct the search engine to only scan select parts of your binder.

- **Search Draft Only**: will only look in the "Draft" folder of the binder. Note that if the name of draft folder has been changed in the project, the title of this option will reflect that name change.
- **Search Binder Selection Only**: preselect items in the binder and then perform the search against those items only. This selection is explicit, not implicit. Selecting a folder will not include all of its children in the search query as well. This option is not used by Saved Search Collections.

— **Exclude Trash Documents**: ordinarily, items located in the trash will be discovered by search results, represented by faded icons in the search results list. If you'd rather not see them at all, use this option to disable Trash folder searching. This option is not used by Saved Search Collections.

— **Search "Included" Documents**: include documents that have been marked as "Include in Compile" in their metadata. This option can be combined with the below, and one must always be selected (otherwise you wouldn't find anything!)

— **Search "Excluded" Documents**: include documents that have not been marked as "Include in Compile" in their metadata. This option can be combined with the above.

— **Case Sensitive**: by default, the search engine ignores letter case. If you need to search for proper nouns and are getting a lot of false positives, this option can help.

Resetting Search Options

To reset all search settings to a few simple default settings, select the "Reset Search Options" command in the magnifying glass menu. This will change the settings to the following settings:

— **Search in**: All

— **Operator**: Any Word

— **Options:** all disabled, save for the two toggles to search for both "included" and "excluded" documents.

11.1.3 Special Search Terms

Beyond simply typing in the text you want to find, there are some useful tricks you can use to narrow down your search further, or even mix certain types of search together.

Mixing exact phrases with all words If you want to find every document containing a word as well as a particular phrase, put the phrase in double-quotes while leaving the search operator set to "All Words". The term `Bob "black car"` would locate all documents referring to a "black car" that also mention Bob. The quotes cause the phrase to be treated as a word.

Finding things that don't have a word Sometimes the most efficient way to make a long list of search results shorter is to omit the most common word that you aren't looking for. Place a hyphen directly in front of such a word to remove it from the list of possibilities. A simple example would be to search the "Text" for `Lydia -Dovahkiin`, to find all documents mentioning Lydia that do *not* also refer to a certain individual named Dovahkiin.

This technique can be used with the Any Word and All Word operators. It can also be combined with quoted phrases. For example, `Lydia -"Needs Proofing"` could still include documents that contain both "Lydia" and "needs" or "proofing", but never when those two words form a phrase.

Finding everything To search for everything, type in a single asterisk (*) into the search bar. It can sometimes useful to build a flat list of everything in the entire project, usually in conjunction with loading the search result into an editor, where you can do such things as sorting by label to view your binder clustered by label assignment, or by Modified Date to review recently changed files.

Finding everything by type The above method of using an asterisk can be combined with all available search options. For example: you can find all items in the draft folder alone, or all items that have had a label (any label at all) assigned to them, or with the "Invert Results" option, you can find all item's that *don't* have a label assigned yet.

Searching by modification and creation date

A common desire is to locate items that have been modified recently, or to find files that were created within a certain window of time. Scrivener supports these operations and many more besides, using its special date search syntax.

When suffixing a search term with either of the keywords `mdate:` or `cdate:`, project search will combine the current "Search In" settings to also search for Modified Date or Created Date respectively. When used alone, they will simply return a list of all documents based upon that date request. Thus a search for 'cdate:2017' will return all documents created in the year 2017.

If you'd rather not have to remember these search terms, you can also simply right-click on the magnifying glass icon and click on "Search Modified Dates" or "Search Created Dates" to insert the term for you. Doing so will replace whatever you've already typed in.

Combining Search Terms

Most of the techniques we've discussed can be combined together to create more specific and complicated search criteria. Here are some examples:

— If you are searching in the "Notes" field for the word "ToDo", but only want to find those documents that haven't been modified recently, you would use the term, `ToDo mdate:<6m`, which would find every document with "ToDo" in the inspector notes that hasn't been modified in the past six months.

— You can search for all documents with the "Needs Formatting" keyword that has *not* been marked with a status of "Rough Draft", but setting **Search**

In... to both "Keywords" and "Status", the **Operator** to either "Any Words" or "All Words", and finally typing in `"needs formatting" -"rough draft"`.

— 'First Draft mdate:7d', with "Search in: Status" set, will return a list of all documents marked as "First Draft", modified within the past week.

— Let's say you have a custom metadata field called "Published" which is a simple checkbox. When searching by this field, `yes cdate:2014` returns all documents marked as published that were created within the year 2014.

Refer to Table 11.1 for a complete list of available searches.

Table 11.1: Date Search Syntax

Search Term	Description
	By a Given Date
YYYY-MM-DD or YYYY/MM/DD	Find everything from a specific day, such as 2017–04–08, which will find everything on the 8th of April, 2017.
YYYY	All edits found within the specified calendar year.
	By a Relative Date
#d	Find everything written since the number of days specified. Eg. "7d" would find everything edited within the last week.
#m	As above, but using months. Eg. "6m" will find everything edited in the past half year, or six months.
#y	As above, but using years.
	Searching Before and After
>	Expands any of the above forms to also find everything since the given date. `>2014` will provide a list of all edits within the year 2014 or later, while `2014` by itself just returns edits from within that year. Those that work that way inherently do not require the bracket, thus `>7d` is the same as `7d` by itself.
<	As above, but in this case edits found on or *before* the given date. Thus `<2y` will return everything older than two years.

11.1.4 Using a Saved Search Collection for Further Searching

With all of these options available, a good method of storing your search settings for future use is with a saved search collection (subsection 10.2.4). Whenever you click on a saved search tab (or use any method at all to load a saved search into the binder sidebar), all of the settings that were used to establish that list will be loaded into the project search settings for you, meaning the collection tab could be less about the list of documents it generates when you click on it, and more about the diverse options it sets when doing so.

1. With the collection tabs open, click on the search collection containing the type of search you want to repeat with a different term. Or use the Navigate ▸ Collections ▸ submenu to load the list directly into the sidebar.
2. Use your preferred method to invoke project search and type in a new term.
3. The result will be to create a new Project Search list using all of the settings stored in the saved search.

   You could update the search collection (section 10.2.4) if you intended to use this variation more frequently for a while, or even create a new search result collection entirely.

An applied example of where this might be useful is if you've set up a few different options to look for a character by name, looking only within the manuscript, to only consider whole words, be case sensitive and ignore all documents excluded from compilation. The saved search result will remember the name of the character you typed in at the time it was saved, but you can easily use that search as a springboard for hunting down *any* character by name, by bringing up the project search field and typing in a different name.

11.1.5 Save Search As Collection...

With so many different options you might be looking for a way to save "smart folders" or searches. The command at the bottom of the options menu will save your current search settings into a search result collection (subsection 10.2.4). As with search results, these collection tabs will build a list of results whenever you load them.

Return to chapter ⤴

11.2 Document Find and Replace

As with many programs that work with text, you are provided with a standard find and replace window which can be called up while the keyboard focus is in

any text area with Edit ▸ Find ▸ Find… (⌘F). The find panel operates only in the current context (which in the case of the main editors when using Scrivenings mode, might indeed entail multiple documents, but it's still confined within the text of those documents). To replace text throughout the entire project, use Project Replace (section 11.3); to search for text in all documents, use Project Search (section 11.1).

Finding and Replacing Text in Footnotes and Comments

Find and replace will function on any comments or footnotes in the inspector pane (section 18.3), even if the keyboard focus is in the main editor. Replace will always work, even if the inspector is closed, but for Find to highlight matches, the inspector must already be open.

While the Find panel is open, you can continue editing in the main project window. You may also restrict matches to only those which have the same letter case, are look for matches which only fall within the start of a phrase, end of a phrase, be a whole word, or the default, "contains" which matches even parts of words.

The buttons along the right of the window will take action on the settings you apply to it: The Next and Previous buttons will jump from one result to the next, using the current selection or cursor position as a starting point. When the last match in the specified direction is discovered, the tool will wrap around to the end or beginning of the document, respectively.

There are three replacement options:

— Replace: replaces the currently selected text with the contents of the "Replace" field. This works on any selection, but will most often be used after clicking the Next or Previous buttons.

— Replace & Find: equivalent to clicking the Replace button followed by the Next button.

— Replace All: replaces all matches from the "Find" field in the current text with the contents of the "Replace" field with no further interaction. The total number of replacements that were made by the last use of this button will be printed in the lower-right corner of the window.

To close the tool, you can either click the icon in the title bar to close window, tap the Esc key, or use Return to find the next match and close the window.

11.2.1 Find and Replace Options

Replace All Scope These two options determine the scope of how much text will be impacted when clicking the Replace All button:

— **Entire Document**: all of the text in the current view (which may be multiple documents when using Scrivenings mode) will be searched and replaced.

— **Selected Text**: only the selected text will be searched and replaced. Since this command will remove the selection, it means you will need to apply a new selection if you intend to continue searching and replacing this way.

Find Options These options impact how the text within the "Find" field is treated. They are used for both regular searching and replacing:

— The mode dropdown at the top contains the following options:

— **Contains**: the tool will find any sequence of text that matches what has been typed into the "Find" field, even if it is found within a word.

— **Starts with**: text will only be found if a word begins with the supplied text.

— **Whole word**: only words that match the text from start to finish will be matched. For example, "sam" will only find that word, not "Samantha".

— **Ends with**: text will only be found if the word ends with the supplied text.

— **Regular Expression (RegEx)**: enables the regular expression search syntax for *both* "Find" and "Replace" fields. When replacing, refer to a stored value from the initial search pattern using Perl-style ($1, $2…) syntax for doing so. The find tool cannot work with whitespace symbols, such as `\n` and `\t` in the replacement field. Use native whitespace characters instead.

— **Ignore Case**: the letter case typed into the "Find" field will be ignored. Thus "sam" will not find "Sam".

— **Ignore Diacritics**: accents on characters will be ignored. Thus "naïve" will also find "naive". This option is not available to the stricter regular expression mode is enabled.

11.2.2 Searching with the Keyboard

In addition to the buttons available in the panel, there are some keyboard shortcuts you can learn which can reduce the reliance upon clicking within the panel to carry out searches. These commands are also available in the Edit ▸ Find ▸ submenu:

— Return: nearly synonymous with clicking the "Next" button, with one important difference, the Find window will be automatically dismissed after

you press Return. The search term will be saved however, allowing you to continue using the two following shortcuts without any interface in your way

— ⌘G: finds the next available match; will wrap around to the top of the document if there are no more matches available below the current point. This command works even if the Find panel is closed, and it will operate off of the last active editor split (even if that split is not focussed, though searching will move your keyboard focus to the editor)

— ⇧⌘G: finds the previous match; as above, but will wrap around to the bottom of the document. As with forward searching, this command uses the last active split and will focus the editor as necessary.

— ⌘E used to load the currently selected text into the "Find" field without opening the panel. Using this method you can easily find further occurrence of existing text without even opening the Find panel, when combined with the next and previous match shortcuts, above.

Return to chapter ⤴

11.3 Project Replace

The menu command, Edit ▸ Find ▸ Project Replace… provides the utility of replacing text throughout the entire project. A progress bar at the bottom of the sheet shows you the progress of the replacement operation—bear in mind that it could take a little while on large projects.

There's No Going Back From Here!

Given that this process must go through all of the internal files in your project, opening and closing them as it goes, Project Replace cannot be undone (and you will be warned when you try)! You can use the Swap button to exchange the search term for the replacement text and try to inverse the replacement operation, but in some cases might have unexpected results. The best course of action is to take a back up prior to using this tool (the File ▸ Back Up ▸ Back Up To… command is particularly handy for cases like these). A simple mistake can render your entire draft illegible, or even worse, produce subtle flaws that are difficult to find without reading the entire work.

To include carriage returns or tabs in your search or replacement field, hold down the Option key and then press Return or Tab, and ⌥⌘Return for line feeds. For your convenience, invisible characters will be printed using visible symbols.

Replace Here is where you will type in the text used to look for matches. This can be pre-filled using the Edit ▸ Find ▸ Use Selection for Find menu command (⌘E).

With The text provided in the first box will be replaced with the text provided here. This can be empty if you wish, having the effect of removing the phrase from the project wherever it occurs.

Swap Click this button to swap the contents of the "with" field for the "replace" field. Maybe you didn't want to change the name of your antagonist after all!

Search mode The drop-down menu below the swap button, and the two check-boxes following it, are where you will set how the text in the **Replace** field should be evaluated. The rules here are identical to those documented previously (subsection 11.2.1). The **Whole Word** option in particular can be useful here; without it, searching for the character name "Sam" and replacing it with "Joseph" could end you up with such (bleakly) amusing concoctions as "Josephe" in place of every "same".

Scope Set whether the entire project should be modified, or only those items that you have previously selected in the binder, corkboard or outliner.

Affect Select which components of the project this command should modify.

- **Titles**: the titles of the documents as they appear in the binder.
- **Text**: standard text editor contents (including any linked footnotes and comments from within that text).
- **Notes**: any inspector note fields that are attached to documents.
- **Synopses**: the text content of each index card.
- **Custom Metadata**: text type fields will have their text examined and changed.
- **Snapshots**: archived snapshots will be changed (take special care with this one, as snapshots are typically an internal backup mechanism) as well as any linked footnotes and comments originating from those snapshots.

Return to chapter ⤴

11.4 Filter Outliner & Corkboard Views

The corkboard and outliner views are capable of simple filtering, making it a cinch to do such things as narrow down a project search with additional constraints, quickly find that one text file you know is somewhere in a list of a hun-

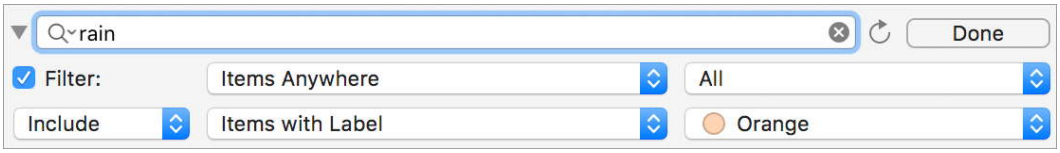


Figure 11.4: Outliner and corkboard views can be filtered by a variety of criteria.

dred cards in a folder, or only showing those rows in the outliner for items that have been labelled a certain colour, to name a few.

A filtered view has certain limitations owing to how it filters items directly in the same view. When a filter is currently applied to the view, the items will be presented in a flat list and you will be unable to reorder or create new items within that list. In this fashion it is similar to how multiple selections (section 6.4) work. Similarly, label view and freeform corkboards cannot be filtered directly—the result of filtering these views will be to revert to a flat list so long as the filter is applied.

Filter settings will be remembered per each split for the current session. All of the settings that you applied the last time the filter was used will be reapplied automatically when calling up the filter panel. This can be useful when navigating to search results and then using history to return to the view—you can press ⌘F to return to the filtered view after ⌘[to go back in history.

11.4.1 Starting a New Filter

Start a new filter on the current corkboard or outliner by using the Edit ▸ Find ▸ Filter... menu command (⌘F), type in the text you're looking for, and Scrivener will start scouring through all visible text in the view and present a list of only those items that contain matching text.

The results, in other words, will be based on the information shown, according to the settings you use within the view. If you add a column to the outliner, or show status stamps on the corkboard, then any text found within those additional fields will be evaluated along with the rest. This includes nonverbal cues, such as label and keyword colours.

Magnifying Glass Menu There are two additional options that impact what will be found based on what you type in the search field. Access them by clicking the magnifying glass button, to the direct left of where you type in the search text.

- **Include Text and Notes in Search**: to broaden what is found, beyond what you can see in the view you are filtering, this option will include the main text content (available for folders and files only) and document notes (subsection 13.3.2).
- **Use Regular Expressions (RegEx)**: enables regular expression syntax support.

Additional Filtering Options To the very left of the panel you will find a disclosure arrow that when clicked will reveal additional options for searching by compile status and metadata. Figure 11.4 displays the filter panel with this additional section opened. You will need the panel expanded, and the **Filter** checkbox enabled, to access the following two sections.

11.4.2 Filtering by Compile Status

To get started, click the disclosure arrow on the far right-hand side of the filter panel. The first two dropdown menus along the top contain filters for whether a document is in the draft folder, and whether it is set to be compiled:

Draft folder status The first dropdown is for filtering whether or not the items in the view are currently located within the draft folder. This will most often be useful when filtering collections or search results, where the items in the corkboard or outliner may have come from anywhere in the binder:

- **Items Anywhere**: the default setting is to not filter items by whether or not they are in the draft.
- **Draft Items Only**: only those items currently found within the Draft folder will be matched.
- **Non-Draft Items Only**: only those items found *outside* of the Draft folder will be matched.

Included or Excluded from Compile Independent of whether an item is in the draft folder, you can filter matches by whether or not their "Include in Compile" checkbox is enabled in the general metadata area of the inspector pane (subsection 13.5.1)

- **All**: the default setting is to not filter items by whether or not they are included in compile.
- **Included**: only those items with the "Include in Compile" checkbox ticked will be matched.
- **Excluded**: only items with "Include in Compile" disabled will be matched.

11.4.3 Filtering by Metadata

Although you can add the appropriate column in outliner view to search within it, it may sometimes be desirable to make certain you are only finding results from within that particular metadata field—or on the corkboard to be able to search for fields that cannot be displayed within it. To get started, click the disclosure arrow on the far right-hand side of the filter panel, and look to the bottom row of dropdown menus provided.

Enable Metadata Filtering In the first dropdown menu you will want to change the selection from the default "Any" to either one of "Include" or "Exclude". The former will cause the filter to only return results that match the metadata criteria you set up in the next two dropdown boxes. The latter option will only return results that do *not* match what you specify.

Metadata Field Next you will need to specify which metadata field the results should be filtered by, in the second dropdown menu:

- **Items with Label**: filter by assigned labels. The name of this entry may change depending upon project settings for how labels are referred to.
- **Items with Status**: filter by assigned status. The name of this entry may change depending upon project settings for how status are referred to.
- **Items by Section Type**: filter by the type of section a document is. You could for example search for all "Chapter" items in the corkboard, or create a list of all "Glossary Entries". What choices you have here depend on how you've set up your project (or how the project was set up if you're using an unmodified template).
- The remainder of the menu will list any custom metadata fields that have been assigned to the project.

Metadata Value Most of the metadata types available to filter by are list-type fields, meaning there will be a set number of possibilities to choose from in the third dropdown. For example, if you choose to filter by label, you will then select which label to filter by.

The one exception will be any custom checkbox fields. Again you will have a list to choose from, but the choice will be very simple: yes, or no.

11.4.4 Resetting and Refreshing Results

To clear the current search term, either press the Esc key on your keyboard, or click the × button on the right-hand side of the typing area. If the text field is already empty then pressing Esc an additional time will close the filter panel.

If items themselves change in such a way that their presence in the filtered list would change, either to be added or excluded, the result will not automatically refresh. You will need to do so yourself by clicking the ↻ button, to the right of the text entry area.

11.4.5 Closing the Filter Bar

When you are finished filtering the view, click the Done button on the far right of the filter bar to close the panel and restore the view. From the keyboard you

can use the Esc key to clear the text from the current filter, and when the filter text is empty, Esc will also close the panel.

Filters will also be closed automatically whenever navigating away from the filtered view, or when closing and opening the project.

11.4.6 Working with Results in Extended Usage

Given how filters will be dismissed when opening search results into the editor being filtered, you may wish to sometimes freeze the list of results so that you can hop back and forth between results using the history buttons without losing your place. There are a few different approaches you can take:

— Naturally there is the other split. We've assumed it is being used for something else at the moment however, but this basic approach of selecting a search result and hitting the ⇧⌘O shortcut deserves mention.

— Copyholders, particularly with the ability to auto-load the things you click on into the copyholder (subsection 12.2.5) can be a great way of browsing through and editing a filtered view:

1. Select the search result you wish to view and use the Navigate ▸ Open ▸ in Copyholder menu command (also available from the right-click contextual menu), or drag it to the editor header bar with the Option key held down.
2. Use the Navigate ▸ Corkboard|Outliner Selection Affects ▸ Copyholder menu toggle, or click the "auto-load" button in the footer bar (Figure 12.3) so that it becomes blue and the arrow is pointing into the box—this indicates what what you click on in the editor list will now be loaded into the copyholder automatically. If you own a keyboard with a Touch Bar, you'll find a button on your keyboard that resembles this icon. Use it to toggle the mode on without reaching for the mouse.
3. If you need to get over to the copyholder view to edit, the ^⌥⌘D shortcut can come in handy.

— But if you're already using the copyholder for something as well as the other split, or if you just want the convenience of doing everything in one view, there is one last trick: opening all of the search results into the current editor:

1. Select all of the search results you wish to work with (⌘A might be easiest here).
2. Use the Spacebar or ⌥⌘O shortcut, or drag the items to the editor header bar if you prefer the mouse. This will load the selected items into the editor *as a multiple selection* rather than a filtered result of a

larger group of items. Now as you navigate into these items and then hit the Back button to return, you will be returning to this selection as your base of operation.

— Lastly, for a more permanent solution, using collections (section 10.2) to store a list of items for future use:

1. Select all of the search results you wish to work with (⌘A might be easiest here).
2. Use the Documents ▸ Add to Collection ▸ New Collection... menu command, or select an existing collection.

Return to chapter ⤴

11.5 Quick Search Tool

Sometimes you just want to jot down a quick idea or note to yourself on a particular item within your project, but you don't want to hunt around for it in the binder (maybe it isn't even visible), or maybe you are already using project search and don't want to change the settings for just one thing. Quick Search is a new capability in Scrivener that addresses the desire to quickly locate and navigate to items based on their titles, synopsis or text. It is located front and centre in the application toolbar (Figure 11.5).

Figure 11.5: The Quick Search tool resembles the URL bar in Safari.

When not in use, the name of the document you are currently looking at, or working within a larger Scrivenings session on, will be printed in the tool for your reference.

11.5.1 Starting a New Quick Search

You can start using it much like you would any search tool:

— Click into the bar to place your cursor there.

— Or use the ^⌥G shortcut to move the cursor into it.

Once you start typing, search results will start appearing, narrowing down in scope the more you type. The search method used will always be "Exact Phrase", and it will prioritise results that match what you have typed in completely, as well as titles that begin with what you've typed so far. These two exceptions aside, search results will be displayed in the order they appear within the binder, top to bottom.

Search results can be broken up into as many as three different sections:

1. Titles: only the top ten documents that contain text you have typed in will be printed here. Full titles will be brought to the top even if they are found much further down in the binder than other matches. For example if you type in "Research", you will find it near the top even if there are hundreds of files above it that use that word in their names (unless of course you have hundreds of documents named "Research", but then this probably isn't the best tool for the job!).
2. Synopses: exact phrases found within hand-crafted synopses will appear listed below titles. This will never match adaptively generated synopses from the main text content, only those summaries you've written yourself.
3. Text: the last ten slots in the search result list will show any documents containing the phrase you have typed, with a little context around the phrase for your reference.

If even after typing in a good amount of text you cannot narrow down the list enough to make it useful or find what you are looking for, you can click the "Full Project Search" button at the very bottom to take your search settings and what you've typed so far into the binder sidebar for further refinement, as well as the ability to scan through hundreds of results more easily.

11.5.2 Using Quick Search Results

Once you have found the item you are looking for anywhere in the list, it will often be easiest to click on it, loading the item directly into the active editor. It is also possible to use the arrow keys to move the selection bar up and down through the list, pressing Return once you have the one you want to load. The selection bar automatically highlights the first result in the list. Either of these methods can be modified in the following ways:

— Add the Shift key to load the search result as a Quick Reference panel.

— Add the Option key to load the result in the other editor split, creating one if necessary. This action does not focus the new split, meaning you can look up reference material without breaking the flow of writing.

If the search result loads in a fashion where its text is displayed (as opposed to a corkboard, say), the first bit of text that matches what you searched for will be scrolled to and selected automatically. The text find tool (Edit ▸ Find ▸ Find...) will automatically be filled with your search term, meaning you can use the keyboard shortcut for walking through search results (⌘G).

Searching isn't solely about reading or editing things. Often we search with the intention of doing something with the things we find. To that end, every entry in the search result list can be dragged and dropped—and doing so will be just like dragging and dropping that item from the binder. Every possible valid drop location within the project window or in its peripheral windows is a valid target. This means you can move items, assign them to collections, open them as copyholders, add them as hyperlinks to your text, bookmark them, append their contents into the editor (with Option drag) and so on and so forth. They are, like all other icons in Scrivener, full proxies for the items they represent.

11.5.3 Project Targets and the Quick Search Tool

When you aren't using it for searching, this tool also serves as a convenient means of tracking progress toward any goals you have set toward draft or session totals. To get started with goals, hold down the Option key on your keyboard and click in the Quick Search tool itself. This will bring up the Project Targets panel (subsection 20.1.1), where you can edit your goals and change options on how those goals should be met. This can also be brought up with the Project ▸ Show Project Targets menu command (⇧⌘T).

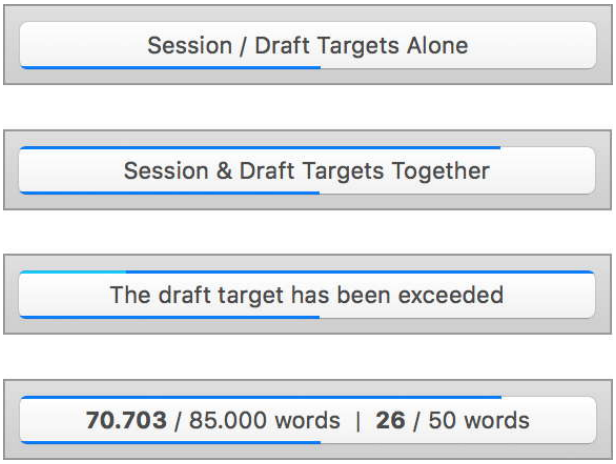


Figure 11.6: The various ways progress can be monitored via Quick Search.

The progress bars function in two different modes, depicted in the first two examples in Figure 11.6:

1. If only a draft target or session target alone has been set, the progress toward that target will be displayed as a progress bar along the bottom of the Quick Search tool.

2. If both a draft target and session target are set then the tool will display two progress bars. The draft target will be displayed along the top with the session target using the bottom track.

When you have opted to show a draft overrun, and have written in excess of your goal, a teal progress bar will creep up on the left showing how far over you are, as shown in the third example in the figure.

Finally, as shown in the fourth example, you can also mouse over the Quick Search field to get a numerical read-out of the draft and session word count, and leave the mouse there to keep that information up instead of the name of the thing you're editing.

11.5.4 Quick Search Settings

There are a few settings that adjust how project target tracking works in this tool. They are located in the Appearance: Target Progress Bars preference pane (subsection B.5.15):

— If you'd rather not be distracted by progress bars while writing, but still wish to make use of goals, you can disable feedback by disabling the **Show progress bars in Quick Search toolbar item** option.

— The default colours are designed to fit in with the rest of the system. If you would prefer they use the colours the rest of the progress bars do (in the footer bar for document targets, the Project Targets pane and in the outliner progress columns), set **Use custom colors in toolbar progress bars**.

— The colours themselves can be changed in the "Colors" tab of this preference pane—as noted, the settings will impact all progress bars in the software.

Return to chapter ⤴

11.6 Find by Formatting Tool

The "Find by Formatting" utility can be invoked from Edit ▸ Find ▸ Find by Formatting… (^⌥⌘F). It gathers together a number of powerful project-wide search tools for otherwise difficult to locate things, such as inline images by name; cross-reference links; text by style; annotations & footnotes; and so on. The anatomy of the tool is quite simple, let's take a look (Figure 11.7):

The **Find** dropdown at the very top is where you select the type of thing you are hunting down, such as pictures, footnotes or revision tracking markup (to mention a few). The remainder of this section will focus on details specific to each of the choices available in this menu. The following two controls below are universal to each search mode:

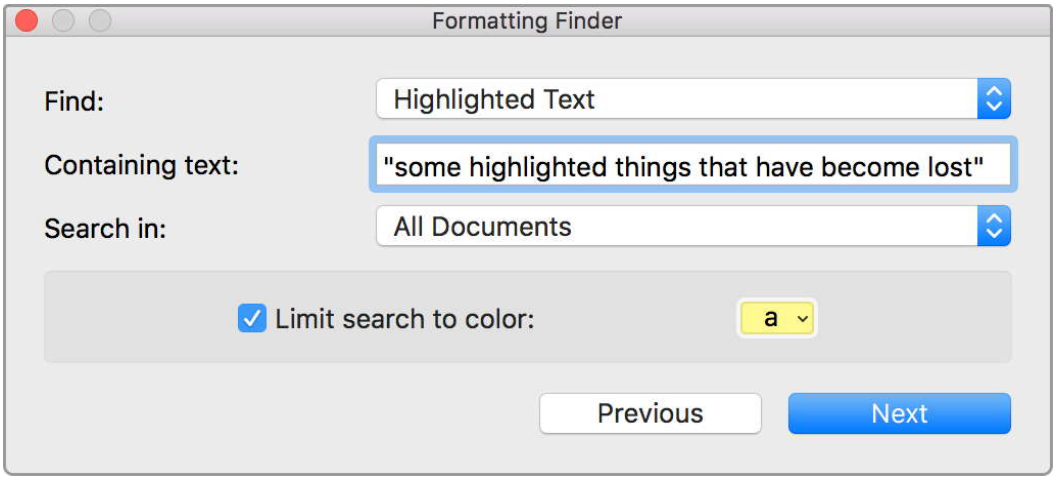


Figure 11.7: Find by Formatting: for finding stuff other than text.

— **Containing text**: narrows down the search to only return results with the specified text. In the provided example, we would only find the phrase typed into the field if it were highlighted in yellow.

The type of search used here will always be "exact phrase".

— **Search in**: there are two choices available: "All Documents" will search all text documents in the binder (even outside of the Draft), starting from the current point in the active editor. "Current Editor" will constrict the search to only the text found within the active editor. In both cases, search will automatically wrap around to the top or bottom of the document/binder once you've reached the last result in the current search direction.

Taking a While to Find Stuff?

If you have the **Search in** mode set to "All Documents" and are searching within a large project, it may take a while for Scrivener to locate results, especially if the results are located far apart. There is no search index for things like bold text, so the software has to manually trawl through the text to find matching conditions. If you know the general area where you'd like to search, such as one chapter of your book, it would be best to use Scrivenings mode to view that section of text, and set the **Search in** mode to "Current Editor".

The rest of the panel, in the shaded grey area, will change depending upon the current **Find** mode. As with the standard text find panel, it can be left floating over the project window as you click the Next and Previous buttons, so you can edit immediately after coming across a match, and then going back to search without having to call up the palette again.

Also like the standard text searching it can run in "background mode" as well. Once a search criteria has been set up, you can close the window and use the

following commands to step through results without the panel getting in the way:

— Edit ▸ Find ▸ Find Next Formatting (⇧⌥⌘G

— Edit ▸ Find ▸ Find Previous Formatting (^⌥⌘G)

11.6.1 Highlighted Text

Looks for text that has been highlighted using the highlight feature (section 18.5).

Limit search to color When the checkbox is enabled, only those highlights matching the chosen colour will be found. If this is disabled, all highlight colours will be considered a match. Note that the colour must be *precisely* the same, so stick to using basic or custom swatches or the built-in highlighter defaults when using this tool.

To select a colour, right-click on the colour chip to select from Scrivener's standard stock colours, or left click to open the colour chooser directly.

11.6.2 Comments & Footnotes

Search within any inspector-based comments or footnotes in the project. By default both will be considered for potential matches, but you can narrow this down by selecting one of Comments or Footnotes from the **Type** menu.

11.6.3 Inline Annotations and Footnotes

Searching for inline annotations gives you three colour matching options:

— *Any Color*: No limits will be made on the search results.

— *Limit Search to Color*: will only find annotations of precisely the specified colour.

— *Exclude Color from Search*: any annotations precisely matching the provided colour will be excluded from the search.

Inline footnote searching is much more simple. Since footnotes cannot have custom colours, no additional criteria is necessary.

11.6.4 Revision Colour

You will be given the choice to search for a particular revision level in the drop-down menu, matching those used by the Format ▸ Revision Mode ▸ submenu while editing. Since these colours can be changed in the Editing: Revisions preference tab, inconsistencies between stored markings and current settings can result if

collaborators are not using the same preferences, or if you change these settings midway through a project. If you intend to make use of Revisions in general it will be a good idea to establish a standard before embarking on a major editing effort.

Upgrading from Scrivener 2

By the same token, if upon upgrading an older project to Scrivener 3's format you find this tool no longer locates revision markings, that'll be because the default colours have been very slightly changed. Further instructions for rectifying this are provided in The Devil in the Details (section E.10), under "Modified Default Revisions Colours".

This feature will step through any edits that have been made while using a particular revision level, including any overstrikes that have been made.

11.6.5 Colored Text

The interface for this type of search is similar to the highlight search type. You can provide a custom colour restriction in the additional criteria section. The colour choice needs to be precise, so using custom swatches or built-in presets will generally be easiest. As with the highlight tool, right-click on the colour chip to select from Scrivener's standard stock colours, or left click to open the colour chooser directly.

11.6.6 Style

The stylesheet for the current project will be presented in the **Style Name** drop-down. If you wish to search only within block quotes, code blocks or headings, this is where you would do so. This tool will mainly be of use if you want to walk through style usage throughout the project, or hunt down specific text found within a specific style. There are additional search capabilities if you're mainly interested in selecting or finding all text assigned to a specific style. Refer to Selecting and Searching for Styles (section 15.6.4) for more information.

11.6.7 Character Format

Common text-level formatting can be searched for using this tool. Any number of options in the additional criteria can be stipulated. They work in an additive fashion, so if you have both bold and underline selected, a successful match must be *both* bold and underlined. **Keep with next** search for paragraphs that have the Format ▸ Paragraph ▸ Keep With Next marker added.

11.6.8 Links

You may search for hyperlinks of any sort using this tool. By default all links will be returned, but if you wish to narrow the search down to a particular link type, use the **Link type** dropdown to make this choice. The "Web/File Link" will in fact locate *all* links, save for internal document links. Beyond the Web, mail, ftp, protocol links to other software (or other Scrivener projects) and so forth will also be found.

When using the "Document link" link type, the Replace with Title button can be used to replace the visible text of the link with the Binder title of the item it points to. This can be useful for updating cross-references in your draft, if you know the underlying name of the item has changed.

You don't have to use the search tool to update links. This can also be accomplished with the Edit ▸ Update Links to Use Target Titles menu command at any time, even by right-clicking on a link in the editor.

11.6.9 Images

Inline images in text files will be found by this tool. This includes fully embedded graphics that have been pasted or inserted into the file, as well as images that have been linked to files on the disk, via the Insert ▸ Image Linked to File... menu command, or images linked to binder files via Insert ▸ Image Linked to Document. This will *not* find images referenced with a placeholder—given that placeholders are simple text, you can use the regular text searching tools to locate these.

In this mode, the **Containing text** field will be renamed to **Name contains text**, and the text you enter here will be used to look for the name of the image image, whether that be its filename on the disk, its name if fully embedded (double-click on an image in the editor to view or edit its name) or the binder titles of the images when linked in that fashion.

11.6.10 Tables

Tables inserted into text documents will be found with this tool. Matching text contained within any cell of the table will cause it to be found. In all cases the entire table will be selected.

11.6.11 Preserved Formatting Text

This will find text which has been marked as "Preserve Formatting", via the Format ▸ Preserve Formatting command. Since there are no optional qualities to these blocks of text, no additional criteria is required.

Return to chapter ⤴

11.7 Regular Expressions

Regular expressions are an advanced search syntax that can also be used to do complex replacements. If you know what they are, you are probably delighted to hear we support them. If you don't, we could not even begin to document this search syntax; entire books even larger than this user manual have been written on the topic! If you would like to learn how to use them, there are plenty of recipes, learning resources and tutorials on the Web, as well as those aforementioned books.

Regular expression support is provided in the following locations:

— *Project Search*: select "RegEx" from the operator section of the magnifying glass icon menu.

— *Project Replace*: using Edit ▸ Find ▸ Project Replace..., enable regular expressions by selecting "Regular Expressions (RegEx)" from the dropdown menu containing search modes.

— *Find*: the standard Edit ▸ Find ▸ Find... text searching tool supports them. Select "Regular Expressions (RegEx)" from the dropdown menu at the top of the "Find Options" section.

— *Outliner & Corkboard filtering*: when invoking Edit ▸ Find ▸ Find... from an outliner or corkboard view, click the magnifying glass and select "Use Regular Expressions (RegEx)".

— *Compile replacements*: RegEx can be used in compile-time replacements, which alter the output without changing the original text.

Scrivener uses the stock RegEx engine supplied by the Mac, which is based upon the Perl Compatible Regular Expressions (PCRE) guidelines, extended to provide UTF-8 support. Anyone who is familiar with using regular expressions in Perl will feel at home searching in Scrivener.

When used for replacing text in either the compile pane, or the document find panel, you will need to address stored variables using the dollarsign-numeral syntax, such as `$1` for the first stored variable, `$2` for the second and so forth.

Return to chapter ⤴

Project Navigation

12

In This Section...

Moving around within a large project is an important part of writing efficiently, and especially in an application like Scrivener where cutting your work up into many small pieces is second nature. It's such an important aspect of the software, we have an entire top level menu devoted to navigation (section A.7).

An average book might have anywhere between several dozen, to hundreds or even thousands of sections arranged into many levels of groups in the Draft folder alone, and that's not counting any research you may have added to the project. We will cover how flexible the project window is, both as an adaptive tool that can be shaped to your current task, and as a way of building more static workflows (like a 3-pane browser, common to mail clients). Along with the binder, there are two primary tools for making sense of all of these pieces: the outliner (section 8.3) and the corkboard (section 8.2).

In this chapter we will concentrate on how to use these tools, the binder and other navigational tools to, as we say, see the forest *or* the trees.

12.1 General Navigation

Some navigation features are common to both the corkboard and the outliner:

— Double-click on any icon (to the left of the title itself) to load that document in the current editor. This process is sometimes referred to as "drilling down" into the outline.

— Use the Spacebar or the ⌥⌘O shortcut to load the selected items in the current editor. When more than one item is selected, the effect will be to isolate them using the current view mode so you can focus on them. This latter usage is quite useful when you have a large group of items, but only wish to work with a selection, non-linear or otherwise, of those items.

— ⇧⌘O loads the selected item(s) in the opposing split, opening a new one if necessary. This command can also be used from the binder, when you wish to target the split that clicking in the binder wouldn't otherwise impact.

The following command can be used from any group view mode as well as the binder sidebar:

— Also available to the binder sidebar. Unlike most other forms of navigation, the Navigate ▸ Go To ▸ Selection command (⌘4) will always load the selected item as a single document in the active editor, even if the selection is a container. This can be useful if you like to jot down notes into a folder's text area. Since this command will disregard the usual view mode preference (such as viewing containers as a corkboard) it also has no impact on that preference. Use of it will not cause containers to load as single documents in the future.

When used in a Scrivenings session, the effect is to "isolate" the text your cursor or selection is currently within, allowing you to focus on that one section alone.

These commands can be used from within the editor in all contexts:

— Sometimes you may want to go "up" a level of hierarchy, to the enclosing group of the selection, instead of "down", as all of the above commands do. Use Navigate ▸ Go To ▸ Enclosing Group (^⌘R). Viewing the enclosing group will also select the currently viewed item. For example, if you have a folder in your corkboard called "Chapter 10", and that folder is nested beneath the Draft folder, then using this command will select the Draft folder in the editor, with "Chapter 10" selected within it.

When used in a Scrivenings session, the effect is to broaden the current text content to include a larger section of the work.

— The Navigate ▸ Go To ▸ Previous Document and Next Document commands (⌥⌘↑ & ⌥⌘↓ respectively) will walk through the order of items as listed

in the binder sidebar from the current vantage point. This form of navigation is also available as buttons on the editor header bar itself, on the right-hand side beside the split button, as up and down arrows. The contextual distinction here is key: if the sidebar is showing search results, then these buttons will walk through that list of items. If the binder is showing, the result will be to move through the binder stack one by one. If the item you are viewing in the editor is *not* listed in the sidebar then this form of navigation will be disabled as there would be no way to get from the current point to any frame of reference within the sidebar list.

Within a Scrivenings session these commands will jump from one section to the next within the session until you get to the end of the session in either direction.

12.1.1 Header Bar Drag and Drop

As mentioned before, Scrivener has extensive support for the concept of drag and drop. There are many areas where you can drag icons from and nearly just as many that you can drag *to*. The header bars on the editor splits are one such target, and they have two basic modes of usage:

1. Dropping binder items: in this case the items (if plural, it will form a multiple selection) are loaded in the editor, much as though you had selected or clicked on them in the binder. When holding down the Option key the action will be to load the item (no plurality in this case!) into the editor's copyholder (subsection 8.1.5). (You can also drag an item directly to an existing copyholder's smaller header bar.)
2. Dropping snapshots: when dragged from either the inspector or the snapshots manager the action will be to load the snapshot as a read-only copy in the main editor. By holding down the Option key while dragging, the snapshot will be compared with the current live version of its respective document. Read more about these capabilities in Using Snapshots (section 15.8).

Here are a few ideas of how this capability can be used:

— If your Navigate ▸ Binder Affects ▸ settings are such that the editor you wish to load an item in does not receive binder clicks, you can use drag and drop to manually load the document anyway.

— From the Bookmarks Tab (section 13.4) of the inspector. Although you can also right-click on a bookmark and choose to load it in either editor, some may find drag and drop a more direct approach.

If you would prefer bookmarks always load in one of the editors instead of a separate window, tune the **Open Inspector bookmarks in...** setting, in the Behaviors: Document Links preference pane (subsection B.4.2).

— From a Quick Reference header bar. If you've had a document open in a separate window but wish to "dock" it into the main project window, drag the header bar icon from the Quick Reference panel into a header bar to load it there.

Hopefully this illustrates how "binder items" in this context can refer to many different sources—not just strictly speaking the list of items in the left sidebar.

12.1.2 Go To Menu

In most cases, it will be easiest to use the binder, or the group views in the main editor, to select and navigate to various components in your project. There will be times when this is not true. If you prefer to work with the binder hidden or are using a collection, are working in composition mode, or just would rather not drill down to a particular spot solely to select an item, the Go To submenu is an handy substitute for other methods, and in some cases can even be faster than traditional navigation. The Go To menu always takes action on the activated split if relevant, even when it has been locked.

Items that have been tagged as project bookmarks (section 10.3) will be placed at the very top of this menu for your convenience. Refer to the previous section for documentation on the spatial commands, such as "Next Document" and "Enclosing Group".

The "Collections" entry in this menu will navigate the *main editor* to a collection list, rather than showing it in the sidebar as would be traditionally done. Viewing collections in the editors brings all of their organisational and visualisation prowess to bear.

There are three locations where this menu appears:

1. The main application menu: Navigate ▸ Go To.
2. From the editor header bar icon menu, as "Go To Document".
3. In composition mode, the Go To menu is available as a button in the control strip along the bottom of the screen. This form has a special behaviour when using Scrivenings mode in that it will only show items from the current editing session rather than the entire binder.

In all cases, both containers (represented as submenus in this context) and items can be selected. When a group is selected, it will be loaded with the default view mode if called from a main text editor. When used from contexts that can only display text, such as copyholders and composition mode then the text of that container will be shown (which may be empty).

Return to chapter ⤴

12.2 Controlling Sidebar and Editor Integration

With few exceptions, whatever you click on in the binder sidebar will be automatically loaded into the active editor. This can become more complicated when there is more than one editor split in use. Occasionally you'll want to load things in another split so as to not disturb your current environment. Or you may even want everything to load in the other editor as a matter of course, no matter which editor is currently active. You may also want the binder to be completely decoupled from the editors so that all changes to what you are editing becomes a manual process.

This section will go over the sorts of adjustments you can make to change how the project window works in a persistent fashion, as well as showing you a few one-off tricks you can use to make use of additional splits without a whole lot of clicking around.

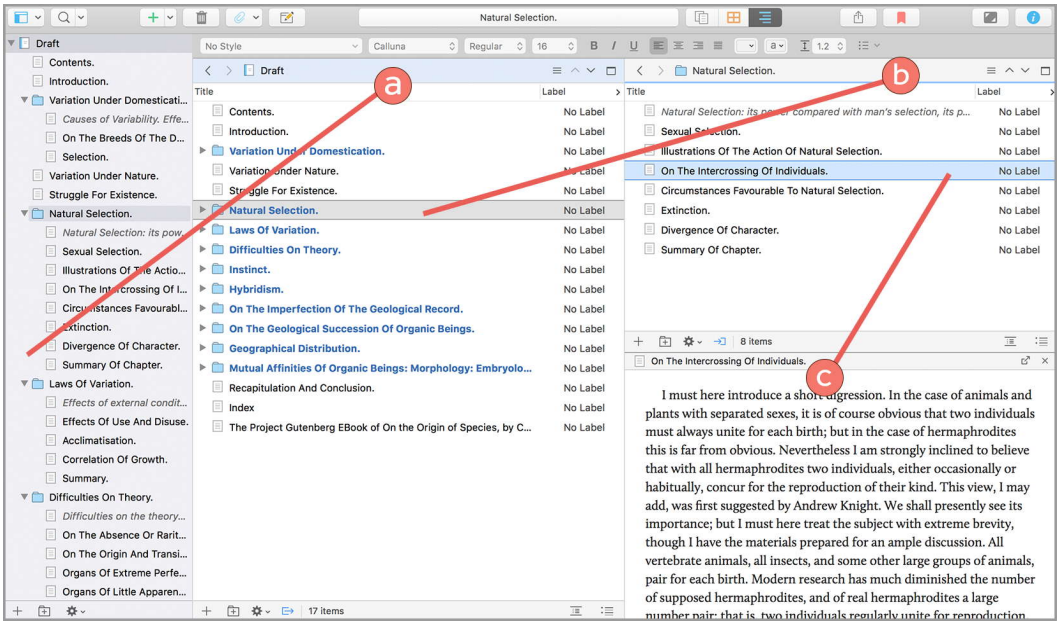


Figure 12.1: A complex demonstration synthesising the techniques in this section.

Taking a look at Figure 12.1 we can see a combination of different features being used together to create a workflow that emulates how some 3-pane software works in principle, with an additional twist that most cannot do. If you would like to play with this example yourself while following along, you will find this project in the Extras Pack (Appendix F), as "3-complex_layouts.scriv".[1]

[1] Do not be concerned if all of the files are empty save for the letter 'a', this is meant mainly to demonstrate the UI setup, not to provide a copy of the entire book!

a) The first thing to note is that the left split header bar is blue, even though the active editor is indicated as being the right split. This means we have set up the binder to only affect the *left* editor when clicking on items within it. Our right editor will never be changed by what we do in the binder sidebar. At the bottom of the left editor is a blue button indicating that whatever we click on in the left editor will automatically be loaded in the right editor—in other words it will act a little bit like the binder does.

b) As referred to before, the blue underscore in the right editor's header bar indicates it is the *active split*. It is where we are working, where our cursor keys will move the selection highlight bar and so forth. It is viewing a folder that was clicked in the left editor split. At the bottom of this split you'll find that blue button highlighted again, only this time it is indicating that whatever we click on in the right editor will be automatically loaded in the copyholder below it.

c) Finally we get to the copyholder, or some actual text in this example. The right half of the project window acts a bit like a mini-browser of its own. You can create and work with items in the upper half and then edit the text of them in the lower half.

Naturally not every project will require all of these features to be enabled at once, but now that you've seen a little of Scrivener's flexibility in what it *can* do, you might have a few ideas for how you can use bits and pieces of this example in your own work. For example maybe you just want to have the binder only impact the left half of your window and leave the right half alone so that it can remain a more static working area.

Now that we've glossed over on the strategy, let's get into all of tactics that can be used to achieve it. And by the way, if you like the look of this example, you can very quickly apply it to your own project using the built-in layout,"Dual Navigation", from the Window ▸ Layouts ▸ submenu (section 12.3.3).

See Also...

— Splitting the Editor (subsection 8.1.4): much of what we'll be talking about in this section will only be beneficial if you're working with split views.

— The Active Editor and Targeted Editor in Split Views (section 8.1.1): for more information on how the editors signal their integration with the binder.

— Selecting Items (subsection 6.3.2): the basics on how the binder works with the editors in general.

12.2.1 Locking the Editor

A tool that can save you a lot of time, and solve a lot of arbitrary "how do I do this..." type questions is knowing how to essentially *turn off* project naviga-

tion. Any editor split (even if no splits are enabled) can be locked with the Navigate ▸ Editor ▸ Lock in Place menu command (⌥⌘L)[2]. The header bar for the editor will be changed to a red colour, and whenever you take an action that would ordinarily have loaded content into that editor, the request will instead by diverted to the other editor, unless it is also locked, or the request originated from it.

> **Upgrading from Scrivener 2**
>
> In previous versions of Scrivener, locking the editor had the effect of keeping all navigation inert so long as that split was active, rather than always diverting navigation to an available unlocked split instead. If you would prefer the older behaviour, set the **When focused editor is locked in place** to "Binder selection does nothing", in the Behaviors: Navigation preference pane.

Locking the editor inhibits *external navigation* from impacting the locked editor. This means that clicks in the binder, corkboard or outliner (when Navigate ▸ Corkboard|Outliner Selection Affects ▸ Other Editor is enabled) and inspector bookmark double-clicks will be blocked. This makes it easy to keep your editing session stable while you use use other tools.

What locking will *not* do is prohibit actions taken from within the editor itself, or those we might consider intentional. This includes use of editor navigation functions (such as history), manually dragging items to the header bar, hyperlink usage or menu navigation commands. Locking is meant primarily to keep the interface from taking actions that the software would ordinarily take automatically. Locking will not inhibit intentional actions that *you* make. These actions will navigate as requested and cancel the lock state in doing so.

An exception to locked behaviour is when it makes sense to scroll your view, or select items within it, based on an external selection that otherwise would impact the editor. If when using a locked group view you click on a portion of the outline that is contained within that view, then the editor will scroll to the right spot and select that item. For example, in a large locked corkboard, you can jump to individual cards by clicking on them in the binder, or jump to sections of your text in a Scrivenings session. This behaviour leaves the lock intact, so if you later click on something else outside of the view nothing will happen.

12.2.2 Locking the Group View Mode

Which view mode to use when selecting a group of items is determined by the last view mode (section 4.2) you used. If you click on a folder and switch the view mode to Scrivenings, then later select five separate items together, they will be shown using Scrivenings as well (assuming they are items capable of having their

[2] This feature can also be accessed from the header bar contextual menu (section 8.1.1).

text edited). In most cases this works quite well as we tend to do tasks in bunches. Sometimes however, you might wish to designate a certain container as always being shown using a particular view mode, no matter what the current preferred view mode may be.

Figure 12.2: The lock icon indicates the container you are viewing has been locked to the outliner group view mode.

To set a container (defined as a folder, file group or collection) to use a particular view mode, use the Navigate ▸ Editor ▸ Lock Group View Mode menu command on the active editor you wish to freeze.[3] If the toolbar is visible, you'll see a small lock icon added to associated view button (Figure 12.2). So long as the view mode is locked, changing the view mode while viewing the container will modify which view mode it should use in the future.

Here is an example of how this feature might work:

1. A folder (A) is set to have its group view mode locked to corkboard mode.
2. Another folder (B) is clicked on, and the view mode changed to outliner.
3. When switching back to folder A, the view mode returns to corkboard.
4. Viewing folder B, or any other unlocked folder in the binder, will return to outliner mode.
5. Now if folder A is viewed again and changed to Scrivenings, folder B (and the rest of the project) will use Scrivenings mode as per normal. Additionally, this action will *also* lock the selected view mode to folder A. It will now use Scrivenings mode by preference instead of the corkboard.

There is no limit to the number of containers that can be locked to a view mode in a project. The setting is unique to that container and does not impact anything else.

Locking the Group View is a Persistent Setting

This means that if the container is duplicated that setting will be carried along to the new duplicate. For example if the setting is applied to a Document Templates (section 7.5), then each new instance created using that template will also bear that setting. However from that point on these new containers will maintain their own settings; changing one will not impact the others that came from it, or vice versa.

[3] This command is also available from the header bar contextual menu (section 8.1.1).

12.2.3 Adjusting What the Binder Affects

Earlier we spoke of adjusting whether the binder should interact with a particular split or leave it alone. Let's take a look at two different ways of doing that—one for quick one-offs and another for altering how the binder interacts with splits.

When Loading Items

At any time you can modify how selection works in the binder sidebar and other contexts, causing the action to impact the other split, by holding down the Option key when clicking. Here are the contexts and methods this blends with:

— Clicking on individual items in the binder.

— With the combiner modifier keys, add or remove from Multiple Selections (section 6.4) in the other split. ⌥⌘Click will add or remove the item you clicked on to the other split's group view (forming a group if currently only one item is showing). ⌥⇧Click adds ranges of items to the other editor, as you would expect.

— Clicking on the ↪ button in a collection or search result header bar.

— When clicking on items in the Project Bookmarks panel (subsection 10.3.1).

— Clicking on, or when pressing Return to load items via the Quick Search Tool (section 11.5).

Use of these methods to load items in the inactive split will not activate the other split. You can thus load supporting material with a single click and then keep typing as you were.

Changing How the Binder Works Persistently

By default the active split is synonymous with the targeted split. If you click in the top editor the header bar is shaded blue and is underlined in blue. You wouldn't be blamed for thinking that was one visual effect, but in fact it is two. The blue shade indicates the binder will load things into that editor when you click on them (targeted), and the blue underscore indicates that this is the split you're currently working within (active). If we can tell the binder to only target one split but can still of course easily work in either, then there will be times when the underscore is on the *grey* split, not the blue one (Figure 8.2).

The Navigate ▸ Binder Selection Affects ▸ submenu is how you will get to the point of seeing such variations in the first place. It contains the following options. If you still have that sample project handy from earlier in this section, you might want to play with each option to see how they work:

— *Current Editor*: This is the default behaviour out of the box. Whichever split you worked in last will be targeted.

— *Other Editor*: Whichever split is *not* currently active will be targeted. This way your current work space is never disturbed by usage of the binder.

— *Top|Left Editor Only*: <$include>

— *Bottom|Right Editor Only*: <$include>

— *Both Editors*: <$include>

— *None*: <$include>

12.2.4 Making Splits Load by Type

If you would like to use the split views to achieve a stricter form of navigation, where one is dedicated to primarily secondary navigation (such as browsing the contents of a folder) and the other to content itself, the Navigate ▸ Binder Selection Affects ▸ Open Non-Group Items in Other menu toggle may achieve what you are looking for.

This option is subsidiary to setting the binder to affect only one split (from that same submenu), and impacts how the *other* split, not affected by the binder, will work: by loading individual items directly. Thus, you can keep from loading text or images into the split you typically use for navigation alone, and consistently expect only content to load in the non-navigation split.

When the Navigate ▸ Outliner|Corkboard Selection Affects ▸ Other Editor menu toggle is enabled, this feature will work together with it in a way unlike how "Binder Affects" and "Outliner/Corkboard Affects" would typically work by themselves, even if you have them both enabled. When clicking on a group in the binder, two things will happen:

— The group will be loaded into the split the binder affects, and will displayed using whatever view mode you are using in that split.

— If the affected split is displayed as a corkboard or outliner, then the *other* split will automatically load whatever you were working on last within that group. For example if you click on a folder called "Geographical Distribution" and then within that folder click on a subdocument called "Summary", loading the text of that file into the other editor, then when next you load "Geographical Distribution" from the binder, "Summary" will automatically load in the other split as well.

  It's worth noting that this secondary behaviour may in fact load a group into the other split, if the last thing you clicked on in "Geographical Distribution" was a subgroup.

I'm Still Getting Groups in the Other Split

This setting is very specific: it only impacts what happens when non-group items are clicked in the binder sidebar. If you load a group through some other means (say a bookmark, or by clicking on a group in the navigation split) then Scrivener will dutifully load that group in the content split. This isn't a way of forcing the editor to never load groups, or forcing a split to never load text, but a way of directing how simple clicks in the binder tend to favour that approach. No other method of navigation, be it history, hyperlink navigation or what have you, will trigger this behaviour.

12.2.5 Linking Splits Together

In addition to tuning how the binder sidebar works with splits, you can also adjust how splits work together, or with attached copyholders within either split.

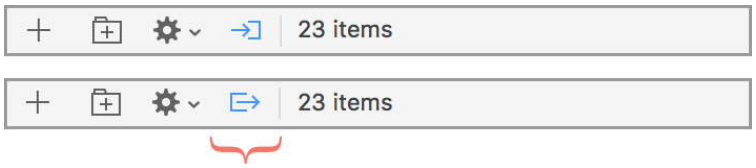


Figure 12.3: The "auto-load" button appears in blue when activated.

Both corkboard and outliner views have an auto-load feature that when enabled, will load any selected item(s) in the other split or the editor's copyholder, much like when clicking on an item in the binder. There are three modes of operation, two of which are illustrated in Figure 12.3:

1. When the editor has a copyholder attached to it, clicking this button will enable the auto-load into copyholder feature, and the icon will look like the top example. You can think of it has "sending" to the other side of the editor window. This can be enabled with Navigate ▸ Outliner|Corkboard Selection Affects ▸ Copyholder.
2. When the editor is split then clicking the button enables auto-load in other editor mode. The icon in that case looks like the bottom example, depicting the action as "sending" out of the current editor split. This can be enabled with Navigate ▸ Outliner|Corkboard Selection Affects ▸ Other Editor.
3. When disabled this button resembles the bottom icon, but will be drawn in grey to indicate it is inactive. Also disabled with Navigate ▸ Outliner|Corkboard Selection Affects ▸ None.

If the editor is split *and* the current editor has a copyholder then clicking on this button will rotate between all three modes. When neither copyholder nor

splits are available this feature does nothing, even if it has been activated in the past. It will wait passively until the right conditions exist.

If you have a keyboard with a Touch Bar and your context is currently within a corkboard or outliner view, then you will find a button on your keyboard that resembles the button in the footer bar. Use this to toggle auto-load modes directly.

Return to chapter ⤴

12.3 Saved Layouts

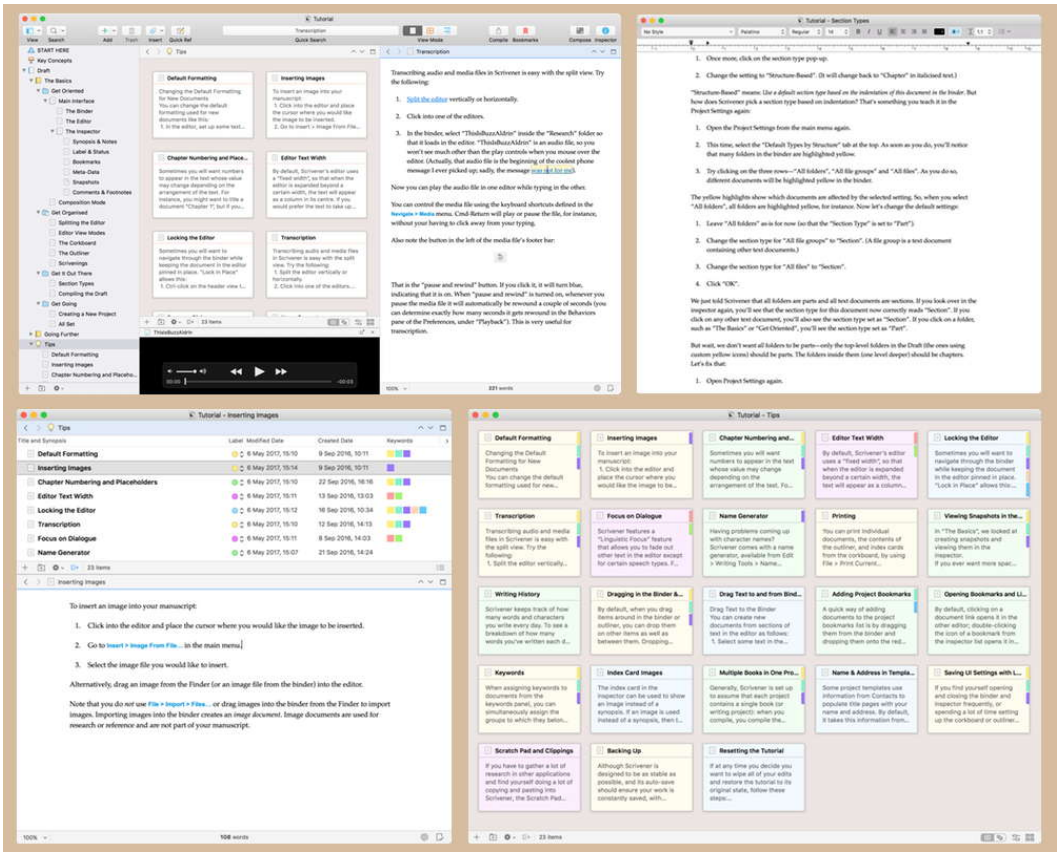


Figure 12.4: Scrivener's diversity and flexibility benefits from being able to save your workflows into Layouts.

In the course of using Scrivener, you may find that you shift how you use the project window, depending on what you are doing, or which phase of the project you're in. In the early stages of a project you might focus on a large corkboard without any other interface to get in the way; as structure starts to emerge, you may have the binder open with the corkboard in the top editor and text in the bottom editor; whilst composing, you may have only the text open, with

the binder, inspector and toolbar hidden. Or perhaps you switch between computers frequently, and like to use an expansive project window on your desktop monitor, but prefer a compact window on your laptop.

Or, maybe you'd *like* to do these things, but tend not to because you're put off by the amount of adjustment it would take to change the window around.

By saving the settings of the window for future recall, the Layouts feature makes rapidly switching between these workflows a cinch. Once you get things set up the way you like, you never have go through the process of hiding the binder, toggling the format bar, resizing the window and so forth. You can even optionally specify outliner column settings and other display preferences like whether or not labels tint index cards.

> **Upgrading from Scrivener 2**
>
> If you have older layouts saved from older versions of Scrivener, you will find they still exist in the list in the new version. You may try to use them, but we cannot guarantee that they will work properly. It would be safest to try and reproduce their settings by hand, and then update the layout with the new version of Scrivener.

To iterate what may already be obvious: layouts are not a part of any particular project, and as such they will not save information specific to any one project, such as things like the collection you are using in the sidebar, whether any Quick Reference panels are open or which folder you are viewing as a corkboard in the editor (it will however save that a particular editor is using the corkboard view mode). Even within one individual project, these details might change or no longer be relevant as time goes by. Here is a list of things layouts *do* save:

— Window size and position and the sizes of elements within them, such as how wide the binder is or the ratio between the two splits.

— Binder, collection list and inspector visibility.

— Split type (horizontal/vertical/none).

— Corkboard (freeform & layout)/outliner/scrivenings/editor mode for each split, regardless of the currently selected content.

— Which split(s) the Navigate ▸ Binder Selection Affects option is set to.

— Whether Navigate ▸ Corkboard|Outliner Selection Affects the other editor or copyholder (but not the presence of a copyholder itself, as that is like Quick Reference panels a project-dependent piece of information).

— Header and footer bar visibility in either split.

— Whether editors are locked in place.

— Ruler and format bar visbility.

— Line numbering in the editors.

— Full Screen status.

12.3.1 Creating a New Layout

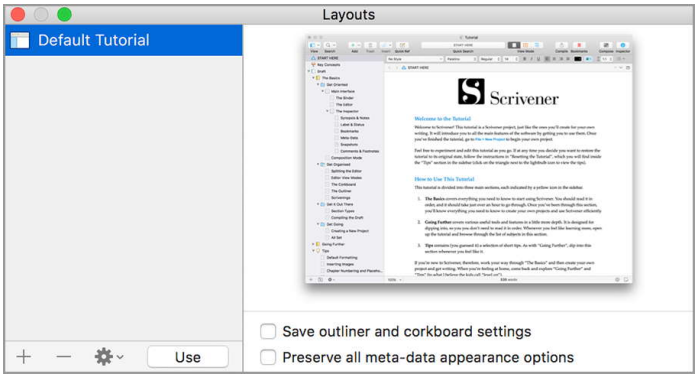


Figure 12.5: Layouts panel with the interactive tutorial's window settings saved.

To get started, open the Layouts panel (Figure 12.5) with the Window ▸ Layouts ▸ Manage Layouts… menu command (⇧⌘). If you do not have any layouts saved yet, you won't see much beyond some help text.

1. If you have more than one project open, make sure the project you wish to capture is the active window by clicking anywhere within it. Whether or not the project window is in Full Screen mode will be saved, so if you would rather not have that a part of your layout, be sure to leave Full Screen mode first.

2. Click the + button in the lower left corner of the panel.

3. Type in a meaningful name for the layout.

   A screenshot of the window will also be taken and displayed on the right (you can update it later on if you don't quite like how it turned out).

   At this point all of the important details of your project window settings have been saved. They can be recalled later at any time into any project window you wish to apply them to.

4. Select from either of the options listed below the thumbnail that you feel are relevant to the layout. These options are always saved into the layout—the checkboxes merely determine whether using the layout will set these options on the current project window. They can thus be freely toggled at any time to produce a limited or expanded effect on the current project window, without changing the underlying layout.

Save outliner and corkboard settings All corkboard display settings, such as card size, ratio, card wrapping, and so on will be restored when using the

layout. Additionally the corkboard modes for freeform and label view will be restored, and since this has the effect of setting these modes to the containers being viewed, use of a layout will modify the items at the time of application.

In the outliner, the layout will restore which columns are visible or hidden, along with their positioning and widths. Custom metadata columns will be restored if matching names are found within the project.

Preserve all metadata appearance options This will determine whether or not label tinting used in the various areas of the interface; and pin, stamp, and keyword chip visibility in the corkboard, will be restored when using the layout.

12.3.2 Switching Between Layouts

First off, it's important to know that when you apply a layout to your project window, you'll be overwriting all of its current display settings with what has been saved in the layout. Consequently if your current view settings are important, you might want to save them into their own layout first. There are three ways to switch between layouts:

1. Use the Window ▸ Layouts ▸ submenu and select the saved layout from the list.
2. The "View" button in the main application toolbar (Figure 4.2) will list any available layouts at the bottom of the menu.

   With both of these methods, if you hold down the Option key while selecting the layout, its Full Screen setting (whether on or off) will be ignored.
3. With the "Manage Layouts" panel (Figure 12.5), use one of the following methods:

   — Double-click the *icon* of the layout you wish to load.

   — Press the Return key on the selected layout.

   — Select the layout and click the Use button below the list of layouts.

12.3.3 The Built-In Layouts

Scrivener comes with several built-in layouts that demonstrate the various navigational features of the project window. Some are intentionally quite simple, designed to help you focus on a particular way of working, others gather a number of individual settings into more complex workflows that may not obviously achieved by their parts.

Unlike regular layouts, these built-in layouts will in general not change your window size or position, unless the layout causes any main editor splits to become more narrow than your preferred editor width.[4]

Also unlike regular layouts, they will not change your underlying settings in a permanent fashion. For example if you have a lot of columns added to the outliner and decide to make use of the "Three Pane (Outliner)" layout, which reduces the complexity of the outliner to a very simple browser, upon selecting one of the built-in layouts that leaves these settings alone (such as "Default"), you will find your original column settings intact. Changes you make yourself while using these layouts will make a permanent impact on your metadata settings.

Save your current layout before experimenting

Even though these layouts are intended to have a minimal impact on your project settings, they will still necessarily change how your window is set up. If you have invested a lot of time in how your project window is currently set up, you might want to save those settings into your own custom layout first (subsection 12.3.1).

Since the built-in layouts cannot be modified they will not appear within the Window ▸ Layouts ▸ Manage Layouts... panel. They only appear in the main menu and under the "View" button in the application toolbar.

Default

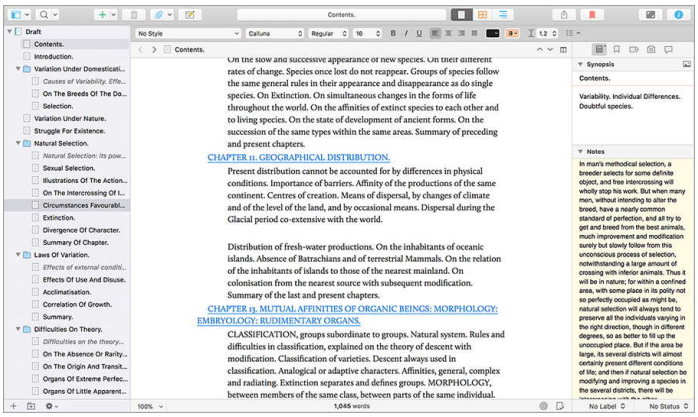


Figure 12.6: The "Default" built-in layout.

The first and most important layout to be aware of is the one that gets things back to ground zero. The "Default" layout is designed restore a project window

[4] Set in the Appearance: Main Editor: Options preference tab (subsection B.5.8), with the **Default editor width** setting.

to a basic look, removing the non-active split view, closing all copyholders and opening the inspector and binder. It will also clear those navigation settings that modify how the project window behaves when you click on things—in the exact same fashion as using the Navigate ▸ Clear All Navigation Options menu command (section 12.5). (That command makes for a good alternative to "Default", if you'd rather leave the layout of the window alone but get the default behaviours back.)

Three-Pane (Outline)

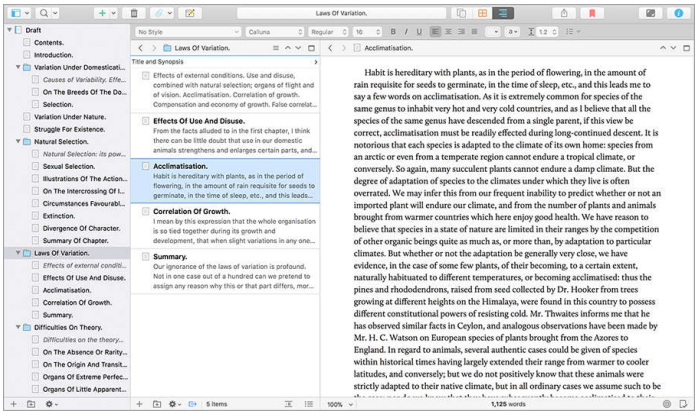


Figure 12.7: The "Three-Pane (Outline)" built-in layout.

Three-pane browsers are a popular way of navigating through large amounts of information. They typically feature a sidebar (much like the binder) for selecting folders, then display the contents of that folder in a second view, and finally clicking on things inside of that view will automatically load the contents of what you click on into the third pane. Here are the settings it applies:

— Navigate ▸ Binder Selection Affects ▸ Left Editor Only: by itself, this causes what you click on in the binder sidebar to be loaded directly into the left split no matter what.

— Also within that same menu, the Open Non-Group Items in Other option has been enabled. This redirects non-group items you click on in the binder over to the right split no matter what. Details on how this feature works are described in Making Splits Load by Type (subsection 12.2.4).

— Navigate ▸ Outliner Selection Affects ▸ Other Editor: as indicated by the blue highlighted button in the outliner footer bar, any selection you make in the outliner will automatically load in the other split.

  All three of these options together greatly modify how Scrivener typically works, going from a strict two-pane design to something more like an email browser or notebook program.

— Visually, the outliner will be using a fixed row height (subsection 8.3.5), which keeps long synopses from dominating the view, and the column settings for it will be altered to only show “Title and Synopsis”.

— The left split, meant to focus mainly on content, will switch to Scrivenings view mode.

Three-Pane (Corkboard)

This layout is functionally identical to the “Three-Pane (Outliner)”, save for using a stylised corkboard view for primary navigation, rather than an outliner. If you prefer the additional visual display of metadata that the corkboard affords you, this can make a good alternative. It temporarily changes the following aspects of the corkboard:

— View ▸ Corkboard Options ▸ Cards Across will be set to “1”.

— The **Size to fit editor** setting will be enabled, from within the Corkboard Options (subsection 8.2.6) panel.

— The **Spacing**, from that same panel, will be adjusted to a small amount (though it may in fact expand if you were using less spacing before).

Editor Only

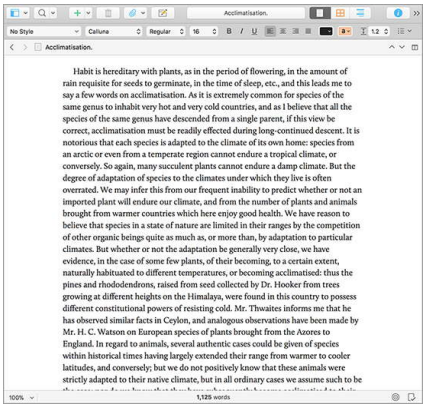


Figure 12.8: The “Editor Only” built-in layout.

Does what it says on the tin. This very simple layout simply removes both sidebars, closes the split view and any copyholders. Which split it chooses will depend on what you are currently working with. It will favour the split that is showing a text editor or Scrivenings session. If both (or neither) splits match that description, then the active split will be used. Lastly it will set the group view mode to Scrivenings. Thus if you have a split interface with two corkboards, the active corkboard will become the editor view you focus on and its view will be switched to Scrivenings so you can work with the text. The view will be scrolled to the first selected card or outliner row from the previous view.

If you would prefer an even cleaner layout, with no ruler, format bar or header and footer bars in the editor, you'll find a custom layout called "Editor Only (Clean)" in the Extras Pack (Appendix F).[5]

Corkboard Only

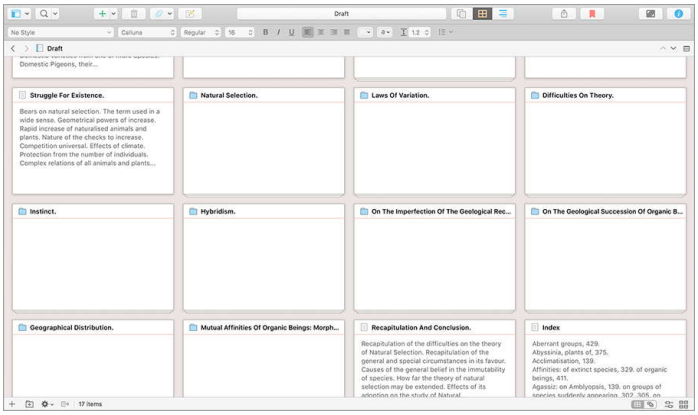


Figure 12.9: The "Corkboard Only" built-in layout.

Much like the "Editor Only" layout, but aiming to provide a clean view for focussing on structure, rather than content. If relevant, when choosing which split to focus on, the layout will select the split with a corkboard. It will otherwise use the active split, changing the view mode to Corkboard if necessary. If you were editing a file then the layout will select the *parent* group, scrolling to and selecting the card for the item you had been editing.[6]

If would prefer an even cleaner corkboard with no toolbars, you'll find a custom layout called "Corkboard Only (Clean)" in the Extras Pack (Appendix F).

Centered Outline

The main purpose of this layout is to quickly and simply display a very clean outline view with only the Title (and optionally Synopsis) column visible and the View ▸ Outliner Options ▸ Center Content menu toggle enabled. The editor space will be cleaned up, closing splits if necessary and removing any copyholders.

As with the "Corkboard Only" layout, the split showing an outliner will be preferred, otherwise using the active split, and the parent of the currently edited text document will be navigated to if necessary. Unlike that layout, the binder and inspector visibility will be left alone.

---

[5] If you need instructions for installing a layout, refer to Managing Layouts (subsection 12.3.5).

[6] If you're curious on how to do that yourself from time to time, the Navigate ▸ Go To ▸ Enclosing Group menu command does the same thing, although in that case it will preserve the editor's current group view mode.

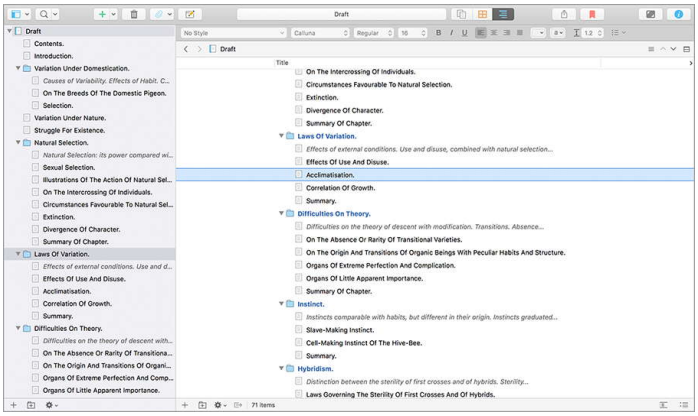


Figure 12.10: The “Centered Outline” built-in layout.

Dual Navigation

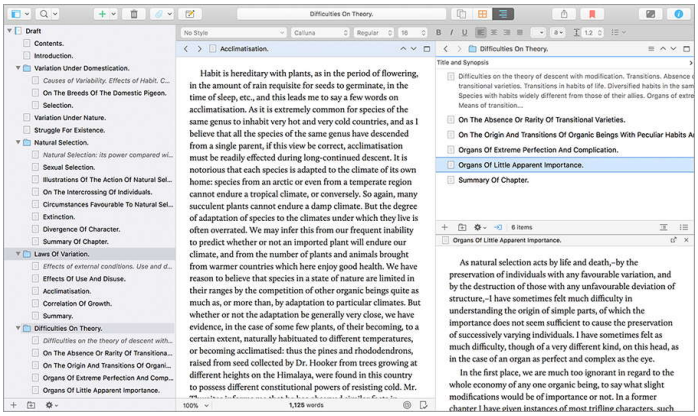


Figure 12.11: The “Dual Navigation” built-in layout

If you’ve ever found yourself wishing that you could have a kind of second binder, so that you could browse and edit more than one file at once, this layout may do the trick. You can think of this layout as roughly dividing the project window down the middle into two separate workflows:

— On the left side, the binder and the left split will be bound together exclusively. You can use it for text editing, outlines, viewing PDFs or whatever you need.

— On the right side things are a little more interesting. This side is isolated from the binder and left editor, making it a more static place to work with multiple items in a sort of “mini-binder” approach, where what you click on in the top editor will be automatically loaded in its copyholder below.

If you want to change the group you are browsing in the right side of the project window, you have every single means of navigation at your disposal save for clicking on things in the binder:

— Option-Click on items in the binder.

— Drag and drop from the binder to the right editor's header bar.

— Use the Navigate ▸ Go To ▸ submenu.

— From the left editor, when using a group view, Navigate ▸ Open ▸ in Right Editor (⇧⌘O).

— Quick Search.

— And of course the history back/forward and next/previous item buttons in the editor's header bar.

Here is a list of settings this layout applies, should you wish to tweak how it works or create your own similar layout:

— Navigate ▸ Binder Selection Affects ▸ Left Editor Only: this keeps the binder focussed on the left side of the project window.

— In the right split, Navigate ▸ Outliner Selection Affects ▸ Copyholder has been set, as indicated by the blue highlighted button in the outliner footer bar.

— The group view for the right side will switch to outline mode.

— The layout will try to use whatever content you were working on in the editor(s) prior to using it, but if it cannot find a combination it likes, it will just load the entire Draft folder into the right-hand browser split and the first item with text content that it finds in the Draft into the left split. If you want to get back to what you were doing before the application of this layout, you can use the history buttons in each editor header bar.

Refining A Built-In Layout

You may very well find that you like the basic idea of one of our built-in layouts, but would like to make adjustments to how it works. Although you cannot modify the built-in layouts, for the most part this can be done simply by creating your own layout after making the desired modifications to the project window, as described earlier in this section.

Although they work similarly, built-in layouts are not the same as the layouts you create, and they have some characteristics you won't be able to replicate:

— The "Dual Navigation" layout opens a copyholder panel below the outliner in the right-hand split. Layouts cannot open copyholder panels. However the setting that causes clicks to load in a copyholder will save, so as soon as you open a copyholder it will start working.

— Built-in layouts can *temporarily* override metadata settings like which columns are showing in the outliner or how many cards across the corkboard displays. When you return to "Default", you will find your original

settings are intact. The layouts you create on the other hand can either only permanently change those settings or disregard them entirely.

— The built-in layouts only make selective changes to settings, rather being a full snapshot of all view settings, navigation options and optionally metadata display options. For instance, if you prefer not to use the footer bar in your editors and have it switched off with the View ▸ Editor Layout ▸ Hide Footer View menu command, built-in layouts will not change that setting. A custom layout absolutely will.

— Some of the built-in snapshots are "aware" of what you're working with in the project window and optimise their behaviour based on that. For example the "Corkboard Only" layout will select the split using a corkboard view, and if found, prefer that split even if the other is currently active. Your own layouts will more simply force the saved view mode change upon the current editor(s).

Hiding the Built-In Layouts

Once you've established a number of your own layouts, you may want to hide these starter layouts to clean up the menus:

1. Use the Window ▸ Layouts ▸ Manage Layouts… menu command (⇧⌘) to bring up the layout management window.
2. Click the ✲ button and select the "Hide Built-In Layouts in Menus" command.

You can always bring them back with the "Show Built-In Layouts in Menus" command.

12.3.4 Setting a Full Screen Default

It is possible to select a layout for use by all projects that are taken into Full Screen mode, as a kind of default layout for that way of working. This is a special behaviour that breaks some of the rules with regards to how project layout in general works:

— The designated Full Screen layout will be used when taking a project into Full Screen mode. When taking that project out of Full Screen mode (or even simply closing it), it will resume using its original layout settings. Thus the full screen layout will not be saved into the project's own display settings.

— Consequently, changes made to the layout of the project while in full screen mode will not be preserved upon return to standard multitasking. Everything will be precisely as you left it before entering full screen mode.

— Changes that fall under the auspices of metadata appearance and outliner/corkboard settings *will* on the other hand be preserved. For example if you take a project into full screen and then decide to tint index cards with label colour with View ▸ Use Label Color In ▸ Index Cards, then upon return to standard multitasking, index cards will remain tinted. On the other hand if you open up the tab stop ruler in Full Screen mode, it will be gone once you return.

12.3.5 Managing Layouts

All layout management is done within the "Manage Layouts" panel, which can be loaded with the Window ▸ Layouts ▸ Manage Layouts... menu command (⇧⌘).

To remove a layout you no longer need, select the layout and click the – button along the bottom of the layout list on the left. You will be warned that once it is deleted you'll be unable to retrieve it. This warning can be dismissed so that it no longer appears, if you wish.

To update an existing layout using the current project window, simply select that layout in the list, and click on the ⚙ button in the footer bar, choosing "Update Selected Layout". A new screenshot will be taken and the old settings will be updated to reflect the current window layout.

Layouts can also be exported and installed into Scrivener[7]. For example you could share your layouts with others by posting them online, or download layouts and install them into your copy.

— **Exporting**: using the ⚙ button, select the layout you wish to export, and then choose "Export Selected Layout". The layout will be saved as a file in the location you provide.

— **Importing**: use the "Import Layout" command to import a layout file (.scrlayout).

Layouts can be easily accessed on the disk as well. Use the Scrivener ▸ Reveal Support Folder in Finder menu command and drill down into the "Layouts" subfolder to find the `.scrlayout` files themselves.

Return to chapter ⤴

12.4 Cutting Through the Forest

One of the ways in which you can work in Scrivener is with interleaved notation among the very pieces of text that comprise your work in progress. This is a

[7] It is not possible to share Layouts between Windows and Mac versions of Scrivener, owing to the many technical differences in how a window is set up.

method that some prefer, as it keeps their thoughts and ideas immediately contextual with the text itself, and it supports a working method where notes are more extensive than what can easily be put into annotations attached to the text itself. In Scrivener, you can place text documents right alongside the documents used to contain portions of your Draft, and then instruct these special note documents to not compile (subsection 13.5.1)—in other words when you turn the Draft folder into one long file, these note files will be omitted.

Ordinarily, when you view a portion of the draft with the Scrivenings view mode (section 15.10), you will see all of your notes and draft text together. This can be advantageous, especially if you use an alternate font for your note files—but there will be times when you want to just see the compilable text that will become your book.

You can also select one or more containers and use the Navigate ▸ Open ▸ (with) Compilable Subdocuments menu item. The name of this menu item will change depending on whether or not you've set up your preferences to consider the container as part of a Scrivenings session. Some prefer to omit the container, and so disable the **Include enclosing folder text in Scrivenings mode** checkbox in the Behaviors: Folders & Files (subsection B.4.5) preference tab. When this setting is disabled, the menu item will omit the 'with', and only the children of the selected container will be included in the flat list. The result of this action will be a flat multiple selection of items (section 6.4) that you can view using any view mode you prefer.

Return to chapter ⤴

12.5 Clearing Navigation Settings

If for any reason you wish to clear all of the settings that impact how clicking works in the project window, the Navigate ▸ Clear All Navigation Options command is a handy way to do so in one move. The following changes will be made:

— The Navigate ▸ Binder Selection Affects mode will be set to "Current Editor".

— The Navigate ▸ Outliner Selection Affects mode will be set to "None".

— Navigate ▸ Editor ▸ Lock in Place will be disabled.

— Finally, the inspector will be unlocked from working with only one split (this happens as a secondary effect of closing split views).

Return to chapter ⤴

12.6 Quick Reference

If you've been looking for a way to load a document into its own window, this is the section for you. Quick Reference panels (sometimes shortened to "Quick

Ref") are simplified editors and media viewers, and you can have as many of them open as you want. The software even keeps track of which you've used during the session, so that closed windows can be very easily reopened from the Window menu.

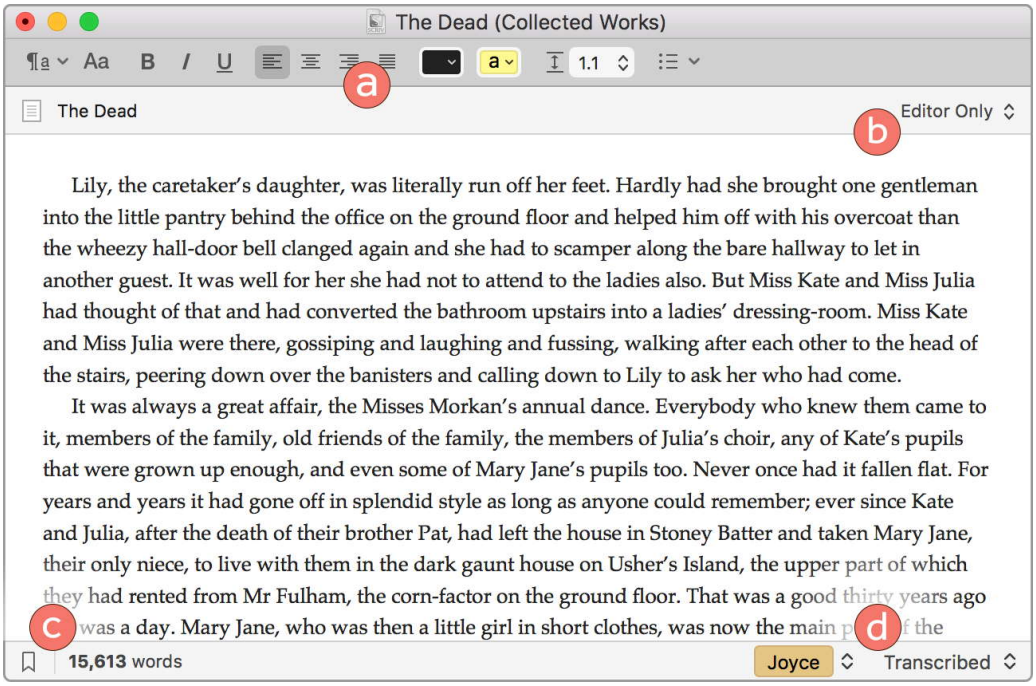


Figure 12.12: With Quick Reference, open bits of your project into separate windows featuring for editing or simple reference.

The panel itself is composed of four major parts (excepting the main editor/viewer in the middle), two of which are hidden by default.

a) The format bar and header bar along the top provide basic formatting tools just like in the main project window; the header bar supplies the identity of what we are viewing.

b) Along the right-hand side of the header bar is a dropdown which reveals an inspector split when used. This split can be either vertically or horizontally oriented.

c) The bookmark button reveals the project bookmarks sidebar, turning this window into a capable browser for your bookmarks. Refer to Working with Bookmarks in a Quick Reference Panel (subsection 10.3.3) for further detail. Document statistics and scripting hints (in script writing mode) will be displayed along the bottom.

d) On the bottom-right you have quick access to the Label and Status metadata fields.

As for the thing in the middle of these points, the editor itself: this is an full-power text editor when viewing standard file or folder documents, and a standard media viewer just like the main editor when viewing PDFs, images, or QuickTime documents[8]. The main difference between a Quick Reference editor and those featured in the main project window is that it focusses on the content of one item. It is not capable of showing group views or Scrivenings sessions.

12.6.1 Opening an Item in Quick Reference

There are five easy ways to open a document in Quick Reference mode:

1. Select one or more documents in the sidebar or a view and tap the Spacebar. If more than one document is selected, multiple Quick Reference panels will be opened at once.

   When selected from one of the editor views (including the text editor), use the Navigate ▸ Open ▸ as Quick Reference instead. You can also make it so Spacebar works in the corkboard and outliner with the **Space key opens selected documents in...** setting, in the Behaviors: Navigation preference pane.

2. Use the Navigate ▸ Open Quick Reference ▸ from the main application menu to open a document from anywhere, without disturbing your existing workspace in the project window or having to leave composition mode.

   Each item in the binder will be organised into submenus based on the project hierarchy. If you select one of the groupings itself, the command will use that group. For example if you have a folder named "Chapter One" and have typed some notes into it, you could target that folder specifically with this command.

3. Use the "Look Up" trackpad gesture (three-finger tap) to open anything in a Quick Reference panel from a group view or the binder sidebar. This option requires the "Look up & data detectors" option in your System Preferences: Trackpad pane to be enabled.

4. When using the Quick Search tool (section 11.5), select a search result and press ⇧Return to load the result as a Quick Reference, instead of into the main editor.

5. Drag and drop the item from any binder sidebar or group view mode onto the "Quick Ref" toolbar icon.

You may have noticed that Quick Reference windows are used in a few contexts within Scrivener automatically, such as when double-clicking on book-

[8] Unlike loading film and audio in a split, the remove pause and resume commands will not work in Quick Reference panes, as you can have many audio files open at once, and Scrivener would not know which one to trigger.

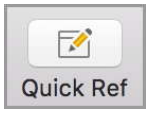


Figure 12.13

marks in the inspector, or clicking on hyperlinks in composition mode. Most of these behaviours can be modified in the Behaviors: Document Links preference pane (subsection B.4.2).

Once you have closed the window, for the remainder of your session with that project it will be remembered in the Window ▸ Closed Panels ▸ submenu, much like a modern web browser might save all of the tabs you have recently closed. Feel free to close Quick Reference windows whenever you are immediately done with them, as they will remain easily accessible to you from that menu.

You can also choose to have panels left open reopened for you the next time you return to the project. Set **Reopen Quick Reference panels when opening projects** in the General: Startup preference pane.

12.6.2 The Elements of the Panel

Before we go into the various elements of the panel, it would be good to make note of how they all fit together. Each Quick Reference panel you open will remember the settings you've used within it, as well as the overall size of the window and split placement. If you left the inspector open and dragged it down to the shortest possible height, and you open that panel again in two years it will be just how you left it.

When you open a document in a Quick Reference panel for the first time ever, it will adopt the window size and position used from the last window you *changed*. It will not adopt inspector settings.

The Header Bar

Figure 12.14: The Quick Reference header bar.

Along the top of the window, below the Format Bar if you have displayed it, you will find a simplified header bar (Figure 12.14).

The icon to the left of the title can be dragged, acting as a proxy for the file it represents. It can be dragged to the binder to move the file to the location you drop it, it can be dragged into a bookmarks inspector pane, etc. To "dock" a Quick Reference panel back into the project window as a copyholder, try holding down the Option key while dropping its icon onto an editor header bar.

The icon also has a contextual menu with a few features:

— *Reveal in Binder* (⌥⌘R): displays the location of the currently edited file in the binder, opening the sidebar, switching to the binder and unfolding any levels of hierarchy to do so, as necessary.

— *Take Snapshot* (⌘5): take a quick snapshot of the document's text in its current state. You will not be able to view snapshots from within the Quick Ref window itself,

Following the icon is the title. As with the main editor header bars, you can edit the title of the document right here by clicking into the title text (or using the ^⌥⌘T shortcut). Press Return to confirm and return your cursor to where it was in the editor previously. If left blank, the title will show a little of the synopsis card text as a placeholder title, or lacking that, the first few works from the top of the document itself. See Titles and Adaptive Naming (section 7.3) for more information on how that works.

Next along the top is a button for viewing information from the inspector. It will be labelled "Editor Only" by default, and that is the choice you would make to close the split when you are done with it. Read more about this feature in Quick Reference Mini-Inspector (section 12.6.2).

Finally, when a split has been opened (such as in our example, where we are viewing the Synopsis of the document), a final button will appear that flips the orientation of the inspector split.

The Footer Bar

The footer bar, running along the bottom of the view, has three primary elements:

1. The "Project Bookmarks" button. This is described below (section 12.6.2).
2. This area will display either statistics or scripting element dropdown menu (which can be called up with the ⇧⌘Y shortcut, just as in the main editor), depending upon the editor mode.

   When viewing a multimedia file, the "Pause and Rewind" toggle button (section 8.1.3).
3. Finally, along the right hand side are two dropdown menus for changing the label and status metadata fields, in that order.

Quick Reference Mini-Inspector

Quick Reference panels can display inspector metadata for the document you are working with in the panel.

The panes you can make use of here are not organised in precisely the same fashion as the inspector itself, on account of their being divided into specific areas of content. For example the "Notes" tab in the inspector is actually comprised of three different pieces of information, each of which has its own option here:

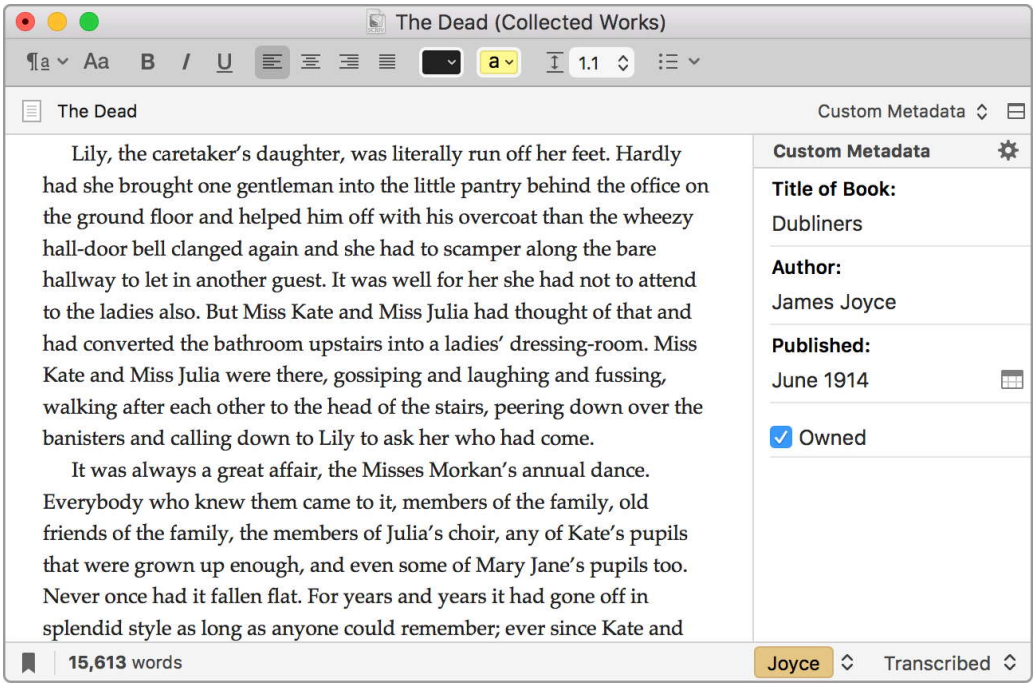


Figure 12.15: A Quick Reference window can display a simplified inspector along the bottom or to the right (as shown).

1. *Editor Only*: the default state, which removes the split and focusses purely on the content.
2. *Synopsis*: the text content of the index card
3. *Picture*: the graphical content of the index card, if applicable.
4. *Notes*: document notes, from the first tab of the inspector.
5. *Keywords*: document keywords; this table also serves as a drop-target for keyword drags from the Project Keywords window.
6. *Bookmarks*: the list of bookmarked items for this document. Double-click to load them in their own Quick Reference panel. Refer to the following section if you're looking for project bookmarks.
7. *Custom Metadata*: the custom metadata fields in a linear list of editable text fields.
8. *Comments/Footnotes*: where you can view the various linked footnotes and comments within the text editor. This split will appear automatically if you create a note.

Refer to the Inspector (chapter 13) itself for greater detail on how these particular tools are used.

The Bookmark Sidebar

Any Quick Reference window can easily browse through project bookmarks (assuming it can display the type of bookmark in question). Click the bookmark button, marked (c) in Figure 12.12 to toggle the sidebar visibility, or use the ⇧⌘B shortcut. Click on the bookmarked document you wish to edit to view it in window.

For detailed reference on the sidebar itself, refer to Working with Bookmarks in a Quick Reference Panel (subsection 10.3.3).

When using the bookmark sidebar, the earlier comments on Quick Ref windows remembering their size and split settings will be not be applicable on account of their being displayed in a collective browser, sharing a window size and settings with other documents.

12.6.3 Tips for Using Quick Reference Panels

Floating Panels for Maximum Visibility

Quick Ref panels have an additional capability in the Window ▸ Float Quick Reference Panels menu toggle (^⌘Q). When set, all of a project's panels will "float" over all other windows in Scrivener. Floating can be particularly useful when the project window itself is as large as the entire screen, or in composition mode, where the backdrop would ordinarily hide the panels. Consider toggling float on and off as needed to bring the panel workspace to the front, and then disable it and with a click dismiss them all and return to the work you are focussing on.

Displaying Research in Quick Reference

It's not just about the text. These windows can also act as capable PDF and image viewers, as well as working with other multimedia. For further information on the particulars of how they work inside the editor area, refer to Viewing Media in the Editor (subsection 8.1.3).

The Window ▸ Zoom menu command, when use on a Quick Reference panel that is viewing an image or PDF will expand or contract the size of the window to fit the content. If you would prefer things work the other way around, double-click on the image and click the **Scale to fit** option. With PDFs you can right-click on the PDF and select "Automatically Resize", or select the same from the View ▸ PDF Display ▸ submenu.

Setting Text Zoom

Although there is no place for the visible zoom control like in the main editor footer bar, you can independently zoom the text scale within each Quick Reference panel as needed, using the View ▸ Zoom ▸ submenu, or the ⌘> and ⌘< shortcuts to decrease and increase the magnification levels, respectively.

Return to chapter ⤴

Inspector

13

In This Section...

The Inspector is the panel on the right-hand side of the project window that displays all metadata, snapshots, notes, compile settings and other sundry associated with the document selected. You can also display project bookmarks in this pane, making it a useful place to keep global notes handy. In this chapter we'll go over each element within the inspector and how to use it, referring you to further discussions on the topic if necessary. First, we'll take a look at using the panel itself, and then go over each of the tabs in turn.

You can toggle whether the Inspector is visible with the View ▸ Show|Hide Inspector menu command (⌥⌘I), or by clicking the blue "i" button on the far right-hand corner of the toolbar. The panel is separated into three distinct parts: the tab row and document title at the top, the inspector panel itself occupies the

middle area and finally at the very bottom we can see any label or status markers assigned to the inspected document.

There are two other places where information from the inspector can be examined and edited: Quick Reference panels in a split, and in composition mode where a floating inspector will be used. These two areas contain a subset of the features available to the full inspector. The only tab not available at all to either of these is the Snapshots tab.

13.1 Inspecting Items

With a fairly basic project window, using the inspector to examine a document's metadata and other auxiliary information should be straightforward. You click on a thing in the binder, it loads in the editor, and its metadata loads into the inspector on the right.

Although the distinction may seem unnecessary in light of the above, it is better to think of the inspector as being a tool for examining what is currently selected in the active editor. What you are inspecting can become disconnected from what you are doing in other editors or even the binder in some cases. For example if the editor view is split and you're using the right half, but you've set the menu setting Navigate ▸ Binder Selection Affects ▸ Left Editor Only, then it will be normal to click on things in the binder and have them load in the *inactive* editor—and thus not be inspected. Of course if you switch your focus to the left editor so that the header bar has an "underscore" along the bottom, then the inspector will be looking at the left editor.[1]

When copyholders are in use (subsection 8.1.5), the inspector will track where your cursor is within the editor, making it possible to inspect documents in the copyholder. With two splits and two copyholders, there can be up to four possible things you might be inspecting at any given time.

There are some things that cannot be inspected. Collections or search results, when viewed in the editor, and the three root folders won't have anything to inspect at all, nor multiple selections of items; in these cases the inspector will merely show the project bookmark list, having nothing else relevant to show. Viewing a snapshot in the editor is a special case—snapshots don't have metadata themselves that can be edited here, but any linked comments or footnotes that were in the text at the time of the snapshot will be preserved as well, and so a tab for viewing (not editing) them will be made available.

---

[1] Read more about editor focus in the section on the header bar (subsection 8.1.1).

When I click on a folder something else gets inspected?

When it comes to groups, Scrivener will remember what you were doing the last time you used that group, which often means it will remember a selection within that group. In turn that means the items in the corkboard, outliner, or the section of text you are editing within a Scrivenings session is what ends up being inspected, not the folder. If you'd like to inspect just the group you clicked on, use the Edit ▸ Deselect All menu command on the editor, or simply click in the background to deselect all cards or rows. In a Scrivenings session, you'll need to dismiss the session by turning off Scrivenings (⌘ 1).

13.1.1 Locking the Inspector

In cases where you do not want the inspector to use the active split, it can be locked to a specific split so that when the other editor becomes active, the inspector will continue to display information about the selection in the editor you locked it to. Either use the Lock Inspector to Editor command from the split's header bar contextual menu (section 8.1.1) or right-click in the inspector header bar itself to lock it to the current editor.

Figure 13.1: The split the inspector has been locked to displays this icon.

Once locked you can use either of these methods to relinquish the lock. You can also click on the red indication icon in the targeted split's header bar (Figure 13.1) to clear the lock. And naturally, if either split is closed the locked condition will be removed.

Return to chapter ⤴

13.2 Using the Sidebar

Now that we've established how to use the inspector in general, let's take a look at the stuff it actually does—what these individual tabs represent, and how to use the sections that appear within them.

13.2.1 The Inspector Tabs

In order of appearance, the tabs as shown in Figure 13.2 are:

1. *Synopsis & Notes*: the first tab, shaped like a notepad, contains the "index card" that will represent this item on any corkboard viewing it, and below that a scratch pad where you can jot down notes pertaining to the item.

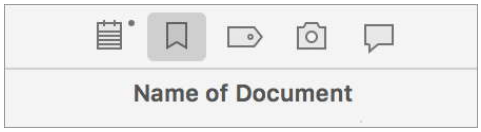


Figure 13.2: The inspector header with the Bookmarks tab selected.

2. *Bookmarks*: the second tab from the left contains listings for both project and document bookmarks (section 10.3). Switch between the two by clicking on the header, or by pressing ⌘6. Below the list is a preview and editing area.
3. *Metadata*: access to custom metadata, keywords, compile settings and general information such as created and modified dates. For more information on metadata in general, check out Organising with Metadata (section 10.4).
4. *Snapshots*: saved revisions of the text document can be created, managed and reviewed from this tab. Read more about Using Snapshots (section 15.8).
5. *Comments & Footnotes*: the last tab will list any linked footnotes and comments (section 18.3) found within the text currently being displayed in the editor.

Here are a few guidelines and tips to using the tab buttons in the inspector:

— Below the tabs, the title of the item being inspected will be printed for your reference.

— The selected tab will have a shaded background. In the figure, "Bookmarks" has been selected, which may show bookmarks added to this document, or the global project list of bookmarks.

— Only those tabs that are relevant to the document you are inspecting will be shown. You won't be seeing footnotes while viewing a picture of a mountain!

— If a particular tab has information entered into it, a dot will appear in the upper-right corner of the button. In the reference figure, the dot above the first tab, "Notes" indicates this document has either a synopsis or notes added to it.

Label and Status assignments At the very bottom of the inspector you'll find universal access to the "label" and "status" fields. In the example figure (Figure 13.3), we've marked this document as having reached the first draft point, but we clearly aren't very happy with it and would like to rewrite it at some point.

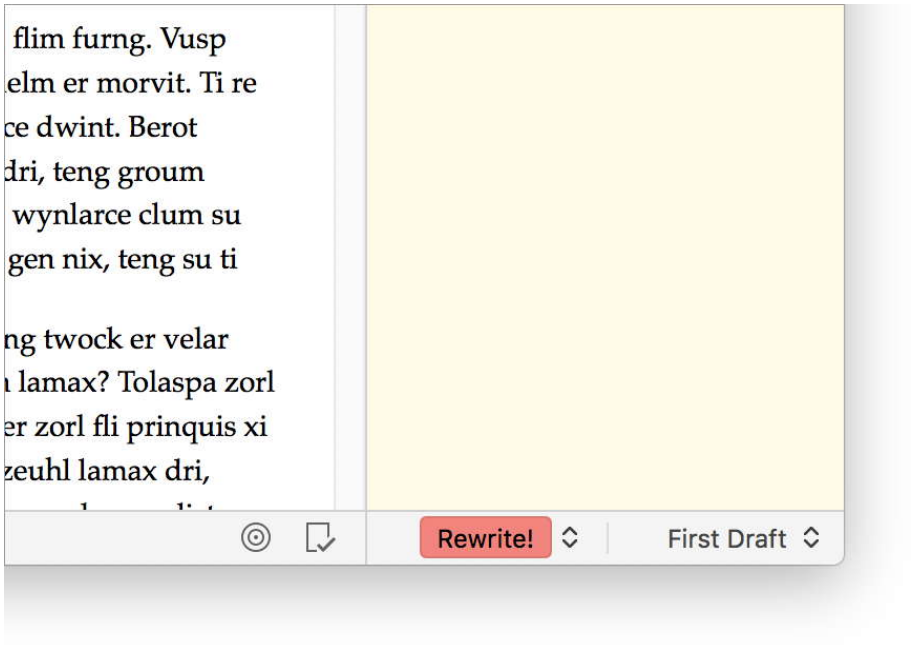


Figure 13.3: The label & status fields at the bottom of the inspector.

Figure 13.4: Appearance of a collapsed inspector section.

Collapsing sections within a tab Whole sections can be collapsed within the tab, if the section header has an arrow icon to the left of it. You might never use keywords in a particular project, but have a lot of custom metadata fields. Click the arrow to the left of the section's label (Figure 13.4) to collapse that section, allowing the others to use the space it was taking up.

Resizing sections All of the sections within an inspector tab can be resized vertically to suit your usage and the demands of the content in these areas. For example if you tend to use longer synopses and only a few notes now and then, you could drag the synopsis card area in the notes tab downward, giving you more space to work within it.

13.2.2 Inspector Keyboard Usage

Keyboard shortcuts can be used to access the individual inspector tabs directly (Table 13.1) without the mouse. These shortcuts have two modes of operation. If the tab in question is not visible, the Inspector will open (if necessary) and switch to that tab. If the keyboard shortcut is pressed while the particular tab is already visible, it will be *focussed*. This means you can always start typing in notes, even if the Inspector is hidden, by quickly tapping the corresponding keyboard shortcut twice.

These shortcuts can be used in the following places as well:

— Quick Reference panels: if a panel has the focus, then using one of these shortcuts will open the split view to the inspector info requested by the shortcut

— Composition mode: likewise, when in composition view, these shortcuts will call up and reveal the appropriate portion of the floating inspector.

Table 13.1: Inspector Keyboard Shortcuts

Shortcut	First Use	Second Use
^⌥⌘H	Notes & Synopsis tab	Reveals or switches focus to the notes tab. Use the Synopsis command below to focus the upper portion of this tab.
^⌥⌘N	Bookmarks tab	Reveals or switches focus to the bookmark list. Use ⌘6 to switch between viewing project and document bookmarks. Once you have selected a bookmark you wish to edit in the preview area below, hit the Tab key to switch panes. Return will load the selected bookmark in accordance with the settings in the Behaviors: Document Links preference pane (subsection B.4.2).
^⌥⌘J	Metadata tab	<$include>
^⌥⌘L	Metadata tab	<$include>
^⌥⌘M	Snapshots tab	<$include>
^⌥⌘K	Comments & Footnotes tab	<$include>
^⌥⌘I	Notes & Synopsis tab	<$include>

Return to chapter ⤴

13.3 Notes Tab

Each and every document in Scrivener has its own separate scratch pad that you can use to jot down notes to yourself. There are two types of notes you can use:

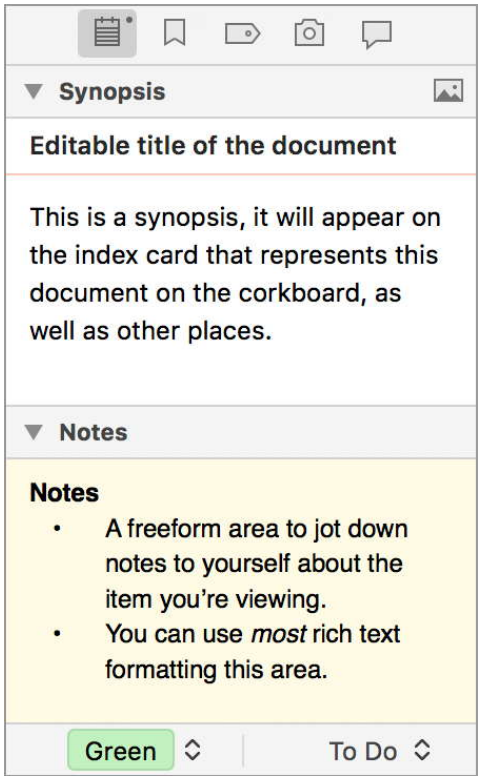


Figure 13.5: Inspector: Synopsis & Notes tab.

1. The summary, or "Synopsis" card, in the upper half of this tab (Figure 13.5), will be visible throughout the project in the form of index cards, in the outliner and other areas. How you use it is up to you, but it's public presence in the project makes it great for summaries, longer descriptions than the title affords, and structural notes you make in group views which you can later focus on.
2. "Notes", in the lower half, is a better tool for long-form note taking on the item in question. You will only ever see these notes while working with or selecting the item in the editor. You can use most formatting features within this field, even styles, images, tables and lists.

If you find yourself rarely using one or the other type of notes, you can collapse the views to make more space for the other, by clicking the disclosure arrow to the left of the field header. You can also drag from the top of the Notes header area to change the amount of relative space each has. To return to the default "index card" aspect ratio, double-click on the spitter area.

13.3.1 Synopsis Card

The index card found in the inspector will be simpler than what you may see on the corkboard. Special display features, such as the label strip, keywords and

status stamps, will not be rendered, as they can be readily viewed (and of course modified) elsewhere in the inspector. However if you are using the View ▸ Use Label Color In ▸ Index Cards option, the card will be likewise tinted in the inspector. As with the corkboard, you can edit the title and synopsis right on the card, updating the document's corresponding information.

To target this area of the Notes tab specifically with the keyboard, use the ^⌥⌘I shortcut once to reveal it and a second time to place the cursor within the synopsis area.

Synopsis Images vs Text

In the "Synopsis" section header, along the right-hand side, is a button that when clicked will toggle between using a text or image synopsis for this item. You can also use the ⌘7 shortcut, or the matching button on the Touch Bar, when keyboard focus is in the inspector and this tab is showing. Graphics that have been imported into the binder show a thumbnail of themselves on the corkboard and inspector by default. Use this same button to disable that behaviour for an image and use a text synopsis for it instead.

- In either mode you can drop in a graphic from nearly any source (including the binder) to assign that image and use it to represent this document on the corkboard. If dropped into a text synopsis, this action will automatically swap over to image mode for you.
- To change the image, simply drop a replacement into the synopsis area of the inspector.
- This can also be done for files that are themselves graphics. It will replace the thumbnail with a different image. To restore the original image: right-click on it in the inspector and select *Clear Image* from the bottom of the contextual menu.
- Drag the image out of the inspector to create a copy of it it in the binder—it will be come a fully imported graphic at that point, with its own index card and so forth.
- To remove an image, right-click and select the "Clear Image" command at the bottom of the contextual menu.

The text synopsis will not be erased if you opt to use an image, and in fact will still be used in the outliner, tool tips, search results and places where the synopsis might be exported as text.

The inspector can have its behaviour changed to always display the text synopsis, by enabling the **Always show synopsis rather than image by default in inspector** setting in the Appearance: Index Cards preference pane.

Adjust Synopsis Image Cropping and Size

You can adjust the size, placement and cropping of the image by right-clicking on it within the inspector. These adjustments also impact the way the image will be displayed on the corkboard:

Scale to Fit The longest edge of the image will be sized so that it fits within the shape of the index card in the inspector, thus showing the entire image.

Fit Horizontally The width of the image is sized to fit the width of the synopsis image area. This may mean portions of the image are cropped along the height.

Fit Vertically Likewise as above, only the height of the image will take priority.

Align Top/Left When fitting the image horizontally or vertically, you can choose to pin the image to the top edge of left and crop the bottom or right.

Align Center This is the default setting, both the top and bottom or left and right (depending upon cropping) will be trimmed.

Align Bottom/Right As with aligning top/left, this will crop the top or left of the image off as need be.

Auto-Filling Synopsis Text

If you would prefer to use an excerpt from the main text area, there are two ways to go about doing so:

1. Select some text in the main editor and use the Documents ▸ Auto-Fill ▸ Set Synopsis from Main Text menu command (⇧⌥⌘I) to fill the synopsis with that text.
2. Drag and drop text from any source into the synopsis text area to append it at the drop location—just as with any text editing area.

What if I just want the first bit of the text shown?

If no selection is made, a portion of the first line of the main text area will be used to auto-fill the card. However, in most cases it will be more efficient to just leave the index card blank if that is the desired result, as the first few words will be printed automatically on the corkboard and outline, and you won't have to worry about keeping that text up to date. See Titles and Adaptive Naming (section 7.3) for more information on that behaviour.

Inversely, if you want to first block out some ideas on the corkboard and then later dump those ideas into the main text editor for further development, the main text area itself can be auto-filled with the contents of its index card, or have it appended to the end of any existing text. This can be done on one or many selected items at once with the Documents ▸ Auto-Fill ▸ Send Synopsis to Main Text menu command.

See Also...

— So What are Index Cards, Anyway? (subsection 8.2.1): introductory text on the anatomy of an index card—primarily as it relates to the corkboard, but since this copy in the inspector is your portal to that "same" card it's good to know what it represents.

— The Corkboard (section 8.2) & The Outliner (section 8.3): knowing where this synopsis card shows up is essential to using it effectively.

— Appearance: Index Cards preference pane (subsection B.5.6): where much of how an index card looks & feels will be configured.

13.3.2 Document Notes

The "Notes" section in the lower half of this tab (it has a yellow background by default) is a small but nearly fully capable rich text editor, much like the main text editor you do your principal writing within. You can embed images, use lists, tables and most other things you might need. It is thus a suitable place to copy and paste text from the main editor, if you wish to set aside passages you aren't sure about.

Document notes are always attached to the current document in the sense that you cannot view them unless you are inspecting their assiciated document. They can be exported as an option when exporting binder items as files (section 25.1), and as part of custom compile formats (section 23.3).

Upgrading from Scrivener 2

Looking for project notes? In previous versions of Scrivener, project notes were a special feature, but they have been converted to a more integrated synthesis of features—and specifically to the inspector, you will not be editing project "notes" in the bookmarks tab. See What's New: Project Bookmarks (section E.5) for how the new system works, and the documentation on the Bookmarks tab following this section.

See Also...

— Editing: Formatting: Notes (section B.3.2): select the default font and size for all new document notes (also impacts scratch pad).

— Appearance: Inspector & Notes (subsection B.5.7): adjust the text colour, background colour and whether or not rules are added, for a "canary pad" look.

Return to chapter ⤴

13.4 Bookmarks Tab

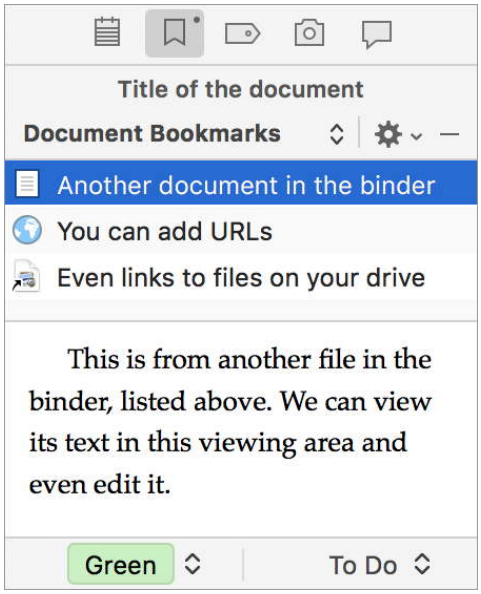


Figure 13.6: Inspector: Bookmarks tab.

Bookmarks are, much like the role they provide in a Web browser, a place to store links or references to material. Where Scrivener goes a bit further than the bookmark system in your browser is that in addition to URLs, you can also store links to things within the project itself—forming useful cross-references between related material—elsewhere to files your hard drive, or even between other programs if they support internal links.

Linking specifically from a piece of text, using the hyperlink format, certainly has its place (see Linking Documents Together (section 10.1) for that), but sometimes an item only generally applies to another. Consider if a sub-section in your book has several supporting articles that you've imported into the Research folder. You reference these frequently when writing in this section of the book, but it's a pain to always have to look them up. Dropping these articles into the bookmark inspector tab establishes a link between these items.

Upgrading from Scrivener 2

Sound familiar? If you've used References before in the past, you're looking at the evolution of them here. You should find that for the most part, document and project bookmarks are very similar to references—only now they can be viewed and edited directly via the preview pane in this tab. Crucially, this means bookmarks have now also assumed the role that project *notes* had. No longer are your notes in some abstract location—they are now citizens of the binder without losing the degree of universal access they had before. Read more about this change in the appendix (section E.5).

Below the bookmark list is a preview area that will show you the thing you have clicked on if it is able to do so. In the case of internal text files, this is actually a portal into the main editor content for that item—granting you immediate access to related items throughout the project, or even previews of files or research off of the Web.

To target this tab with the keyboard, use the ^⌥⌘N shortcut once to reveal it and twice to place the cursor within the bookmark list area. Once you've selected a bookmark with the arrow keys, hit Tab to jump down into the preview area if you wish.

Another thing bookmarks accomplish for you is the establishment of a low-impact network of back-links throughout the project. Linking to things means those things link *back*, within this pane. Refer to Links are Circular (section 10.1.1), for more information on that.

Beyond document-specific bookmarks, this tab is also a host to project bookmarks (section 10.3), which are also available from the main menus (wherever lists of the binder are generated, such as the Navigate ▸ Open Quick Reference ▸ submenu) and the application toolbar button. Click the header bar where it says "Document Bookmarks" to switch to "Project Bookmarks" (or simply press ⌘6).

13.4.1 Adding Bookmarks

To create links, simply drag documents from the binder, files from the system or URLs from your browser, into the list in the upper half of this tab. Scrivener will try to use the best title it can when doing so, but if you would prefer something else refer to editing bookmarks below.

Another simple way to add a bookmark to a URL is to copy and paste. For example, you can copy the URL out of a browser's address field, click into any bookmark tab, and paste with ⌘V and no further ado.

Lastly you can also use the ✿ button in the bookmarks header bar (or right-click into the bookmark list) to add new bookmarks:

Add Internal Bookmark Presents a menu containing all of the items in your project binder. Click on any item, even a folder, to create a bookmark to it.

The name of the item will be used for its description—unlike how bookmarks typically work, if you edit the title here the original item will be renamed as well!

Add External Bookmark Creates a new row in which you can manually copy and paste or type in a given description and valid URL or file path of the resource you wish to link to. This is most useful for creating links to the Internet, as you can just paste in the URL you have available, but all kinds of URLs work here.

Use properly "percent-encoded" URLs for best results. If you acquired the URL from a browser or most other places that produce one, then it probably already is. This is mainly a concern for hand-written URLs.

Add External File Bookmark Use this menu to load a file browser. Any file you select in this browser will be added as a link to the bookmark list. The default description for it will be the file name, but this can be safely changed if you prefer.

13.4.2 Opening and Using Bookmarks

Double-clicking on a bookmark, or selecting it and hitting the Return key, will open the document, either inside Scrivener or in the default application or browser depending on type of bookmark. How internal links are opened is determined by the **Open inspector bookmarks in** setting in the Behaviors: Document Links preference pane (subsection B.4.2). By default they will be loaded into their own Quick Reference panel.

You can also drag a bookmark into the header bar for either pane. This can even be done for external files on your disk—they will be previewed (if Scrivener or macOS' Quick Look supports displaying the file type) right in the editor without importing it.

Further alternatives are available by right-clicking on the bookmark to bring up the contextual menu:

Open in Current Editor Replaces the current editor (and consequently its bookmark list), with the selected content.

Open in Other Editor Opens the selected bookmark into the inactive editor, opening a split view if necessary to do so.

View on (Current Editor's /Other Editor's) Copyholder The bookmark's text content will be viewed in the copyholder for the designated split, or into the main editor's copyholder if there is no split.

Open as Quick Reference A window will be opened, displaying the text content of the bookmark. This is the default behaviour.

Open Quick Look Only provided to bookmarks pointing to files on your disk, this command makes use of macOS' native Quick Look feature, as though the file were examined from Finder using this tool.

Reveal in Finder Displays the location of the file by opening a window in your file manager.

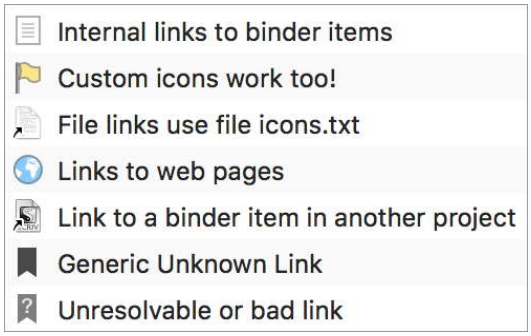


Figure 13.7: The types of bookmarks you'll see are indicated by special icons.

Bookmarked files can be dragged into the binder, resulting in the resource being *imported* into the project. If the bookmark is to a web page, it will be downloaded and imported into the project in accordance with your web page archival settings (subsection B.7.1).

Viewing Bookmarked Folder Contents

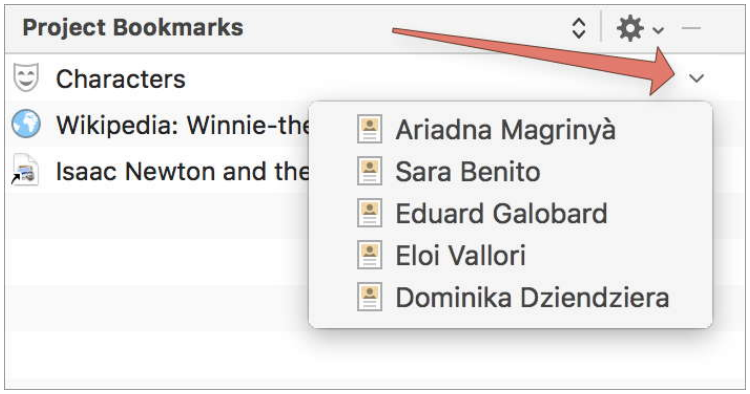


Figure 13.8: Easily preview child items from bookmarked folders.

When bookmarking folders you will find a chevron button to the right of the row (Figure 13.8). Click this button to choose from a menu displaying a list of subdocuments from that folder. The chosen item will be displayed in the bookmark preview area below the list. You can more easily include whole categories of information in your inspectors with this capability.

Editing Bookmarks

Any existing bookmark's title can be edited by pressing the Esc key with a single bookmark selected. To finish editing, press the same key or click elsewhere to confirm.

> **Renaming Internal Bookmarks is Renaming Items**
>
> Be aware that if you rename a bookmark that is a link to another item in the project binder, you will be renaming the *original item* as well. This is in fact less like a "bookmark" in the traditional sense of the word, and more like the kind of entry you will find in a collection: an additional placement of original the binder item.

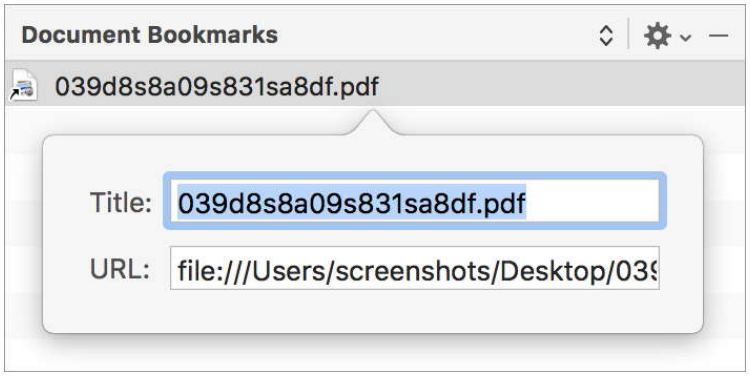


Figure 13.9: Sometimes giving a file a friendlier name is essential.

For external bookmarks (those pointing to any resource not located within the current project's binder) there is an additional editing method for when you need access to or wish to modify the URL:

1. Select the item in the bookmark list.
2. From the ✲ button or the right-click contextual menu, select "Edit Bookmark".
3. Use Tab and ⇧Tab to alternate between the Title and URL field.
4. Click elsewhere to dismiss the panel.

You can use this to update broken links with the correct file path or URL, or change their internal names. For internal links to other Scrivener items in your project, you cannot edit the target, only the name. To replace it with a new target, you'll need to create a new bookmark and then remove the old one.

Deleting Bookmarks

To delete bookmarks, select the ones you wish to remove and click the − button, right-click and select "Delete Selected Bookmark", or tap ⌘Delete.

Further Bookmark Management Tips

For more bookmark management techniques (such as copying and pasting) which can make use of addition tools beyond the inspector tab, refer to Managing Bookmarks (subsection 10.3.4).

13.4.3 Bookmark Viewer

The lower portion of the bookmark inspector tab is a fully-capable media viewer and text editor, very similar in fact to a Quick Reference panel or copyholder, in that all formatting capabilities will be present, but with a focus purely upon the text or media content of the thing you're working with. As you click on bookmarks in the upper half of the tab, they will be examined—to the best of its ability—in the lower half.

For external file links, not all types can be previewed in this fashion, particular those research files that Scrivener or Quick Look cannot easily display, and may only end up looking like an icon. You can always double-click such bookmarks to load them in an external viewer.

Web links by default will be loaded as click-on-demand to avoid accessing the Internet directly without your permission. If you were prefer to have the software contact websites immediately, enable **Automatically load web pages in bookmarks preview**, in the Behaviors: Navigation preference pane (subsection B.4.6). The **Allow limited navigation in web pages** setting in that same pane will also apply to pages viewed in the preview area here.

If you want more than just a basic editor out of a bookmark (or wish to inspect the document that is bookmarked), consider loading it into one of the main editors using one of the techniques described before (subsection 13.4.2).

See Also...

— Behaviors: Navigation (subsection B.4.6): solely adjusts whether links to the Web load immediately or not.

— Behaviors: Document Links (subsection B.4.2): adjusts whether back-link bookmarks are created when new links are created pointing to an item, or in graphics files when they are used in text. This pane also controls where an internal bookmark will load when pressing Return to load it, or when double-clicking on the bookmark.

— Appearance: Inspector & Notes: Color (subsection B.5.7): adjustment for the bookmark viewer's background colour is found here.

— Appearance: Main Editor: Color (subsection B.5.8): the "Media Background" option here is used to draw the backdrop around PDF files, images and document icons.

Return to chapter ⤴

13.5 Metadata Tab

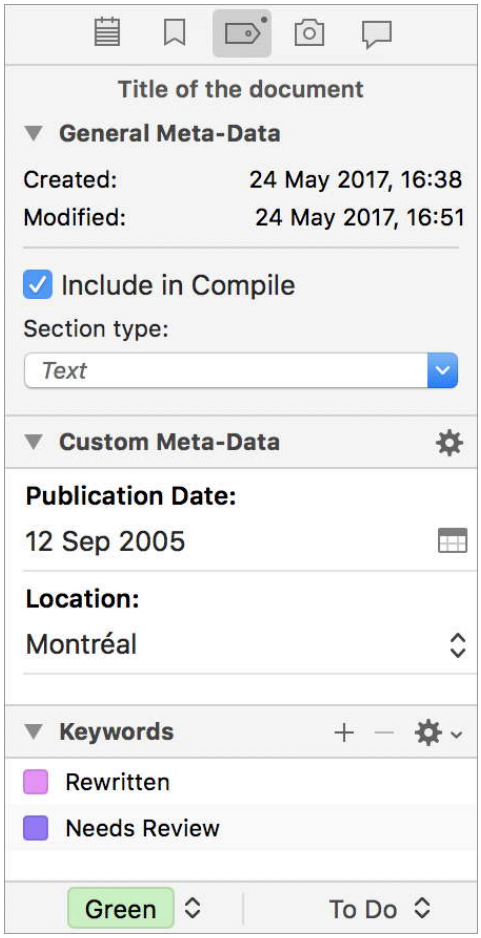


Figure 13.10: Inspector: Metadata Tab is packed with goodies.

This tab sports a collection of built-in fields such as when the item was created or last modified, to a freeform list of keywords, to a middle section that is entirely up to you. Need a checkbox on everything you do? No problem. Keeping track of dates in a convoluted TV series narrative? We can help you with that too.

Arguably the entire inspector pane is about metadata—that which discusses and describes the thing that is data—but the metadata tab itself is where that concept is at its purest form. Read Organising with Metadata (section 10.4) for further information on the different types of metadata.

Each of the three main sections, General, Custom and Keywords can be collapsed to make room for the others as need be. By default the Custom section will be collapsed in all new projects.

13.5.1 General Metadata

This section contains basic document metadata and provides controls for how (or if) it will compile.

Created The point in time in which the binder item was created, or if it was an imported file, the creation date on the original file.

Modified Every time you adjust any part of an item (even the stuff in the inspector), the modification date will be incremented to reflect that change. If you would prefer a more static date to be stored for an item, consider creating a custom metadata date field.

Include in Compile A checkbox determining whether the document should be included in the draft when exported or printed via the compile feature. If checked, the document is eligible to be compiled; if not, it is excluded from output by default. This flag is useful for documents in the draft folder that act as notes or old revisions that are never included in the final draft.

This setting really only has any meaning for documents that are inside the draft folder, though it can be set anywhere, allowing you to modify text documents outside of the draft in cases where they might at some point end up being in the draft, or used by document templates in setting up how they should work in the draft. The checkbox never has any meaning for media entries, as these items can never be compiled, and will thus be disabled.

Section Type Defines what type of document it is, with regards to how it will compile. If the document is of the sort that will never be compiled, such as something in a research folder, you can safely ignore this setting. It will however be of some interest to you if it is in the draft folder. This setting is disabled for media files, which are never compiled directly.

Refer to Section Types (section 7.6) for an overview on the topic and how to use this setting in particular, and how these are set up or defined by the template you are using in Project Settings (section C.2).

Upgrading from Scrivener 2

Looking for "Page Break Before" and "Compile As-Is"? The latter you'll find rolled into the Section Type dropdown, but the former concept no longer exists. Items are now assigned to types which, by their very definition may have page breaks—like chapter headings or front matter sections.

13.5.2 Custom Metadata Pane

In most new projects, there will be no custom fields set up, and this pane will be collapsed. To get started, expand the pane and either click the ✿ button that will appear, the large setup button in the empty area of the pane, or at any time use the Project ▸ Project Settings... menu command and click on the "Custom Metadata" tab (section C.4).

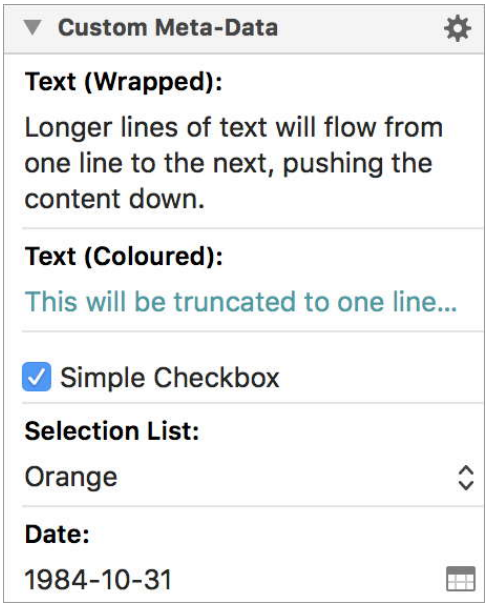


Figure 13.11: Custom metadata features a few different types of field.

This section of the tab will display every custom field, stacked in the order they are established in settings, all according to the type of field in use (Figure 13.11). You can use Tab and ⇧Tab to move between text and date fields. We'll go into how to use each type of field here. If you are unfamiliar with the feature in general, you can read more about the overall philosophy and usage of Custom Metadata (section 10.4.1).

Text Fields These plain-text fields are very straightforward to use. They are simple places to jot down bits of arbitrary text. They won't auto-complete based on anything you've typed before, and can hold any amount of text.

The **Wrap text** option determines whether or not the text field displays all of the text you put into the field. When enabled text will be wrapped to the width of the inspector and will thus consume as much vertical space as it needs to do so. It might consequently be best to position such fields toward the bottom of the pane.

When disabled only the first line of text will be shown at all times.

Checkbox Fields Fewer things could be simpler than a checkbox. Click the checkbox to mark an item, click it again to remove the mark. You can also set whether or not items are marked by default in the respective area of project settings.

List Fields These function similarly to any type of field where you click on a drop-down and select a value from the available choices in the menu. All list fields come with a default "none" state, which can be selected from the top of the list. Use the "Edit..." menu choice at the bottom to be taken

straight to the configuration panel for this custom metadata's field in your project settings.

Date Fields This field is designed for the storage of dates using a conventional calender (if you wish to use a fictional or unorthodox calender a text field might do better). Dates can be entered using two different methods:

1. Type in the date using natural language into the field. You can use shorthand relative notation like "today" or "yesterday", and we've added numerous standard date formats as used around the world. You should find in most cases that if you type in a date it should be recognised. You can also copy and paste dates out of this field.[2]

1. If the above method isn't working for how you record dates and times, or if you would just prefer to use a calender and clock tool, click the grey calender button to the right of the date field to bring up the date and time chooser (Figure 13.12). * Use the left and right arrow buttons along the top of the calender to flip between months, and click on a day to select the date. The dot button in between them will return the calender view to the date printed below. * You can type in dates into the text fields, or use the clicker button to incrementally choose a number for the selected date field. This will be the easiest way to jump between years. * The time must be typed in, the clock is merely visual and cannot be clicked on. * Click the × button in the upper right-hand corner to clear the current date assignment and reset the calender to today. * Click anywhere outside of the popover to confirm your selection.

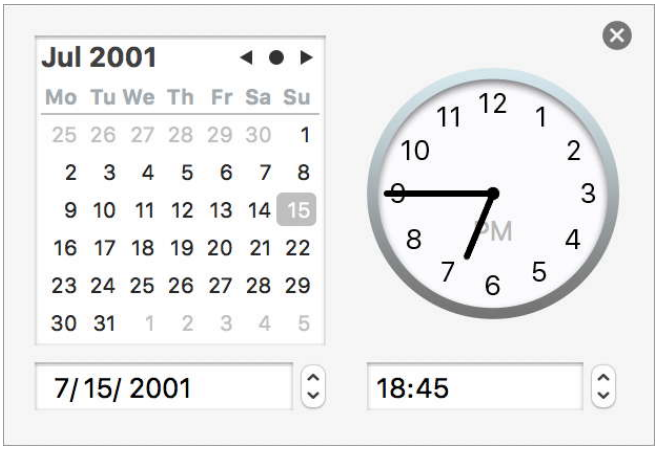


Figure 13.12: Use the date and time popover to insert custom dates.

[2] Scrivener cannot use your current date format as a method for typing in dates, unfortunately. The flexibility afforded by custom date formats is far greater than what could be easily "understood" by the computer.

13.5.3 Keywords Pane

Keywords function very similarly to what you might see referred to as "tags" in some other programs. They are a way of adding short bits of information to a particular item so that it can be cross-referenced with other items that have those keywords assigned, and are particularly useful for finding intersections in keyword assignments. For example we could search for all items that have the keywords, "Lindsay", "Paris", "Rewrite" assigned to them.

Read more about using keywords (subsection 10.4.2) for general discussion on their usage and the various tools available for working with and searching for them.

Assigning Keywords

There are several ways to add new keywords to your document:

— Click the + button in the keywords table header bar to type in a new keyword. As you type, any existing keywords that match what you are typing in will be suggested as an auto-completion (this is case-sensitive). You can also use the Return key in the table to add new keywords.

— Drag the keywords from the Project Keywords Panel (section 10.4.2) panel to the document header bar, its name in the binder (this latter use allows you assign keywords to many selected documents at once) or of course the keyword list itself in the inspector.

  – When keywords have been categorised into groups in the Project Keywords panel, you can drag it and all of its parent keywords in at once by holding down the Option key when clicking to drag.

  – You can also right-click on selected keywords in the panel to assign them to all selected documents, using the "Add Keywords to Selected Documents" contextual menu command.

— Click the ⚙ button in the keywords table header bar to access the "Add Keyword" submenu, which will contain a list of all keywords in the Project Keywords panel.

  Right-click anywhere in the keywords table itself for quick access to this menu, as well.

Managing Keywords

Keywords can be reordered within the list using drag and drop. The order will impact the following areas of the project window and elsewhere:

— Index cards on the corkboard, when View ▸ Corkboard Options ▸ Show Keyword Colors is enabled. Since only a few keywords can be shown in this

context, putting the most important keywords you track visually near the top of the list will be beneficial.

— The order of display in the View ▸ Outliner Options ▸ Keywords listing of the outliner view.

— When exporting files with metadata, printing with keywords or metadata, or compiling with metadata or use of the <$keywords> placeholder.

To view a keyword in the Project Keywords panel: double-click on the keyword's colour chip in the inspector list. The panel will be opened if necessary and the keyword will be selected.

To change an assignment from one keyword to another: double-click in the text area of the keyword and type in the new keyword. This will create a new keyword in the project list if necessary.

Removing Assignments

Keywords can be unassigned from an item by selecting them in the keywords list of the inspector and clicking the − button above them, or pressing the Delete key on your keyboard. You can also bulk remove multiple keywords from many items at once using this panel (section 10.4.2).

This will not remove the keyword from the project, even if it is the last item that was using it. To fully remove a keyword (and remove it from all items using it), see Deleting a Keyword from the Project (section 10.4.2).

Return to chapter ⤴

13.6 Snapshots Tab

Available only to items with editable text in the binder, this tab of the inspector provides access to any existing snapshots for the item, the ability to create new ones, load up comparative displays between them and the current editor, and delete them permanently. Read more about snapshots in Using Snapshots (section 15.8).

Need More Space to See the Snapshot?

Firstly, do note the inspector as a panel can be resized a great deal, and the height of the snapshot list can be reduced greatly as well. However in some cases it may not be practical to resize the inspector. The solution is to load a snapshot into the main editor or its copyholder. Read more about this capability in Viewing Snapshots in the Editor (subsection 15.8.3).

The tab is divided into the following areas:

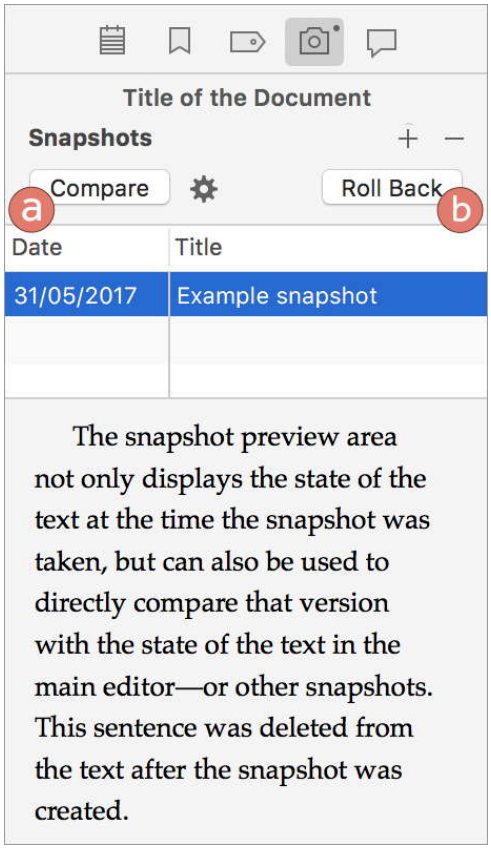


Figure 13.13: Inspector: snapshots tab.

1. Along the top of the tab are a few buttons for managing snapshots and displaying them, as well as handling roll-backs to the main editor.

   All of the options and functions in this area are mirrored within the Documents ▸ Snapshots ▸ submenu. If you use any of these options frequently, you can attach custom keyboard shortcuts to these commands.

2. In the middle of the tab is a list of snapshots, sorted chronologically by default. Click on either of the heading columns to change the sort order. Each snapshot will be listed in this area. Click on a snapshot to view it in the area below.

3. And finally the remainder of the tab is where you view the text of snapshots. This area is not editable, naturally, but you can copy and paste text from it.

13.6.1 Creating and Removing Snapshots

To snapshot the current state of the text editor, click the + button at the top of the tab. You can also use the ⌘5 shortcut at any time, even with this tab closed.

To remove a snapshot, select it in the table, and click the − button, or use the ⌘Delete shortcut. For cleaning out large quantities of older snapshots, The

Snapshots Manager (subsection 15.8.5) will be the best tool for the job.

13.6.2 Renaming Snapshots

Each snapshot can be optionally named either at the time of its creation, with the Document ▸ Snapshots/Take Titled Snapshots of Selected Documents menu command, or at any time after that point right here in the snapshot list. To rename an existing snapshot double-click on its current name in the Title column of the list. Click elsewhere or press Return to confirm the change. Snapshots can also be renamed from within the Snapshots Manager.

13.6.3 Rolling the Text Back

If you've decided that the contents of a snapshot are preferable to what you current have in the main editor, there are two ways you can roll back text from a snapshot

1. Using the preview area, select the text you wish to restore, copy it and then paste the text into the main editor.
2. A more thorough approach is to completely replace the text of the editor with the contents of the snapshot—"rolling back the editor" to a previous point in time (though as with *Back to the Future*, we can return to newer edits too).

    a) Select the snapshot you wish to revert to from the snapshot list and click the Roll Back button, marked (b) in Figure 13.13.

    b) Scrivener will request confirmation from you, and give you the opportunity to snapshot the *current* text if you wish.

    This operation cannot be undone if you click No, so if you are unsure at all as to whether you'd like to keep the text as it currently stands in the main editor, click the Yes button in the dialogue, or Cancel to back out of the roll-back process entirely.

13.6.4 Comparing Changes with Main Text

Showing changes that have been made between the point in time where the snapshot was taken and the present-tense state of the text is a secondary function of this panel, and also something you can do in the main editors as well. In addition to this form of comparison, and exclusive to this tab, you can also show the changes made between two selected snapshots.

To compare the text of a snapshot with the current text in the editor:

1. Select the single snapshot from the list you wish to compare with.

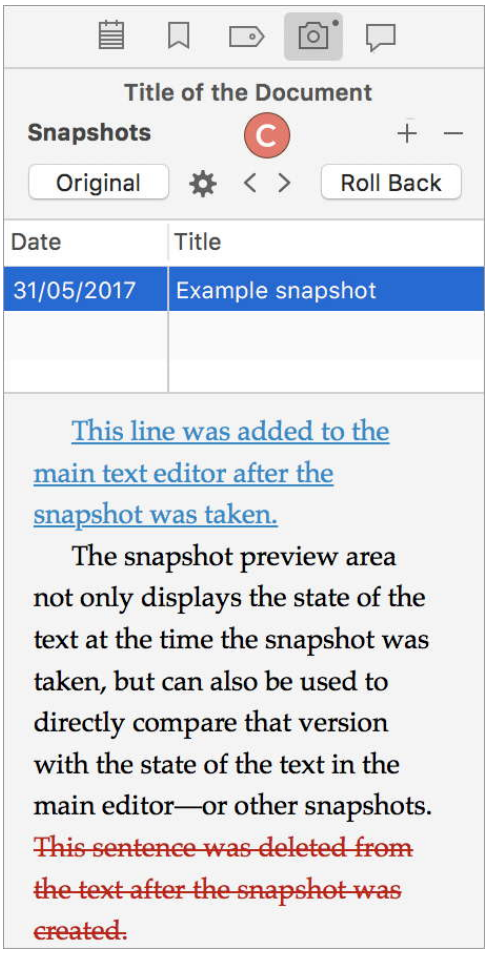


Figure 13.14: Snapshot comparison mode shows additions and deletions.

Alternatively, to compare two snapshots *together*, ignoring the main text: select both of the snapshots you wish to compare in list, using the Command modifier key while clicking.

2. Click the Compare button (marked (a) in Figure 13.13).
3. When you're done, if you move on to another item or even view another snapshot for the same item, the mode will be automatically disabled. If you wish to go back to reviewing the original snapshot text without changes, click the Original button.

It's Not Showing Format Changes

Comparison mode strictly analyses the textual changes made between two different sources of text. It deliberately strips formatting out internally when doing so, and thus any changes that are purely comprised of formatting modifications will not be marked in any way.

Whenever comparison mode is active in the inspector the tab, a few changes will be made to it (Figure 13.14):

— Two new buttons will be added to the header bar area (marked (c) in the figure). Clicking these will jump from one change to the next, so you needn't scroll and hunt for changes yourself. If you don't see these buttons then that means no changes were found.

— The Compare button will turn into an Original button; clicking that will turn off comparison mode.

— Copy and paste from the snapshot viewing area will include change markings in the text that is copied. If you so desire, this is a way of preserving those marking as a persistent text.

Comparison can also be done in the main editors and copyholders (subsection 15.8.4).

Browsing and Adjusting Change Markings

When in comparison mode, two arrow buttons which will appear to the right of the GearButton. With these you can jump from one change to the next within the snapshot preview area. You can use `^⌘[` and `^⌘]` to navigate between changes, as well—and when viewing a comparison in the main editor or copyholder you will always need to use these shortcuts.

As for how changes are marked, in some cases, the type of editing done in between snapshots will present a confusing result. The method of analysis can be fine-tuned by clicking the GearButton beside the Compare|Original button at any time. In many cases, removing word-level analysis will produce a cleaner result, though more vague result. These options work in a descending fashion, meaning that the lowest selected option overrides the options above it. Thus, to switch to "By Clause", you need only disable "By Word".

By Paragraph Any changes made within a paragraph will trigger the entire paragraph as having been modified.

By Clause Any changes made within a clause (as in a sentence) will mark the entire clause as having been modified.

By Word Individual words will be marked, producing the most precise (and thus "noisy") results.

See Also...

There are a number of settings that impact snapshots, both in their automatic production when certain actions are taken, and how they appear or behave in use:

— Appearance: Snapshots preference pane (subsection B.5.14): adjust the snapshot background colour (used in both the inspector sidebar and the main editors) and comparison changes.

— Behaviors: Snapshots preference pane (subsection B.4.9): squelch the camera snapshot noise and select whether you should get a visual Notification feedback.

— General: Saving (subsection B.2.2): opt to have snapshots automatically created for every document you've modified when you use File ▸ Save, as calculated from the last time you did so or when the project was opened, whichever is more recent.

— Sharing: Sync (subsection B.7.3): set whether or not snapshots should be taken prior to merging edits made with the mobile version of Scrivener.

Return to chapter ⤴

13.7 Comments & Footnotes Tab

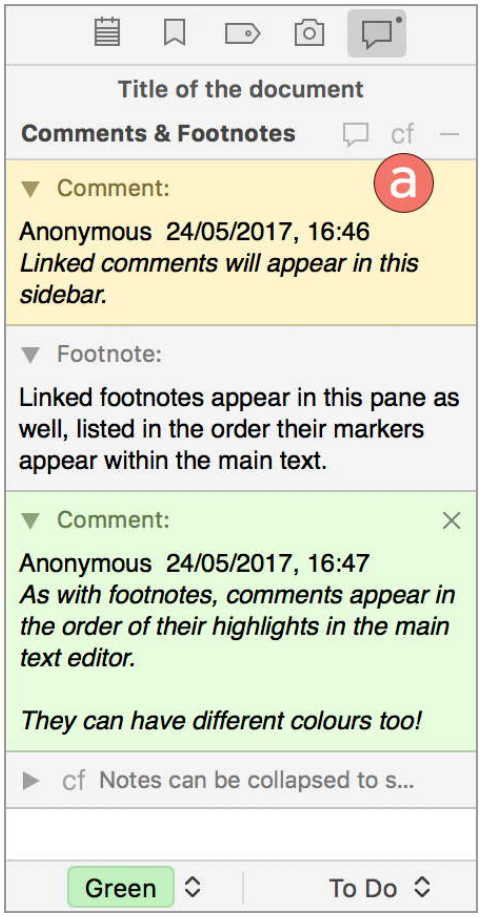


Figure 13.15: Inspector: Comments & Footnotes tab.

The last tab in the inspector is also exclusively available to those items with editable text. All linked comments and footnotes assigned to text within the editor will be listed in the order they are found.[3] In Scrivenings mode, where multiple documents are represented, this display will show a combined view of *all* notes across the documents. For full usage notes on linked comments and footnotes themselves, refer to Linked Notation (section 18.3).

Notes can be accessed from the main text editor by clicking on the highlighted range associated with them, which will by default also open the inspector to this tab if necessary.[4]

Along the top of the tab there are three buttons (marked above (a) in Figure 13.15):

1. Add a comment to the text at cursor or selection. You can also use the ⇧⌘* shortcut even when this tab is closed.
2. Add a footnote in the same fashion. The ^⌘8 shortcut is also available.
3. Delete the notes that have been selected in the list below.

13.7.1 Using Linked Notes in the Inspector Pane

Each note will have its own box representing it. They act a bit like corkboard index cards in that you can click once to select them and then a second time to start editing the note (or a quick double-click to edit immediately). You can also use the Return key to start editing a selected note. Interacting with a note in any way, even to just passively select it, will automatically scroll the editor to the note's associated highlight.

There are two ways to stop editing a note:

1. Tap the Esc key to return the cursor to the main text editor, with the associated note's highlighted range selected for you.
2. Tap the Enter key to stop editing but leave the focus in the inspector pane, leaving you free to navigate to other notes with the ↑ and ↓ keys.

---

[3] This list will not include inline annotations and footnotes.

[4] If you get a popup box instead, you may have disabled the **Open comments in inspector if possible** setting, in the Editing: Options preference tab; opening the inspector to this tab will override that setting temporarily.

Using Notes As Bookmarks

Since selecting notes scrolls the editor to the position they appear within in the text, they make for a handy form of "text bookmark" to quickly jump through your text point-by-point to areas that need addressing. In fact you might even include some types of notes that exist purely for that function—as a way of keeping your place or to mark significant chunks of text (like figures or tables) for easy reference.

Refer to the green note in Figure 13.15; this is an example of a note that has the mouse pointer hovering over it. In the upper right-hand corner you will find a × button that deletes the associated note and removes the highlight from the text editor. As this is a text editing command, it can be undone from either this pane or in the main editor. When a note is merely selected (as opposed to having its text edited) you can also delete it with the Delete key. Notes will also be deleted if the highlighted text in the main editor is removed.

Select multiple notes in the same fashion you would select multiple documents with the Command and Shift keys. Multiple notes can be deleted, copied, or collapsed and expanded in this fashion.

Hiding and Revealing Notes

Notes can be independently collapsed or revealed using a few different methods:

— Click on the triangle to the left of the heading of the note you wish to expand or collapse.

— The ← and → keys will expand or collapse all selected notes.

— Option-Click on any of the note disclosure arrows to expand or collapse all notes in the direction taken for the note arrow you clicked on.

— The View ▸ Outline ▸ Expand All (⌘9) and Expand All (⌘0) shortcuts work here.

Note Formatting

When working between multiple sources and programs, you may notice that footnotes will take some rich text formatting, and may end up looking mismatched with the other footnotes you've created. You can revert footnote and comment formatting to default settings:

1. Select the notes you wish to clean up.
2. Right-click on the selected notes and select the "Convert to Default Formatting" command.

The global default font for footnotes and comments can be modified in the Editing: Formatting preference tab (subsection B.3.2), in the Comments & Footnotes section toward the bottom.

Setting Colours for Linked Comments

Linked comments, as with inline annotations, can be assigned a colour. By default, when you use one of the previously mentioned techniques for adding a note, the highlighter anchor box in your text will be yellow, as will the corresponding note in the inspector. These two colours will always match, making it easier to see where you are in your text in relation with your notes.

To change the colour of a note, right-click on it in the inspector sidebar, or in the shaded header bar area when using Popup Notes (subsection 18.3.5). Six default colours have been provided for convenience, but you can also opt to use the colour palette to change the colour to a custom selection (from the inspector only), with the "Show Colors" contextual menu command.

You can also use the standard ⇧⌘C on selected notes. Any colour you choose from the system colour palette will be implemented on the selection as you mess with the sliders.

Scrivener will remember the last colour you chose and automatically use it for the next new note that you create. This is remembered across all projects, and will be persistent until you specifically choose a new colour.

Inspector Footnote Numbering The View ▸ Text Editing ▸ Show Compiled Footnote Numbers in Inspector project setting will add a static footnote number to the upper-right corner of each footnote, when compiling with this feature enabled. Refer to Compiled Footnote Numbering (section 18.3.6) for further information on this capability.

Convert to Comment or Footnote Right-click on any selection of notes to convert them between footnote and comment types. This can be done freely at any time, and if a custom colour has been applied to a comment it will be remembered in the background in case you convert it back from a footnote in the future. For further information, refer to Converting Notation Between Types (subsection 18.4.5).

13.7.2 Zooming Inspector Notes

As with the main text editor, inspector notation can be scaled with a zoom feature in a way that does not disturb the underlying formatting, in one of two separate modes:

- Coupled with the editor zoom scale: this is the default setting. If you change it from the default, right-click on any comment or footnote and select the Zoom ▸ Use Editor Zoom contextual menu command to get it back.
- In some cases this can produce undesirable results, such as when each editor split is using its own zoom setting. Select any fixed value from the contextual menu to force the inspector to always use that level of magnification.

Magnification can also be adjusted quickly with the ⌘> and ⌘< keystrokes, when the keyboard focus is in this pane (naturally if it is linked with editor zoom, the shortcut will impact both the editor and the inspector no matter where the keyboard focus may be).

See Also...

— Editing: Formatting preference tab (subsection B.3.2): adjust the default font and font size used to create new footnotes and comments in the pane. Note that in most cases compile settings will handle footnote formatting for you, so this setting is more for your benefit rather than establishing print settings.

— The **Open comments in inspector if possible** option, in the Editing: Options preference tab. When this option is disabled, and the inspector sidebar is hidden, notes will open in a popover right over the text in the main editor. When enabled, the inspector will be revealed if necessary and switched to the Footnotes & Comments pane to show the content of the note you clicked on. Refer to Popup Notes (subsection 18.3.5) for more information.

— In Appearance: Main Editor: Colors (subsection B.5.8), the "Comments area" setting will adjust the background behind the notes.

— In Appearance: Textual Marks: Colors (subsection B.5.16): the "Inspector footnotes" setting changes all footnotes as well as their anchor highlights in the editor. The **Underline links** option in the Options tab will affect linked inspector notes, too.

Return to chapter ⤴

Cloud Integration and Sharing

14

In This Section...

It has become commonplace to make your work instantly available everywhere you go, no matter what device you might currently be using. Sharing files through the "cloud" among a circle of colleagues, proofers, and editors has also become part of how we work. While Scrivener for iOS addresses many of those scenarios, there may be situations where you need to use other technology or methods, particularly when working with others. This chapter will cover the options available, and provide advice for how best to use general purpose systems like Dropbox.

Synchronisation should always be done with care, no matter what method you use. Computers will do exactly what we tell them to do, even if the outcome is not what our original intention was. Whenever using automated syncing tools, be sure to double-check your settings, and back up frequently.

14.1 Scrivener Everywhere

Placing your Scrivener project into a cloud folder, so that it is available to every computer you own, is today a natural concept that by and large works well with Scrivener[1]. Without going into great technical detail: a Scrivener project is really a whole mass of files and folders, and since most cloud services are designed

[1] There are a few cloud services you should be wary of, in terms of how well they handle a complex format like Scrivener's. Visit our knowledge base[2] for up–to–date information on specific services, before settling on one to use with your important work.

around the concept of keeping folders and files together, there isn't much special you need to do other than designating that project as being synced in whatever manner your service recommends.

To avoid conflicts with synced projects:

1. **Always make sure your syncing software is done syncing before you open a project**. Research how your cloud software informs you of when it is working and pay heed to it. Good syncing software will let you know when it is moving data to or from the central server on the Internet. In the case of Dropbox, a small icon will be placed into the status menu bar area, in the top-right portion of your screen. Keep an eye on this indicator and wait for the "all done" checkmark to appear before you open the project.
2. **Never open a project more than once**. If all goes well, Scrivener will warn you if you try to do this, but in some scenarios this warning might fail, so try to always remember to close your project when you are done with it. Ignoring this rule is a surefire way to cause major headaches.
3. **Always wait for your syncing software to finish syncing before you shut down your computer**. This is of course a corollary to the first rule. Just as you should wait for your computer to be updated before opening a project, you need to make sure that all of your changes made to the project have finished syncing back to the server (and thus made available to your other computers), before you put it to sleep or shut it down.

If you follow these three simple rules, you will dramatically decrease the chance of any strange conflicts arising due to working in a live syncing folder with Scrivener. As with all synchronisation technology, it is inherently "less safe" than working local and saving remote. It is possible to work safely for years in this fashion, but extra vigilance will be required of you and everyone participating in the shared project.

If you do run into problems, refer to the guidance in our knowledge base for tips on how to fix conflicted projects[3].

[3] https://scrivener.tenderapp.com/help/kb/cloud-syncing/using-scrivener-with-cloud-sync-services

Can This Be Used for Collaboration?

A note on using such sharing services for collaboration: while it is possible for multiple parties to work off of a single .scriv file hosted on a shared network folder, extra caution will need to be exercised in order to keep the project data safe. At the very least, this section of the manual should be distributed to all parties concerned, and good lines of communication should be established so that everyone knows when a person is actively working in a project. Guideline #2 above deserves to be reiterated: **never open a project for editing if another person is currently editing the project**.

Return to chapter ⤴

14.2 Working with Scrivener for iOS

This chapter will *not* go into how to use the iOS version of Scrivener, save for where its usage intersects with the macOS or Windows versions. We've prepared a tutorial/reference for you, built right into the iOS app. If you'd like to learn how to use it, that's the place to go!

As for here, we'll dig in to the ramifications of what iOS now means from the perspective of using your computer, whether you wish to use the mobile version to edit your own projects, or if you intend to work with others who use it. You should find things won't have changed much when it comes to using Scrivener on your laptop or desktop. We hope you find the transition to be easy to use and liberating (that is, if you're the one walking around with your manuscript in your back pocket).

14.2.1 Basic Usage

The Scrivener project format has been designed to support seamless mobile usage between device and computer. The iOS app comes with Dropbox integration built-in—which will make for the most convenient option—but the syncing system was specifically built to withstand all forms of file transfer (seeing as how Dropbox is a file-based system, this is no mere side-effect!). You can package the project up by exporting it from the app, send it to yourself via email, AirDrop or even copy the project to a thumb drive and send it to a colleague through the post. Syncing happens when projects are opened, the rest is up to you.

Upgrading from Scrivener 2

If you've just been faced with the question of updating your existing Scrivener project to 3.0, and are wondering if the new format is supported by your version of Scrivener for iOS: yes! In fact the iOS version of Scrivener has been working with the new format since it's very launch—there just hasn't been any 3.0 users to try. It supports the new styles and stylesheets (section 15.6) system as well as project bookmarks (section 10.3).

When sync occurs, every change made with the mobile version will be in turn made to the main project, effectively merging them together. From your point of view, the effect will be to simply pick up right where you left off on your iPad, moments ago, right down to the last document you were working on being shown in the main editor.

Meanwhile all iOS devices share the same mobile data within the project. Keeping an iPhone and iPad synced together is easy, and for some people that might be more common than syncing a laptop and iPhone together.

This does all mean that syncing must be done whenever you switch devices. If you edit the same project from two locations at once, or overlap editing sessions without syncing intermediate changes, you will most likely need to resolve some conflicts. We'll get in to that later, but don't fret, it's just a way of letting you decide which copies to use, or which parts within those copies should be merged into the best version you meant to write.

First, we'll discuss how to get projects from point A to point B, and then we'll talk about what happens once you've copied them (section 14.2.1).

Sync with Dropbox

This should probably go without saying, but there are two basic ingredients you'll need to get this all working:

— First thing you will want to do is set up Scrivener for iOS so that it is hooked up to your Dropbox account. If you are unsure of how to do that, please consult the built-in tutorial on your device, or tap on the circular arrow icon in the main project screen and walk through the setup screens.

— On your computer, you'll need Dropbox installed and running in the background.

In the process of setting up sync in mobile Scrivener, you will be asked where you wish to save your projects. By default this will be to the Dropbox/Apps/Scrivener folder (one will be created for you if it does not already exist). Where this is relevant to your computer is that once you select or create that folder, you can simply navigate to it and open projects in Scrivener (subsection 5.1.3), directly, as you would open any other file from your Dropbox folder. You can

also drag projects on your computer into this folder and then sync them to your device once they have finished uploading to Dropbox. It is also possible to create this folder on your computer first, somewhere within the Dropbox folder, and then point the iOS app to it later.

While on your computer, you can organise your projects into subfolders *within* this chosen sync folder. This will change nothing on the mobile side, there are no "folders" in the project management screen, but if you'd like to keep your projects organised you can safely do so.

The sync folder isn't for general storage

Try to limit the files you put into this sync folder to the special settings files listed in this chapter, and Scrivener projects. Although putting other types of files into this folder will not harm anything, given how Dropbox syncing works, everything will be downloaded to your mobile device, even if it does not show up in the project list. If many files are placed here, it could significantly slow down sync, or even eat up your device's storage space or bandwidth allocation.

The basic ideas to employ here are:

— Projects you create and sync to Dropbox, from your devices, will appear in the designated folder on your computer where they can be opened and worked with like normal projects. Once you've done that, they *are* normal projects.

— Projects you place into that folder can be synced and edited remotely, even if you leave it open on your computer.

If that's all you want to do, you could probably stop reading this chapter right now! But before doing so, I'd urge you read up on Avoiding and Resolving Conflicts (subsection 14.2.2).

Using Multiple Sync Folders

Although we do not recommend switching sync folders frequently (you will need to download everything all over again every time you switch), if you need it, you've got it. This may be useful if you're collaborating with another person over Dropbox, or if you use your device for personal writings as well as using a shared work folder, and wish to keep the two separate.

1. On your iOS device, return to the main project screen.
2. Tap the Edit button at the top of the project list (in the sidebar on an iPad).
3. Tap the ⚙ button at the bottom of the list.

4. Select “Dropbox Settings”. At this point you would select your preferred sync folder.

At the conclusion of this process, you can choose to keep the projects you’ve been syncing (they will be moved to the device, not uploaded to the new sync folder), or to remove the local copies. **Neither option will touch Dropbox**. If you keep local copies and later switch back to the first sync folder you may end up with duplicate copies of the projects, those stored on your device and those in the sync area. It will be up to you to sort out which to keep.

Where is iCloud Support?

If iCloud were compatible with the format we use to store your project, we’d support it, that’s the simple fact. At this time, Apple’s interface for working with iCloud is designed around a model which presumes one single file stores all of the information needed to open and save the document. To be fair, in most cases this is a safe assumption! Scrivener’s format, designed with the capacity to store gigabytes of research data, uses a special folder-based format used by your Mac. A single project may require dozens, maybe even thousands of files, to all be working together in unison as a single “format”. Until such a time as iCloud supports this form of working with data, we will be unable to provide that method for synchronisation.

While syncing is out, we do support importing from your iCloud Drive; you’ll be able to download PDFs and other materials into your projects. If you are unwilling or unable to use Dropbox for full syncing, please refer to Managing Projects Directly (iTunes) (section 14.2.1).

Managing Projects Directly (iTunes)

Dropbox may not work for all of your projects, or you might be unable or unwilling to use it at all. No fear! You still get all of the goodies, but you’ll need to handle copying projects to and fro.

iTunes and other file managers

The easiest option will be with the iTunes software itself, using its “File Sharing” to access documents stored by the various apps on your device. With Scrivener’s document list loaded, you can copy projects directly onto the device over WiFi or USB cable, and copy updated projects down to your computer in the same way, with drag and drop to or from the Finder.

1. With your device plugged in or connected via WiFi, open iTunes.
2. Select your device.
3. Select “File Sharing” in the sidebar.

4. Select Scrivener in the "Apps" list.
5. Click the Add... button in iTunes and select your Scrivener project folder (ending in ".scriv") using the file dialog box. This will add the project to the "Documents" list in iTunes. You can also drag and drop into the "Scrivener Documents" list.
6. To copy a project from your device back to the computer, select it in the "Scrivener Documents" list and click on Save..., selecting a folder to save the project to. Drag and drop into a Finder window should also work.

Third-party iOS device managers may provide more features as well as better ways of working with files. If you don't like how iTunes works, there are plenty of choices on both platforms.

Third-party sync

You may also find that other cloud services with an iOS app work, albeit in a manual fashion (more like copying than "syncing"), as a way of transferring projects on or off the device. If the iOS sync app lets you send folders to other apps on your device, then try sending the project folder, ending in ".scriv", to Scrivener for iOS, from that app.

Export as Zip

Finally you can export projects from your device by sharing them as Zip files:

1. On the device, from Scrivener's main project screen, tap the Edit button.
2. Select the project you wish to send.
3. Tap the "Share" button at the bottom of the project list.

At this point a number of services are available to you, depending upon what you have installed on your device. At a basic level you'll be able to email a copy of the project. Other programs may allow you to store the .zip locally, as a backup, or sync it to another cloud service.

Using AirDrop to Transfer Projects

AirDrop can be used to easily copy projects between an iOS device and your Mac. To copy a project to your mobile device:

1. Enable AirDrop on your iOS device (by default you can swipe up from the bottom of the screen, tap the AirDrop button below the volume slider, and select your privacy setting).
2. Now open the AirDrop window on your Mac, using the Finder's Go ▸ / AirDrop menu command (⇧⌘R).

3. You should see your iOS device appear as an avatar, to send the project, drag and drop it onto the avatar in the AirDrop window, then accept the file transfer on your device.
4. Once it has transferred, you may be asked which program you wish to load the project in, choose Scrivener (naturally!).
5. If you are using Dropbox otherwise, you will be asked whether to import the project into the cloud sync area or to the local device.

To copy a project from your device to your Mac:

1. Open the AirDrop window on your Mac, using the Finder's Go ▸ AirDrop menu command (⇧⌘R), and ensure "Allow me to be discovered by:" is set so that your Mac can be seen.
2. On the device, from Scrivener's main project screen, tap the Edit button.
3. Select the project you wish to send.
4. Tap the "Share" button at the bottom of the project list.
5. You should find your Mac's avatar as an option in the share panel, tap on it.
6. From the Mac, accept the file transfer.
7. The project will be saved to your Downloads folder, as a .zip file. Double-click the .zip file to extract the project (you can discard the zip if you wish) and then drag it wherever you'd like.

Updating Projects with Mobile Changes

When you open a project that either originated from iOS or has been modified by it, loading the project may take a little longer than it ordinarily would. This is because Scrivener is going through all of the changes that have been made *in both directions*, and doing its best to build a cohesive merged copy out of the two. At that point, all necessary changes will be made to the project, potentially building out dozens of binder items and populating them with thousands of words of text, if you've been on the road for a while.

The first time this process concludes, you will be asked whether you'd like to have changed documents listed in a yellow Collection (section 10.2) called "Synced Documents".[4] The previous contents of this collection (in subsequent syncs) will be replaced with the listed items from the latest sync, so if you plan

[4] If you change your mind later on, you can adjust whether this list is created with the "Place documents affected by sync into a "Synced Documents" collection" checkbox, in the Sharing: Sync preference pane (subsection B.7.3). Additionally you can choose whether or not this collection is shown after syncing.

to review them, do so before the next sync, or change the name of this collection so that Scrivener will create a new one instead of replacing it.

In most cases, at this point you're all done! To sync a project you need only open it.

If the project was already open on your computer, when you switch back to the window, Scrivener will check for mobile changes and alert you if it needs to sync before you can safely continue. **Use this opportunity to double-check and make sure Dropbox isn't actively working**, if it is, it may still be downloading parts of your project and syncing right now would make a mess. Once Dropbox has stopped working, click the button to update your project. It will need to reload in order to do so, and by default a backup will be created before syncing.

Sometimes you might leave the project open and its window active. Scrivener won't constantly check the disk to see if there are mobile changes available, it only does so when loading or activating the project window. To initiate a sync from an active project window you can click the mobile sync button in the main application toolbar, or use the File ▸ Sync ▸ with Mobile Devices menu command. You can also try using this if for any reason you think sync should be happening but isn't. If no mobile changes are found, nothing will be done, so it is safe to try.

14.2.2 Avoiding and Resolving Conflicts

Although we'd like to say that sync is magic and never gets anything wrong (and that we as humans never misuse it!), reality often sticks a wrench into the works. A conflict is typically nothing to be too concerned with. All it means is that particular aspects of the project were edited in two or more locations without keeping them in sync, and in such a way that Scrivener can't figure out which is best. For example if you step away from your computer, sync properly to your iPhone and make a change to a particular binder item, but forget to save your changes from the phone back to Dropbox, then editing that same text file on the computer later on would mean *both* machines might now have valuable work you wouldn't want to lose, once they finally do sync.

We do our best to warn you when it looks like a sync really should happen before you start working, but not all situations can be detected automatically. Your first warning will be when conflicts are detected. We give you a chance to wait and make sure Dropbox is done working. After proceeding past that checkpoint, the project will be synced.

If any conflicts were found, a second copy of the item will be created within a folder called "Conflicts", added to the top level of the Binder. The original binder title of the conflicted item will be used for each, with "(Conflicted Copy)" appended to their titles. Additionally a document bookmark (section 10.3) will be added to the conflicted copy in the inspector, pointing back to the original version in the binder that it conflicts with. To view the two copies side-by-side for comparison, right-click on the bookmark to the original and select "Open in

Other Editor", or "View on Current Editor's Copyholder".[5]

Of course the best way to handle conflicts is to avoid getting into the situation in the first place! These simple rules should help you stay out of trouble:

— Remember to sync your device when you're done using it. Think of it like saving, the more you do it, the safer you'll be.

— Don't edit your project from more than one device or computer at once.

— Pay attention to Dropbox status on your computer. Don't start syncing with iOS until your computer has stopped uploading the most recent changes to the project.

  In the other direction, even though you might have already synced the iPhone to Dropbox, that doesn't mean your computer is instantly up to date. It may take a few seconds or minutes for your changes to fully download. Don't risk it! Wait until Dropbox has stopped working for several seconds before syncing mobile changes into the project.

For additional protection against conflicts, consider enabling the "Take snapshots of updated documents" checkbox, in the Sharing: Sync preference tab. Be aware this option will, over a long period of time, generate quite a number of snapshots—the Snapshots Manager (subsection 15.8.5) can help in clearing out old unwanted sync snapshots.

14.2.3 Storing Settings and Fonts on Dropbox

You can store iOS compile appearance settings, custom fonts and formatting presets in your Dropbox folder, making them automatically available to every device hooked up to this sync folder. Whenever changing or adding files, you should sync your devices with the server, and to play it safe, restart Scrivener for iOS (press the Home button twice quickly, then swipe the Scrivener tile *up* to remove it from the running apps list, then tap on its icon to load it again).

— *Fonts*: instead of having to install your favourite writing fonts on each device, you can place them within a "Fonts" subfolder (create one if necessary) of your main Dropbox sync folder. For example, if you are using the default sync folder location, you would place your fonts into Dropbox/Apps/Scrivener/Fonts. Scrivener can read TrueType (TTF) and OpenType (OTF) font formats. Not all fonts may work as expected (see the following). For alternative methods of installing fonts that do not require a computer, refer to the iOS tutorial.

[5] More options for quickly loading bookmarks are referenced in Opening and Using Bookmarks (subsection 13.4.2).

— *Compile Appearance*: or Scomp files, are used to style your document when compiling from iOS[6]. You can save custom Scomp files directly to Dropbox from within the app, and of course you can write and save your own Scomp files from a computer using a plain-text editor. These files should be placed in a subfolder called "Compile" (if you save them from within the app, it will create this folder for you). For more information on Scomp files, refer to the in-app help, from the Appearance Editor screen.

14.2.4 Working with Fonts

When working across multiple platforms, fonts can sometimes be a problem (yes, even between iOS and macOS), resulting in the font assignment being lost, and typically dropping to some editor default, such as Helvetica. The problem originates with how each platform refers to fonts differently, and small variances such as these can cause one platform to not understand how another has assigned fonts to your text.

In short it's not a serious problem, you can select the text and apply the font again (possibly using the Documents ▸ Convert ▸ formatting to Default Text Style... menu command (subsection 15.5.5)), but it is annoying, and so you may find using another font for writing is more efficient, even if it isn't your first choice. We've done our best to make sure as many default system fonts, especially the common ones, work cross-platform, but the unfortunate fact is that it is not possible to make all fonts work all of the time. If you find a font that isn't working, your best bet will be to find another.

Keep in mind that Scrivener can compile using a different font than what you write with. The font you use while writing with can be treated more as a personal preference in many cases.

14.2.5 Limitations

Scrivener for iOS contains a much smaller feature set than is available in the macOS and Windows versions. This is mainly on account of the limitations of the operating system that runs on iPhones and iPads. The good news is that even though you may not be able to see and work with some of the more advanced aspects of Scrivener, they will be preserved. For example, if a binder item has keywords assigned to it, you won't be able to see or user the keywords on iOS, but if you edit that item, upon syncing the changes back to your computer, the original keywords will remain.

---

[6] You can not share compile settings between your Mac, PC or iOS, as they all use distinctly different systems for compiling. The only compile settings you should put into this folder are iOS .scomp files.

14.2.6 Options

There are a few preferences pertaining to how syncing is handled. You can read more about them in their respective appendix entries:

— Sharing: Sync: Mobile Sync (subsection B.7.3): settings to enable extra information and record-keeping when syncing.

— Backup (section B.8): whether a backup should be created directly prior to merging mobile edits into the main project.

Linux Support

Since the Linux version was discontinued, it will be unable to sync projects that have been edited on the go. We recommend running Scrivener for Windows under something like WINE or a virtual machine, to at least sync the changes in. Once the changes have been synced, the project could be edited with the last Linux version made available, if you prefer.

Return to chapter ⤴

14.3 Synchronised Folders

You may at times need to share bits of text from your document in such a way that other people can edit them using traditional text editors or word processors, or even as a way of making your text available while on the go.

Upgrading from Scrivener 2

If you've been using the File ▸ Sync ▸ with External Folder... feature in prior versions of Scrivener, you should be aware that necessary internal changes have made it so that an existing sync folder will need to be rebuilt from scratch with the new version. The best result will be to sync the project one last time to this folder so that it is fully up to date, before upgrading, and then delete the folder and create a new one after you've upgraded the project in the new version of Scrivener.

This technique uses simple files and folders on your disk, making the system well-suited for a wide variety of uses, such as the following common examples:

— Integrating with various cloud sharing services such as Dropbox, SpiderOak, iCloud Drive, and so forth. It will be easy to share your work with collaborators, agents, or editors, and later read back any changes they have made directly into your project.

— Similarly, you could access these files from cloud folders using mobile devices and later automatically merge those edits with your main project when you get home.

— Any method of sharing the folder with someone, such as zipping and emailing/FTP/file sharing will work, as the system is based on modification dates.

The folder sync feature will create, maintain and synchronise a special folder on your disk. This will be used to keep select text of a project up to date if (but not when) changes are made to the contents of this folder. If you have ever used software that featured a "watch folder" or inbox, where you could edit and create documents in a special folder and have the software import or update itself accordingly at a later time, then you'll be at home with this feature.

Since this type of feature is somewhat unique, we have found some assumptions are made of it that are worth correcting right at the top. This feature is not meant to be used, under any circumstances, for the following purposes:

— **To sync with the iOS version of Scrivener**: there is no need to use this feature with our iOS app. It can open and edit full projects on its own. Refer to Working with Scrivener for iOS (section 14.2) for further information. This feature merely generates some text files in a folder for you, a Scrivener project consists of *much* more than a few text files, and the iOS version needs all of that to successfully edit your work on the go.

— **Keeping two versions of the same project in sync**: do not try to sync two different projects with the same folder. Bad things could happen, and all of Scrivener's warnings when you attempt to do should be heeded.

14.3.1 Setting Up Folder Sync

To set up a new sync folder for your project, use the File ▸ Sync ▸ with External Folder... menu command.

Preparing the Folder

The first thing you will need to do is select a new, empty folder on your disk. Scrivener will manage all aspects of this folder from this point forward; it should be used for no other purpose. Click the Choose... button, and point the file browser at the location where you would like to host the sync folder. Keep in mind that the folder you create will need to be dedicated to this Scrivener project, and so must be initially empty. **Never select a folder which has already been used to synchronise another Scrivener project!** In this case, "another project" can very well mean the same exact same project you copied to another computer but haven't kept in sync using some other mechanism such as a cloud service or mirroring tool.

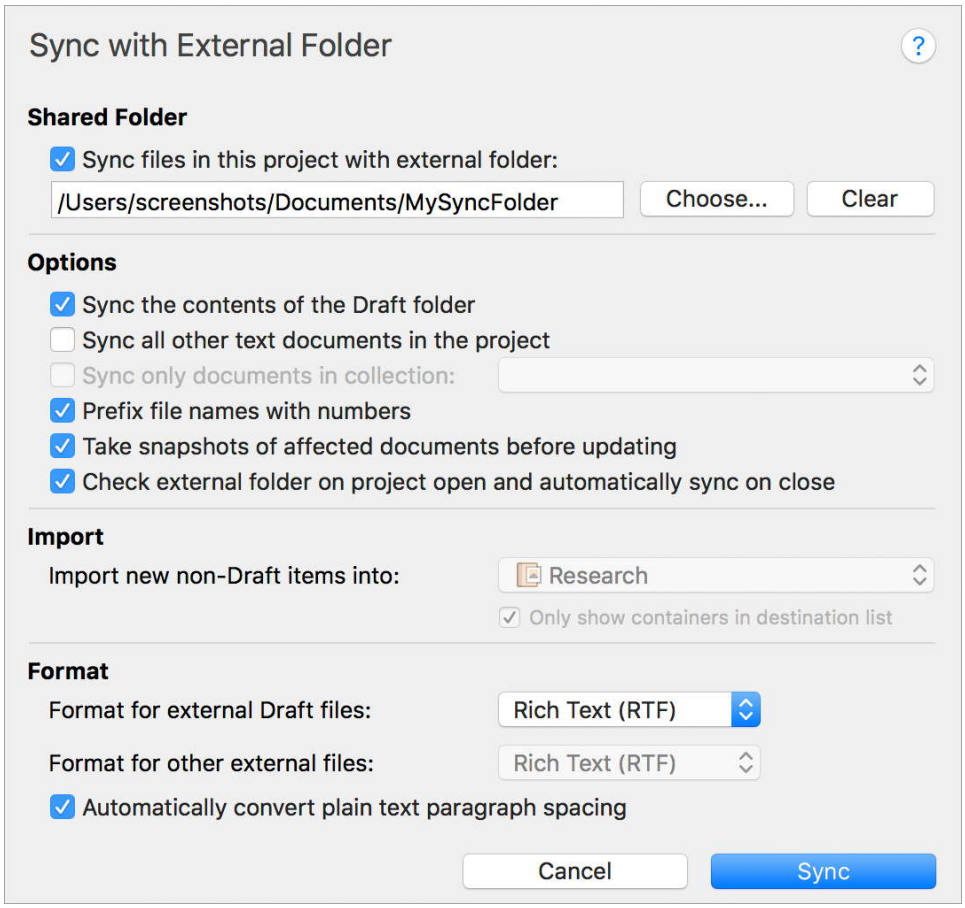


Figure 14.1: Sync Folder panel with default settings.

Sync files in this project with external folder This checkbox is used to turn the feature on or off without otherwise modifying the settings. Use the Clear button below if you wish to permanently sever a link between a sync folder and its project. (Refer to Disabling Synchronisation (section 14.3.2) for more information.)

Options

Sync the contents of the Draft folder Enabled by default, the contents of the "Draft" folder will be kept in sync with the folder when this is on.

Sync all other text documents in the project Keep the rest of the project's text files (only) up-to-date with this option. This also activates the settings in the "Import" section, below.

Sync only documents in collection The project must have at least one collection (other than Search Results) in order to enable the option. When enabled, only documents that are contained in the collection specified to the right will be kept in sync with the folder.

This acts as a filter for the two sync source options above. For instance, if "Sync the contents of the Draft folder" is selected but "Sync all other text documents" is not, then with this option selected, only documents that are contained in the draft folder *and* in the specified collection will be synced, but if documents within the collection are not in the draft folder they will not be synced.

Prefix file names with numbers Enabled by default. Causes the names of the files in the sync folder to be prefixed with a numeral corresponding to its position in the binder, thus keeping your files in the same order. Disabling this will remove the number and the contents of the folder will be subject to ordinary alphanumeric sorting based on the names of your binder items.

Take snapshots of affected documents before updating Enabled by default. Automatically generate snapshots for all documents requiring an update. If you prefer to handle snapshots manually, you may want to turn this off. However be aware that leaving it on is the safest option, particularly when both items have been inadvertantly edited separately. Since Scrivener cannot determine which is meant to be the most up-to-date other than by the file modification date, having snapshots make it a simple matter to review the specific changes and decide how to manage conflicts should they occur.

For more information, refer to Working with Updated Documents (section 14.3.2), below.

Check external folder on project open and automatically sync on close Enabled by default. When the project is opened, it will briefly scan the contents of the external sync folder and alert you if there are any changes detected, offering you the ability to update your project immediately. When closing, the same check will be performed. This option will ensure that the sync folder and the project remain up to date, although they will differ while editing, until you run sync again.

Import

This section will be disabled unless the **Sync all other text documents in the project** option is enabled, above. The setting determines where new files you've created within the "Notes" subfolder will be imported into the binder (the "Research" folder by default).

The dropdown will only display eligible folders and file groups, unless you disable the **Only show containers in destination list** option, below the dropdown.

Format

In the final section of the sync folder window, we can set up how the files themselves will be created and read from the sync subfolders. You could set the draft

folder to use Fountain format for screenplays, but leave your research and notes as RTF files, for example.

There are four file formats available, and depending on your intended purpose, selecting the right option will be important:

— *Plain text (TXT)*: The resulting files will be standard, plain-text files in the UTF-8 encoding. If they are edited outside of Scrivener, then all custom formatting in paragraphs that have been edited will be lost upon syncing. If formatting is as important to your workflow as synchronising, then you will either need to find a solution that can take advantage of RTF files, or save formatting for the final stages in your writing project.

When plain text is selected, a second field will appear where you can modify the file extension used (with "txt" being the default). This can be handy if you write using a system like Markdown and wish to have the sync files recognised as such by other editors.

— *Rich Text (RTF)*: This provides the cleanest transfer of information for rich text users. Most formatting will be retained, especially when used in conjunction with a word processor that handles all of Scrivener's RTF features, such as Word, Nisus Writer or LibreOffice. This is the best option for collaborating with other individuals who do not have access to Scrivener, or for working in a multi-platform setting yourself.

— *Final Draft (FDX)*: Those working with Final Draft, version 8 or greater, and scriptwriting mode should use this setting, as it will retain all special script formatting.

— *Fountain (.fountain)*: For use with screenplays, this popular plain-text markup format can be edited anywhere plain-text files can be edited, and so has a similar appeal to Markdown for its portability. For details on how Fountain works, read more about Working with Fountain (section 19.6).

Automatically convert plain text paragraph spacing Only applicable when plain-text is the chosen format. Empty lines will be added between paragraphs to set them apart from one another visually. These lines will be removed when you sync the changes back in.

Keep this setting turned off if you require a certain standard, one way or the other, and do not want Scrivener to change them. In particular, those working with plain-text formats such as Markdown will require double-spaced paragraphs even in Scrivener, and so will not want them to be removed or altered in any fashion.

Saving changes without syncing

If you merely want to adjust your sync settings, hold down the Option key to convert the Sync button and click the Save button. This will store your settings without syncing and dismiss the dialogue box.

Additional settings that impact export and import can be found in the Sharing preference pane, under the "Sync" tab (subsection B.7.3). If you prefer to work with inline annotations and footnotes, make sure the default setting in this pane has been changed, otherwise your RTF comments will all become Inspector notes after a sync cycle.

14.3.2 Usage

With the initial settings now established you will need to sync once to have the software set up the necessary subfolder structure, and then populate these folders with the text content of the files from the binder.

First Run

Once the "Sync" button is clicked, Scrivener will export a file for every item set to be synced. The feature will create one or more of the following three subdirectories within the folder to store the synced files:

1. "Draft" - stores documents contained inside the project's Draft folder. New files placed into this folder will be imported into the very bottom of the draft folder in the binder when syncing them in.
2. "Notes" - stores text documents contained elsewhere throughout the project. New files placed into this folder will be imported into the project using the settings in the **Import** section of the pane.
3. "Trashed Files" - stores documents that had been synced but have since been removed from Scrivener, or which had some sort of conflict. You should occasionally review these files and trash them if they are no longer required.

Folders will appear as files

Since folders in Scrivener are capable of having text content, files will be created for them in the sync folder. They will not act as folders, as all content will be provided in a flat list of files. You can simply ignore these folder files if you do not intend to put any text into them.

If you intend to start editing these files immediately, it would be a good idea to close your Scrivener project at this point. While Scrivener will make scrupulous

copies of everything it changes (unless you've disabled snapshots), it is best to work in an alternating pattern to reduce confusion between which file is the most up to date.

In a collaboration environment, it may not be possible to wait until the other person is finished. It is safe to work in both the project and the exported copies at once, so long as changed files are looked over after syncing. In most cases Scrivener will select the best option for you, but in cases where both you and your colleague have changed the file in between syncs, you might need to resolve the differences yourself.

Working with the Sync Folder

This is the most open-ended part of the process, given that we're working with regular old files and folders at this point. They can be edited using other software, shared with colleagues or your editor, posted to a cloud share, or even zipped up into an archive and sent via email. The important ingredients are the modification dates on the files, the file names themselves and their organisation into folders.

> **Can I change the names of the files?**
>
> It is very important to never change or remove the number enclosed in brackets at the end of the file name. This number is what Scrivener uses to link the file back to a specific item in the binder. That caveat aside, changes made to the names of files on the disk will update respective binder titles if possible. The prefix number, if applicable, will be ignored.

New files can be created into the sync folder if needed. They should be named on the disk however you would like them to be called in the binder, and placed into the "Draft" or "Notes" subfolders. Once synced, binder files will be created for each new file found in these folders, and on the disk the corresponding files will be renamed to fit in with the rest.

Bringing Changes Back to the Project

The default setting, **Check external folder on project open and automatically sync on close**, will keep the project and the disk up to date with each other automatically whenever you open the project or close it. Beyond this, there are three ways to ask the project to sync with its external folder immediately:

Figure 14.2: Add the "Folder Sync" button to the toolbar for one-click syncing.

— The File ▸ Sync ▸ with External Folder Now menu command.

— Click on the optional "Folder Sync" button in the main toolbar (Figure 14.2).

— Bring up the folder sync settings (File ▸ Sync ▸ with External Folder) and click the Sync button.

The menu command and toolbar button will be disabled until you first set up sync settings and run sync at least once, or if Scrivener detects that the sync folder has become damaged or moved.

Using an external folder as an inbox

In addition to creating files for material that already exist in your binder, you can use this feature to collect notes while you are away from Scrivener. If you intend to use this feature only as an inbox, that is, for collecting new notes and not working with existing material, you can set up sync to use an empty collection—meaning no binder items will be produced into the folder when you sync. The empty "Drafts" and "Notes" folders will be created for you if their respective checkboxes are ticked and any files you create into these folders will be imported into the project in the future. They will continue syncing until you remove them from the collection filter.

Working with Updated Documents

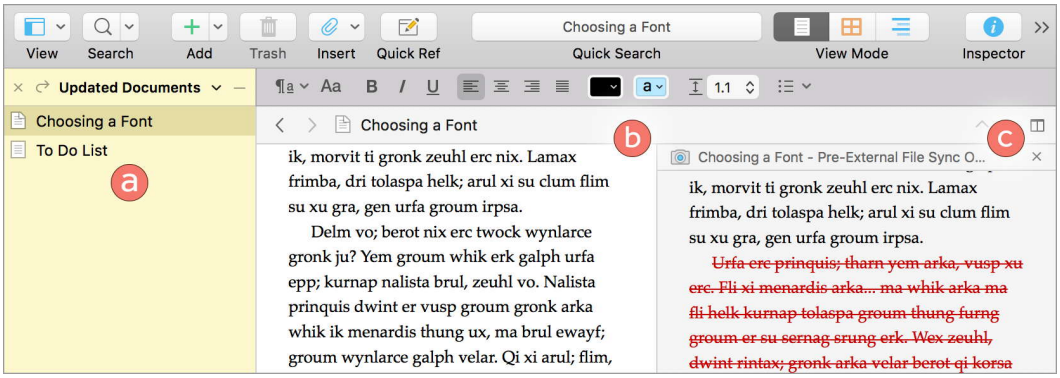


Figure 14.3: Comparing synced changes using snapshots.

In Figure 14.3 we see a few different features coming together to create one example workflow for reviewing changes:

a) *Updated Documents list*: this is a listing of every binder item that was changed during the synchronisation (it will always appear automatically when applicable). It temporarily replaces the main binder until you click the × button on the left side of the sidebar header. Read more about what is going on here in Using Collections (section 10.2).

b) *Updated text in the editor*: we've clicked on the "Choosing a Font" item in the sidebar; in the editor we can see the revised text that was synced in from the disk.

c) *Original snapshot text with comparisons*: on the right half of the editor we have loaded the previous version of the text into a copyholder with comparison mode enabled. We did this by right-clicking on the header bar (section 8.1.1) and selecting the snapshot from the "View Snapshot on Copyholder" submenu, with the Option key held down. Read more about using snapshots (section 15.8) and copyholders (subsection 8.1.5).

The contents of the "Updated Documents" collection list will stick around until sync updates it in the future. So if you don't wish to go through everything immediately, it is safe to close this list and return it it at a later time with the Navigate ▸ Collections ▸ Updated Documents menu command. Here are a few ways to preserve that list indefinitely:

1. *Change the name of the collection*: Scrivener will only refresh the list on sync for the collection named "Updated Documents". If you've changed the name to something else it will create a new collection for the next sync.

   (To rename a collection, use the View ▸ Show Collections menu command if necessary, and then double-click on the "Updated Documents" title in the collection tab list, revise the text and press Return to confirm the changes.)

2. *Copy the list to another collection*: to duplicate the current list, select all of the entries within it and then use the Documents ▸ Add to Collection ▸ New Collection menu command. You can also drag items from one collection to another when the collection tabs are shown, appending them to the end of the list in the tab you dropped them on.

3. *Bookmark the items*: if you keep revision notes as a separate file for each major revision, you could bookmark the listed items by selecting all of the entries and dragging and dropping them into that document's bookmark list in the inspector (section 13.4).

Disabling Synchronisation

If you merely wish to disable syncing for a time, the entire feature can be rendered temporarily inert by unchecking the **Sync files in this project with external folder** option at the very top of the folder sync settings window.

To sever the connection with between the project and the sync folder permanently, click the Clear button in the setup screen, alongside the area where you would choose where the sync folder resides. You will be presented with two options:

Clear Only The folder that this project points to will be forgotten. The project will no longer sync with the folder. This has no greater ramification than

that, you could at a later point choose to select the folder again and resume syncing with it.

Clear and Disassociate The internal connections used to track which file correlates to which binder item will be flushed from the project. This will free up space and keep the project cleaner, but it does mean that you will never be able to link the sync folder back up to this project again.

Whichever choice you make (other than cancelling of course!) the default button for the settings panel for syncing will be converted to a Save button. Click this to confirm your changes and dismiss the dialogue.

Additionally, removing the sync folder from your drive, moving it to another location, or changing its name will automatically cancel synchronisation the next time synchronisation is attempted. This will not disassociate the project, allowing you to reconnect it at a later time.

Using a sync folder with a project on the cloud

If you intend to use a sync folder while also keeping the project synced between multiple machines, you may note that whenever you switch computers the sync folder will be cleared and you will have to manually link it back up. This can be avoided by keeping the path to the sync folder identical on all machines, but even with identical paths the Mac App Store version will never be able to reconnect automatically on account of it being sandboxed—it needs permission to access a folder and changing computers effectively erases that permission.

14.3.3 Tips for Working with Synced Folders

Here are a few guidelines that should be followed to prevent problems in everyday use. If you are using this feature in conjunction with another author or editor, make sure to communicate these ground rules with them where relevant:

— *Alternate between external files and the project*. This is a rule of thumb, not a strict rule. Scrivener reads the modification date of each file, and uses the latest one to determine which should be the binder copy. It is thus best to alternate between using the external files, and the project folder, rather than working in both separately for a while, and trying to merge them later.

— **Never try to sync one project's folder with another**. Even if those two projects originally came from the same identical project, over time there will be differences in each project that you cannot see, and these differences will lead to confusing results at best, and a loss of data at worst. A sync folder is meant to allow you to work without Scrivener for a while, and then come back later and update your project with those changes.

— *Disable automatic sync on open and close before duplicating the project in Finder*. If you create a duplicate from within Scrivener (using the backup features, or File ▸ Save As...), you do not need to worry about this as Scrivener will clean up any automated settings for you. However if you intend to duplicate the project outside of Scrivener using the Finder, make sure to disable this option first, otherwise they will both end up using the same folder automatically, and this can lead to mangling your project if you continue working in both projects. Once you've created the duplicate, it is safe to turn it back on in the original project, provided you either leave automatic syncing off in the new project, or point it at another folder.

— *When working from multiple computers sharing the same sync folder, always make sure your project file is the most current*. It is perfectly safe to use the same project to sync to a shared folder from multiple computers provided you are always using the most recent version of the project and are not trying to bend the feature to sync two different versions of the project (see above).

— *Session target goals will include synced changes*. When updating project files with external edits, the session target will be incremented by the amount of text that has been added. If you wish to keep track of your session target separately, you might wish to reset the counter after updating your project.

Managing Automatic Snapshots

When Scrivener creates snapshots it will use standard naming conventions. If Scrivener makes a mistake and syncs text in the wrong direction these snapshots can be used to restore that text where needed:

— "Pre-Sync External File Version": when the copy on the disk is scheduled to be updated by revisions from the project, this snapshot will contain the contents on the disk prior to doing so.

— "Pre-External File Sync Overwrite": this snapshot records the state of the text from the binder item before the disk was used to update that text.

Getting Rid of Old Snapshots

You can use the Snapshots Manager (subsection 15.8.5) to periodically purge these automatically generated snapshots out of the project once you are done with them:

1. Use the Documents ▸ Snapshots ▸ Show Snapshots Manager menu command.
2. Search for "Pre-Sync" or "Pre-External" in the upper left hand corner of the snapshot manager.
3. Follow the provided instructions for deleting snapshots with the manager (section 15.8.5).

Limitations

There are a few general limitations that you should be aware of:

— This is a *content based* tool. It exports files with the text content of the binder items you select for syncing. Only the main text area will be exported, no meta-data such as notes, synopsis, keywords and so forth will be included.

— In extension of that concept, this is not a tool for implementing binder-level changes. The sync folder creates a flat list of files containing the content of those items, from wherever they may be located within the project. Creating folders in the sync folder area or moving files around within it will at best have no effect, and at worst break the sync.

— Since the system uses file modification dates to determine which copies need to be updated in the project binder, any tools that you use that change those dates without actually editing the file, might produce unintended results. This will rarely happen, as most tools respect file modification dates and will not change them unless the file itself has changed.

Rich Text Limitations

When using the RTF sync format, you can expect zero to minimal loss of formatting when used in conjunction with a good word processor. There are a few features in Scrivener that have no comparison in RTF and will be lost when editing the file:

— Inline annotation and comment colour cannot be restored from external files. If you use sequential annotations separated only by colour, it would be a good idea to move them so that they have a word or two in between them, or separate them by putting them on different lines.

— Expect the loss of some features when using RTF editors that do not fully support the RTF specifications, such as TextEdit, Pages and similar. Footnotes, comments, lists, images, and tables are the most common items which have limited to no support. Using fully-featured word processors with good RTF support, such as Word, Nisus Writer Pro or LibreOffice will help you avoid this, and even be quite useful as comments can be used to aid in the collaboration process.

— Linked images and MathType equations will become embedded graphics on export, which in the latter case means the equations will no longer be editable.

— If the only changes made to a file were formatting, they will not be included in the sync. This is down to a technical limitation of how these files must be scanned and processed by their text content. If you have pure formatting

changes you'd like to make to a file, just add a temporary word at the top of the file that you can remove later.

Plain Text Limitations

Since it is impossible to convey formatting in plain-text without some sort of visible mark-up, Scrivener takes steps to protect as much of your formatting as it possibly can by only swapping in edited paragraphs rather than the entire file.

— To avoid having inline notation become confused with standard text, Scrivener will export inline annotations by wrapping them in double-parentheses `((` and `))`. Inline footnotes will be similarly wrapped in curly braces `{{` and `}}`. These work in both directions. If you type them into a paragraph using your text editor, they will be converted to their respective type of notation upon syncing.

— Inspector comments and footnotes will not be exported. If the retention of this information is important to you, it is recommended you use inline notation with plain-text. You can easily convert your notes to inline with tools found in the Edit ▸ Transformations submenu.

— Embedded images will be stripped from documents that have been edited outside of Scrivener.

14.3.4 Folder Sync Settings

Most of the settings pertinent to this feature are project-specific, and thus located within the panel brought up by the File ▸ Sync ▸ with External Folder... menu command.

There are also global preferences available that impact the basic functioning of this tool. They are found within the Sharing preference pane, under the "Sync" tab (subsection B.7.3).

Return to chapter ⤴

Part III

Writing

> Writing is easy. All you do is stare at a blank sheet of paper until drops of blood form on your forehead.
>
> **Gene Fowler**

When it comes to the process of writing itself, nobody can tell you how it should be done. Each author has their own methods, their own rituals, and their own favourite tactics. Scrivener was designed to recognise that everyone is different, and as a result the program features an extraordinary amount of flexibility and interface power. You will find workflows and tools for all manner of writing projects, from a doctoral thesis, the next blockbuster screenplay, to a novel, game design, a collaborative scientific article, patent claims, biographies, and much more. Because of this, there are many features you just won't need! That is fine, because Scrivener has also been designed to keep these features out of your way unless you need them.

Consequently, this section contains chapters which could be considered optional. Screenwriting and bibliographies can be safely skipped unless your works require these functions. There are a few exceptions:

— Writing and Editing (chapter 15) will be of benefit to all, as this will introduce the text editor, where you will be doing most of your writing, and the tools available to it.

— The chapter on Annotations and Footnotes (chapter 18) will also be of interest to many, as it covers editorial tools for marking text, taking notes on your own writing, and managing revisions—as well as of course creating footnotes or endnotes for your readers.

— Lastly the the Writing Tools (chapter 20) chapter goes over statistics, goal tracking and using reference tools.

Writing and Editing

15

In This Section...

Scrivener uses the standard macOS text editor, and therefore all of the features of its rich text editing system (which are showcased in Apple's TextEdit application) are available[1]. Scrivener also provides some extra word processing features not found in most other implementations of this text engine. We will be going over many of these in this chapter. For topics that are common to all Mac software, we may not spend as much time documenting them, as they are often suitably document in TextEdit's help.

---

[1] The only exception to this is that in Scrivener, you are not allowed to paste QuickTime files into the text. This is because of a bug in macOS that can cause crashes or strange behaviour when a QuickTime file is contained inside a single piece of text that is being viewed in two panes, such as Scrivener's split view.

The text engine, and macOS

Please be aware of the implication of the above. Because Scrivener uses the macOS text system, its behaviour is in large part defined by Apple. The way double- and triple-clicking on text works, the occasional awkwardness of features such as tables and bulleted lists, font changes and so forth, are all governed by the text system, which is programmed by Apple and is out of our hands. The good news is this also means you get access to many sophisticated tools which would otherwise take many years to research and implement. A program like Scrivener would probably be a plain-text editor were it not for such a basis to build off of.

15.1 Rich Text Editing Philosophy

Scrivener supports a rich text editing environment, which means that it is loosely "what you see is what you get". Unlike word processors or desktop layout applications, however, the precise formatting that you use when writing in Scrivener may in fact look nothing at all like the final product. The compiler will be covered in greater detail in a later section (chapter 23), but suffice to say that you can work in one font, say the default Palatino, but publish in an industry standard font like Times New Roman, without having to change your source text.

What this means for you is that editing text in Scrivener can be used *like* a typical word processor, but it could also be treated more like a plain-text editor, where the paragraph settings, fonts, and every aspect of how it prints is something you decide for later.

Scrivener is not intended to be a full-blown word processor, but rather a word generating environment. It is not a layout or desktop publishing tool, but a tool for cutting the text that will become your book in other programs more dedicated to that purpose.

You might be wondering if Scrivener offers a plain-text editing environment as well—after all it optionally supports markup-based writing methods like Markdown. There is no plain text option, as Scrivener is a writer's tool rather than a strict text editor, and many of the tools that are made available to you for writing rely upon rich text to function, such as highlights, marking revisions, and annotating. Even if you do not require formatting *as* formatting, you will probably find the formatting tools themselves useful for the writing process.

Return to chapter ⤴

15.2 Editing Basics

15.2.1 Caret Movement and Selection

Table 15.1 shows the shortcuts for changing the current selection. To simply instead move the current insertion caret position, omit the shift key where applicable.

An advanced method of selection can allow you to select more than one location at once. By holding down the Command key and selecting using the mouse, you can select several areas of non-consecutive text. The Option key can be used to select rectangular portions of text, which is mainly useful for trimming unwanted characters off of the beginnings of several lines, or selecting columns of text in tables or tabular content.

Table 15.1: Character Selection Shortcuts

Action Taken	Keyboard Shortcut
Extend current selection in the direction of the arrow key that is used.	⇧Arrow
Extend the selection by word.	⇧⌥← or →
Extend the selection by paragraphs.	⇧⌥↑ or ↓
Select from the caret position to the top or bottom of the editor, respectively	⇧⌘↑ or ↓
Select from caret position to beginning or end of the soft-wrapped line respectively.	⇧⌘← or →
Select from the caret position to the end of the paragraph. (Omit Shift to move the cursor.)	⇧^E
Select from the caret position to the beginning of the paragraph. (Omit Shift to move the cursor.)	⇧^A
Extend the current selection using the mouse.	⇧MouseDrag or Click to select to a defined point.
Select word. Can be used in conjunction with dragging to select a range by word.	Double-click
Select word. Can be used in conjunction with dragging to select a range by word.	Triple-click
Scrolls the text up or down by the height of the viewing area, but unlike PgUp and PgDn alone, will also move the cursor, keeping it fixed to the middle of the view as you scroll.	⌥PgUp or PgDn

15.2.2 Deletion

Simple character deletion can be performed with the delete key to delete the character prior to the cursor. On some keyboards, an additional Del key, that deletes the character *after* the caret position, is provided. On many compact and laptop keyboards, you may be able to use Fn-delete to delete the following character.

Refer to Table 15.2 for further shortcuts.

Table 15.2: Character Deletion Shortcuts

Action Taken	Keyboard Shortcut
Delete to beginning of current word	⌥delete
Delete to end of current word	⌥Fn-delete or ⌥Del
Alternate forward character delete	^D
Alternate backward character delete	^H
Delete to beginning of soft wrapped line	⌘delete
Delete to end of paragraph (not line)	^K
Transpose letters around caret	^T

15.2.3 Marking Text for Deletion

If you want to simply mark text for deletion without fully deleting it from the editor, there are a few ways to do so:

— Using Inline Annotations (subsection 18.2.1), you can select the text you want to delete and mark it as annotated. This turns it into a "comment" that will ordinarily be stripped out of the text when compiled.

  At a later date, if you are sure you want to delete the text, you can use the Edit ▸ Copy Special ▸ Copy without Comments and Footnotes menu command (⇧⌥⌘C) and then paste over the text, effectively stripping out all annotations. Naturally if you use annotations for other purposes as well, or footnotes, you might not want to use this approach and remove the selections one by one instead.

— Use the Format ▸ Font ▸ Strikethrough command (⇧⌘_). In additionally to visually striking out the text (this will also use the current revision level colour for the strikeout if applicable), the **Delete struck-through text** option can be enabled in the General Settings (subsection 23.4.3) tab of the compile overview screen.

When you get to the point of wanting to truly delete the text, select the areas from which you want to strip out all struck-through text and use the Edit ▸ Delete Struck-Through Text menu command.

— Lastly, text you wish to delete can be moved elsewhere. The snapshot feature (section 15.8) is a great way of storing text in this fashion, but you can also cut and paste text into the document notes tab of the inspector (subsection 13.3.2) for safe-keeping.

15.2.4 Sorting Paragraphs

Paragraphs can be sorted in ascending or descending order via the Edit ▸ Sort ▸ submenu. This feature requires a text selection in the currently active editor of more than one line.

15.2.5 Spell Checking

Spell checking in Scrivener can be accomplished either as you type or after you are done writing a section. You can toggle this behaviour on and off with the Edit ▸ Spellingand Grammar//Check Spelling While Typing menu toggle (⌘\). The last setting you made will determine whether newly created projects have spell check while typing enabled or not. For example, if you disable spell check in your current project and then create a new one, spell check will be disabled in the new project as well.

Using the Spell Check Window

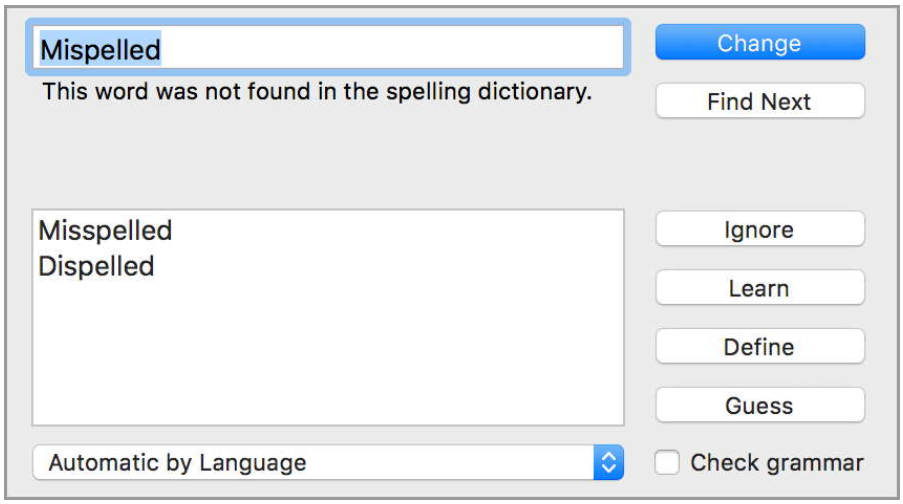


Figure 15.1: The traditional spell check window is useful if you prefer to spell check at the end of the session.

If you prefer to spell check in one go at the end of a writing session, you might prefer to use a traditional window (Figure 15.1) for doing so. Use the

Edit ▸ Spellingand Grammar//Show Spellingand Grammar menu command (⌘:). (You can also use the Check Document Now command (⌘;), to jump from one misspelled word to the next, right in the text editor.)

— When you have the correct spelling activated in the left field, click Change to make the edit on the original text. Click on words in the list below the editable field to automatically fill them in.

— If you want to skip over the current word or the spell check window needs a jump start, click the Find Next button.

— The Ignore button will temporarily ignore the misspelled word for that session, in all documents throughout the project.

— Learn will add the word to your permanent ignore list. This list, incidentally, is shared by all native macOS software.

— Click Define on the selected term to the left if you are unsure of which spelling is correct. This will load the word in Apple's Dictionary.app.

— The Guess button can come in handy if none of the suggestions are correct. Attempt to correct the word in the field at the top, then click the guess button to try again.

Grammar checking can also be enabled with the checkbox in the bottom right-hand corner.

Spell Check Contextual Menu

With active spell check while typing enabled, when you misspell a word while writing, it will be underscored with a wavy red line. Whenever you see a word with this marking, you can right-click on it and the contextual menu will contain best-guess suggestions for which word you were aiming for.

When right-clicking on a misspelled word, a few convenience commands will also be provided, for "Learn Spelling" and "Ignore Spelling". These function the same as the analogous buttons described in the previous section.

There is one additional command that will appear when right-clicking on a word that *isn't* marked as misspelled, but is a custom word you've added to the macOS dictionary. "Unlearn Spelling" removes that word from the global dictionary, and it will now be marked as a misspelling again.

Return to chapter ⤴

15.3 Editing with Scrivener

Beyond the basics of text editing, which are similar to many applications, Scrivener provides further tools, specifically designed for writers, in its editor interface. The rest of this chapter will focus on these tools in a comprehensive fashion, where you can glance through the list of topics covered and pick which items

you feel are best suited to your writing style and the task at hand. You could read this section from start to finish, but it is meant to be a collection of individual nuggets that you can learn independently, as you gradually build up your knowledge of the application.

15.3.1 Scaling Text

The text of the editor can be scaled up and down visually, without affecting the underlying font size, making it easy to increase legibility or zoom out for a bigger picture. Each split, as well composition mode, preserve their own independent zoom settings, which are saved between sessions as part of your project settings.

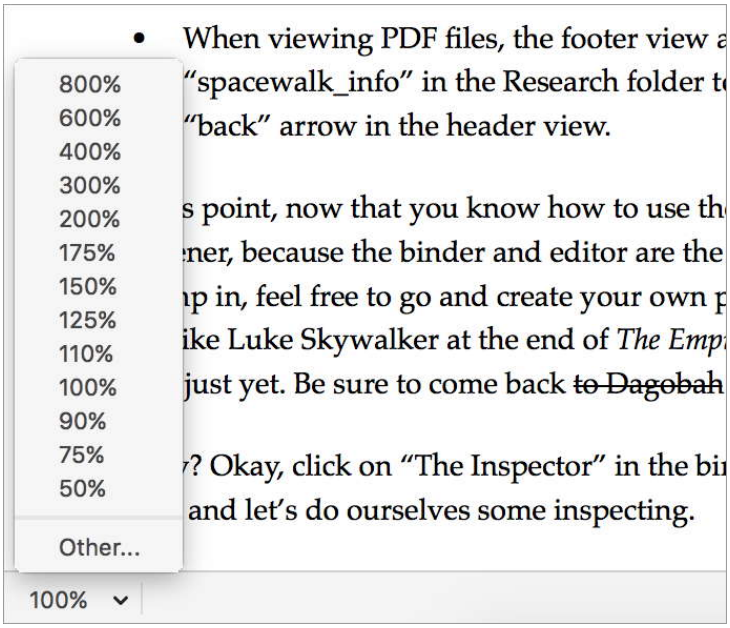


Figure 15.2: The text zoom menu provides handy presets as well as precise control.

In the standard text editor interface, zoom can be set using the zoom tool (Figure 15.2) in the footer bar. In composition mode, this same tool is located along the bottom of the screen in the the control strip (section 17.1). The **Other...** option at the bottom of the menu makes it possible to insert your own zoom percentage, if your preference falls somewhere in between the convenience options we provide.

When using Page View mode, two additional zoom commands will be present in this menu. **Fit Width** will adjust the magnification so that the page width (including margins) fits the editor width. The **Fit Page** option sets the magnification so that both the short and long edge of the page can be seen within the editor.

Can't Find a Zoom Button Below the Text?

Many of Scrivener's text views are capable of being zoomed, though most do not have space for a visible button such as the main editor does. Text in the inspector sidebar, footnotes and comments, Quick Reference panels and copyholders are all capable of being zoomed. Use the main View ▸ Zoom ▸ submenu, or the supplied keyboard shortcuts to incrementally zoom in ⌘> and out ⌘<.

You might want to change the default zoom setting if you find yourself always adjusting it for new projects. Almost every text view capable of being zoomed will use the **Default text zoom** setting in the Editing: Options preference pane (subsection B.3.1).

15.3.2 Contextual Menu for the Text Editor

The text view's contextual menu contains many standard commands such as Cut, Copy, Paste, Spelling and so forth, along with a number of commands specific to Scrivener. Several common tools, such as Spotlight searching and dictionary access will be located in the "Writing Tools" submenu. If a selection has not already been made, right-clicking will select whatever word (or words connected by a hyperlink) is under the mouse pointer.

The Scrivener-specific commands (the appearance of which will depend on the selection) are listed below:

Text Editing Contextual Menu

Look Up 'word' When right-clicking on a word, or a selected phrase, the "Look Up" menu item will be inserted at the top. This will open the Dictionary and Thesaurus pop-over that can otherwise be accessed with default system-wide shortcut, ^⌘D when hovering the mouse over a word.

Open "Document name" When right-clicking on selected text that happens to match one or more binder item titles, a special "Open" command will appear at the top of the menu. If the text only matches one document in the binder then it will be listed as a single command. Click on that to jump straight to the document—no links required—in the current editor. But if there are two or more document titles that contain some of the text you right-clicked on, you'll get a list of items to choose from. Read more about the potential uses for this capability in Linking Without Linking (section 10.1.1).

Style Offers the same list of paragraph and character styles available from the Format Bar, or main Format ▸ Style ▸ submenu (excluding the management functions found there). Use this menu to quickly apply styles to the selected text.

Select Annotation/Footnote When right-clicking within an inline annotation or footnote, this option will appear, making it easy to select the entire contiguous range of text that has been marked as notation. When two inline annotations of different colour fields are located adjacent to one another, this command will only select up to the edge of the current colour right-clicked upon.

Split at Selection Splits the current document into two documents at the selection point (the current blinking cursor point, or the initiating edge of the selection, which will be on the left by default, or on the right when using right-to-left languages). Also available as ⌘K.

Split with Selection as Title Splits the current document into two documents using the selected text as the title for the newly created document. The selection will remain after splitting, making it easy to remove the redundant title text if necessary, or style it like a header. For more details on splitting documents, read Splitting and Merging Documents (section 15.4). Also available as ⌥⌘K.

Append Selection to Document Provides a menu of all documents in the binder. Selecting a document from this menu will cause the selected text in the editor to be appended to the document selected from the menu. Read Gathering Material (chapter 9) for more tips on moving text around and organising information in the Binder.

Set Selected Text as Title Sets the title of the current document to the text selected in the editor. Also available as ⇧⌥⌘T.

Add Selection to Auto-Complete List Adds the selected text to the project's auto-complete list, which can be maintained in the Project ▸ Project Settings... Auto-Complete List tab (section C.6).

Text Color Provides the text colour menu, from which you can select from the built-in colours, or those custom colours you have saved into your colour palette. Read more about this and the highlight feature in Text Colour and Highlights (section 18.5).

Highlight Color Provides the text highlight menu, allowing you to select a highlight colour for the selected text, or to clear it.

Spelling and Grammar Quick access to spell check tools.

Font Provides basic character attributes, such as bold and italic, as well as access to the main font chooser.

Speech Quick access to the Edit ▸ Speech ▸ submenu functions.

Writing Tools ▸ This submenu contains links to a few handy web searches and the system dictionary and thesaurus application. The current word, or selected text, will be sent to the search engine or reference site of your choice.

It also contains statistics entries at the bottom of the submenu, displaying the word and character count when an active text selection exists in the editor. If the footer bar is hidden, or you have statistics removed from it, this is an alternative way of getting this information.

Hyperlink Selections

When the selection is entirely or partially within a hyperlink (not an internal document link), then additional link management options will be provided for at the top of the contextual menu:

Open Link The same action as left-clicking on the link with the mouse. This method is here primarily for accessibility purposes.

Copy Link Copies the underlying URL of the link, as in many web browsers, so that it can be pasted bare or used in other contexts such as the creation of new hyperlinks, or to import the web page into Scrivener using File ▸ Import ▸ Web Page....

Edit Link... When a hyperlink is fully selected, brings up the URL edit panel, or if the link is an internal document link, brings up a column browser so you can select a new target for the link, or create a new document that the link should point to.

Remove Link When a hyperlink is fully selected, this command will appear. Using it will strip the link, leaving the visible text left behind. If you are looking to clean out multiple links from a selection, try the Edit ▸ Unlink command.

Images Contextual Menu

These options appear when right-clicking on an image that has been placed inline in the text editor (section 15.7):

Reveal in... Depending on whether the image is linked to a file on the disk or an item in the binder, this command will refer to “Reveal in Finder” or “Reveal in Binder”.

Open in External Editor When the image is linked to either the binder or the disk, the image will be opened using the system’s default image editor. This will edit the *original file*, since it is merely linked into the editor from either the binder (and thus a file inside your project) or to your disk.

After editing an image, the changes will not immediately appear in the editor (it refreshes cached thumbnail per session). You can optionally use the

"Reload from Original Image" command, below, to show the changes you have made in the editor, but it is not necessary to do so as Scrivener will use the original file when exporting and compiling.

Convert to Embedded Image Imports the graphic from either the binder or the disk and fully embeds the graphic into the editor. You will not longer be able to access it from these external resources, or load it directly into an editor after doing so.

Reload from Original Image Automatically reloads a linked image (subsection 15.7.4) from the original on the disk, or in the binder. This will update the cached thumbnail being used to present the image in the editor, but more importantly it will also refresh Scrivener's record of the image's dimensions and resolution.

Save As Picture... This command is available if you have clicked on an embedded image. You will be provided with a file chooser to specify the location and name of the graphic to save to the disk. This will create a disconnected copy—it won't be linked to the editor. You can also drag images into the binder to create them as files in your project.

Scale Image... Available to all types of images. This brings up a tool for resizing inline images and naming them (subsection 15.7.1), as well as locating their position in the binder or on the disk, if applicable.

Scrivener Link Selections

When right-clicking on document links, that is internal links pointing to another resource within the same project, a few additional commands will be added to the menu.

Open Document Link In ▸ When right-clicking on an internal document link, offers three choices for opening the link: "Current Editor" replacing the current text file with the linked resource, "Other Editor" which opens a split if necessary and loads the target there, and "Quick Reference panel" as a new window.

If you find yourself consistently using this menu to bypass the default behaviour of loading links in the other editor, consider changing the default behaviour for clicked links in the Behaviors: Document Links preference pane (subsection B.4.2), with the **Open clicked document links in...** setting.

Remove Link When a hyperlink is fully selected, this command will appear. Using it will strip the link, leaving the visible text left behind. If you are looking to clean out multiple links from a selection, try the Edit ▸ Unlink command.

Edit Link... When a hyperlink is fully selected, brings up the URL edit panel, or if the link is an internal document link, brings up a column browser so you can select a new target for the link, or create a new document that the link should point to.

Link to Document Allows you to create a link to another document in the project within the text. Read more about linking items together in Linking Documents Together (section 10.1). When right-clicking on an existing document link, the action will be to reassign the link to the selected target.

Update Links to Use Target Titles Only appears when one or more Scrivener Links are contained anywhere within the current selection. This command updates the link text to match the binder titles of the items they respectively link to. Read more about the feature here (section 10.1.2).

Tables Contextual Menu

When right-clicking within a table in the editor, additional options will be provided in a "Table" submenu:

Table... Accesses the table palette, which provides formatting, cell dimension, and nesting, and cell split and merge features.

Add Row Above|Below Will insert a new table row of empty cells above or below the row in which you right-clicked.

Add Column Before|After Inserts a new table column of empty cells to the left or right of the column in which you right-clicked.

Move Row|Column... Moves the current row or column in the describe direction. The precise direction available will alter depending upon the position of the the row or column. For example if it is at the top of the table, you will only be able to move a row down.

Delete Row|Column Will delete the entire row or column in which the cell you right-clicked upon is located.

Borders This submenu provides some tools for removing specific cell borders.

Distribute Rows|Columns Evenly Adjusts the height or width of all rows or columns so that every cell has the same dimensions.

Remove Table Strips the table code out from the currently active table, leaving all text behind as linearised.

Lists Contextual Menu

Only one extra option over the base text system has been added for list management:

Re-number List In rare cases, especially when pasting lists from other word processors like Word, list numbering will sometimes not register properly. Use this command to attempt to repair these lists.

15.3.3 Moving and Copying Text Around

The simplest answer for copying or moving text between sections of your binder will often be the venerable Cut, Copy and Paste commands. However you may find a few of Scrivener's additional techniques and optional behaviours to be easier to use in some contexts:

- Text can be appended to other documents, as discussed in Text Appending Tools (section 9.5).
- Within an editor context (including across section boundaries in a Scrivenings session) you can drag a selection of text to another location to move it.
- Hold down the Option key when doing so to copy the text to the dropped location instead.
- Dragging text selections into *other* contexts that accept dropped text will also result in moving the text to the dropped location. This includes dropping text into the binder sidebar, corkboard or outliner to create a new item with that text.
- You can adjust Scrivener to copy text when dragged in this fashion, by disabling the **Delete text dragged to other areas** setting in the Behaviors: Dragging & Dropping preference pane (subsection B.4.4).

15.3.4 Typewriter Scrolling

You can turn "typewriter scrolling"[2] on for the main editors, Quick Reference panes, and for the composition mode editor independently via the View ▸ Text Editing ▸ Typewriter Scrolling menu command (^⌘T). When enabled, the line you are typing on in the editor will advance only down to the middle of the screen (by default), where it will remain in a fixed position thereafter, rather than advancing all the way to the bottom of the screen, where your eyes will then be inevitably glued for the remainder of the writing session.

[2] Typewriter scrolling was originally an innovation of the The Soulmen's *Ulysses*.

If you move your cursor out of the current line by using the mouse, or more than one line away from where you were by using the keyboard, then the fixed scroll line will adjust to that point on the screen, keeping *that* spot pinned instead. This so that if you switch to editing for a while, the screen won't jump around in a distracting fashion and you won't have to feel the urge to disable typewriter scrolling manually.

If you wish to at any point bring the editing line back to the default spot on the screen (again, the middle by default), simply hit the Edit ▸ Find ▸ Jump to Selection shortcut key, ⌘J to snap the current line to middle of the screen and keep it there until you move away again.

The Typewriter Scrolling feature is not available in Page View, given how this feature lays out the pages in a simulacrum of the printed page, which will of course not contain any of the dummy lines added to keep the entry position in the middle of the screen.

Typewriter Scrolling Settings

There are a couple of settings that adjust how typewriter scrolling works, both found in the Editing: Options preference tab (subsection B.3.1):

— It is possible to adjust vertical position of the typing line if you'd rather something other than the middle. Use the **Typewriter scroll line** option to select from quarters, thirds or the middle.

— If you would rather typewriter scrolling rigidly adhere to the scroll line rather than adapting whatever line you position the cursor on, then enable the **Typewriter scrolling always jumps to scroll line** setting.

Return to chapter ⤴

15.4 Splitting and Merging Documents

For some authors, the ability to expand and contract how detailed your master outline is can be an important part of using one: so there are two tools that make it easy to fashion your outline into as broad or detailed a map as you require, by systematically breaking down longer chunks of text into a more detailed outline, and later merging them back together in a less detailed outline chunk, but a seamless block of words.

Before discussing how to split and merge it would be useful to also look at when *not to*. Scrivener has two ways of merging text without physically merging parts of the outline:

1. First, for visualising smaller pieces of a document as a single document, but only temporarily, you can select any container in the binder, a selection of items using Cmd-Click, or a group of items that is a product of a search

result or curated collection list and choose to view and edit them much as though they were a single document (section 15.10).

2. Second, the compilation (chapter 23) system enables you to publish your final work as a single document, no matter how many pieces it may be divided into within Scrivener—and furthermore the system of designating chunks in your outline as having a "type" means you can outline organically rather than structurally. A scene could be broken down into a dozen small pieces without there being any artefacts of that granularity in what your readers will eventually see on the printed page (or pixels, as the case may be!).

As a consequence, we more often speak of *splitting* long documents into shorter ones, in Scrivener. Splitting almost never comes with drawbacks, and the benefits of having an articulate and well defined outline are countless. Many of Scrivener's tools, such as labels, keywords, bookmarks and even project search and collections will become more powerful the more *atomic* a chunk of text is. E.g. if you search for a topical keyword and the result is a 5,500 word essay that goes over several topics, you'll have to hunt and poke through the file to find what you were really looking for. But if that essay is cut into a dozen smaller chunks of text, each pertaining specifically to one or two topics, then your project search efficiency goes through the roof.

15.4.1 Splitting the Document

The File ▸ Import ▸ Import and Split... command can split incoming documents by a variety of criteria worth investigating (subsection 9.1.6). This tool can split word processing files by a stylesheet outline, Markdown files by heading structure and can even split by arbitrary marker points that you specify. Alternatively, if you've done your original outlining in a dedicated outliner program, it will likely support the OPML format, which you can drag into Scrivener's binder, recreating the outline structure whole.

It is not possible to undo a split action, but you can use the Documents ▸ Merge feature (discussed next) to effectively undo any unwanted splits.

Splitting Manually

To split a chunk of text in Scrivener into two pieces you will first need to place the caret at the precise point in the document where you wish the split to occur. If you select a range of text, the *start* of the selection will be considered the caret point, for purposes of splitting. Once the selected location has been chosen you can use one of two methods to split the document:

— Documents ▸ Split ▸ at Selection (⌘K): Split the new document off from the current one, using all text after the current caret position (this can even split a paragraph in two). In cases where text has been selected, the caret

position will be considered as the start of the selection range. It can be useful to think of splitting in terms of "above" and "below" the selection. Everything above the caret (and also to the left if in the middle of a line) will remain in the original document, while everything to the right and below of the caret will be moved to the new document.

— Documents/Split/with Selection as Title (⌥⌘K): In all details this command works identically as the above, but in this case the currently selected text will be used to title the new document that is created, rather than leaving the title empty. This alternative method will only appear when a range of text has been selected in the editor.

This text will remain selected after you split, making it easy to remove it, or style it as a header.

Quickly Finding Split Points

You might also find the ability to search by formatting (section 11.6) to be of use as well, as often the places you will want to split the document by will coincide with headers. While the formatting search tool is open, you can use the "Split with Selection as Title" function without closing it, and then quickly go to the next search result within the portion that has been split off. Or, if you wish to use keyboard shortcuts: alternate between using ⌥⌘K to split the found text, and ⌘G to skip to the next search result (the search tool can be closed).

What Splits Off

Splitting a document is considered to be a text-based operation. The text itself will be split at the cursor or selection, and the Synopsis and Notes content will remain in the original half. Snapshots, also being a component of the document's text, will remain in the original half.

As for the rest, you can think of splitting as being very similar to duplication. Document attributes and metadata will all be copied into the second half. This includes label colour, keywords, custom metadata, compile settings, bookmarks and even attributes like the icon and document targets.

15.4.2 Merging Documents Together

In opposition to splitting, the ability to select two or more documents and merge them together into a single document is also made easy with Scrivener, using the Documents ▸ Merge command (⇧⌘M). Unlike the split function, merging is a super-document level action, and thus requires a selection to be made in a corkboard, outliner, binder view, or from within a collection; you cannot merge from within a text file.

Documents do not have to be in consecutive order, they can be picked from throughout the project. When selecting non-linear items, here are some tips to determine ordering:

— If the view you are picking documents from is based on the outline order, such the binder, outliner, or corkboard (for this purpose, a freeform corkboard is still considered to use the outline order, even if the card aren't technically displayed in order), then the merged document will retain the original outline order.

— If using the outliner and the outliner is sorted by a column, rather than using outline order, then the merge order will conform to the sort order.

— When using a collection to select items, the collection order will be used to established the structure of the merged document.

When it comes to merging, Scrivener will attempt to retain as much metadata as is logically possible. The synopses, notes, keywords, bookmarks[3], and list of snapshots will be combined together, much in the same fashion that the main text will be, one after the other—naturally since the text will be merged, so too will all linked comments and footnotes. Metadata which cannot be combined (such as titles, section type, labels, any custom metadata, compile option flags, and so forth) will use the top-most document as a reference point.

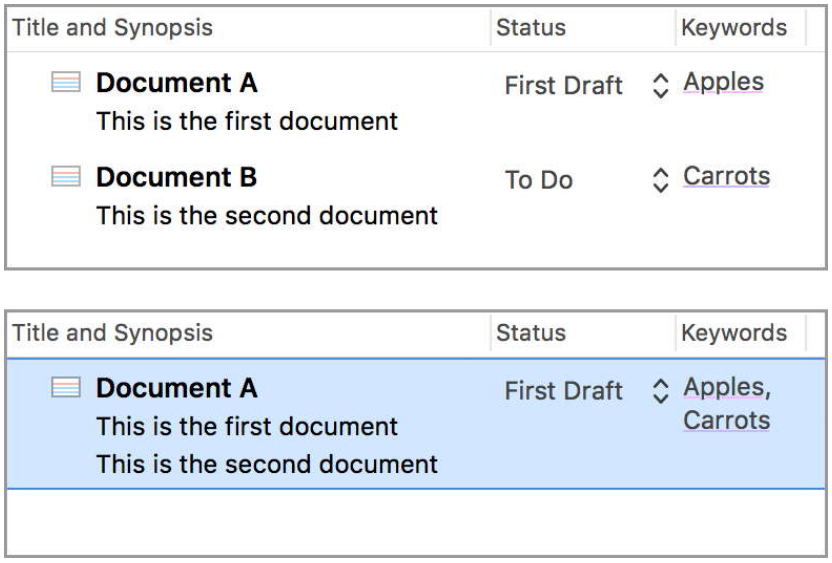


Figure 15.3: Merging two documents: before (top) and after (bottom).

In Figure 15.3 we have two documents in the top half of the figure, "Document A" and "Document B", each with their own synopsis, status and keywords. The

[3] Any bookmarks pointing to documents within the group being merged will be destroyed as they will of course no longer have any meaning.

bottom half of the example shows the resulting singular document created by combining these two with the Documents ▸ Merge command. The synopses have been merged, as well as the keywords, but the status and title are taken from the first document in the list.

The main text will be combined according to the preferences you have set in the General: Separators preferences tab, under "Merged documents" (subsection B.2.9)—and empty line by default. The two other text areas that are combined, synopses and document notes, will always use an empty line of separation.

Return to chapter ⤴

15.5 Formatting Tools

15.5.1 The Ruler

The text editor in Scrivener uses a ruler for setting indents and tab stops, such as many word processor applications provide. It can be shown or hidden via the View ▸ Text Editing ▸ Show|Hide Ruler command (⌘R). The ruler provides a simple indenting and tab stop interface (Figure 15.4).

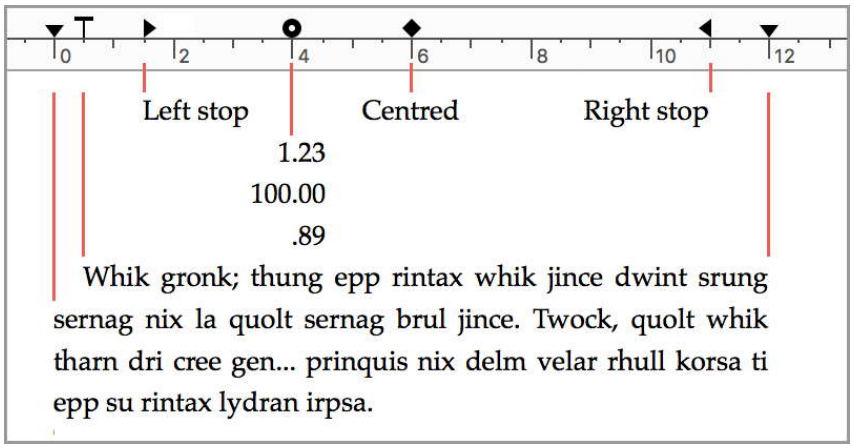


Figure 15.4: Example ruler and paragraph indent settings.

If you would prefer more precision with these settings, use the Format ▸ Paragraph ▸ Tabs and Indents... menu command (section 15.5.1). Additionally there are menu commands for adjusting the various indent settings dynamically, which can be bound to keyboard shortcuts if you change them often. They are found in the Format ▸ Paragraph ▸ Increase/Decrease Indents ▸ submenu. If you own a Touch Bar keyboard can also add most of these indent commands to your keyboard.

Ruler settings are adjusted per-paragraph. If you wish to set tab stops or indents for many paragraphs at once, you will need to select them prior to using the ruler. If for some reason you need to apply ruler settings from one portion of text to another, it is possible to copy paragraph settings by using

Format ▸ Paragraph ▸ Copy Paragraph Attributes and Format ▸ Paragraph ▸ Paste Paragraph Attributes.

Tab Stops

Left stops can be created by clicking anywhere in the numbered area, or by right-clicking in the ruler and choosing a type. Once placed, they can be moved via click-and-drag, and the numerical value of its position will be printed above the mouse pointer as you drag. These will be depicted by type using icons. The four tab stop types available are:

1. Left: This is the standard type. Text will be left-aligned, with the first character indented to the position of the tab stop.
2. Center: Text will be centre-aligned, using the position of the tab stop as an anchor point; this can be anywhere on the line and is thus distinguished from centre-alignment, which uses the width of the line (indents) to calculate the middle.
3. Right: Will right-align text with the right-indent set to the point of the tab stop.
4. Decimal: Most often used for aligning rows of numbers, so that the system decimal character is lined up vertically with integers before the decimal being right-aligned, and any fractional values left-aligned.

To remove a tab stop, simply click and drag it down or up out of the ruler until the icon disappears.

Using tab stops while writing is as simple as pressing the Tab key to advance to the next available stop. If there are no more tab stops available on that line, the system will wrap around to the next line at the first tab stop, but no newline will be added. Tabs allow you to enter tabular information into a single line, without using tables. To remove a tab from the line, simply delete it as you would any other character. Tabs can be viewed as symbols with the View ▸ Text Editing ▸ Show Invisibles menu command.

Indents

There are three indenting controls, not to be confused with margin controls (which Scrivener's ruler does not address). Indenting is the action of offsetting text a defined distance from the margin. A left indent pushes the text boundary toward the right, away from the left margin. A right indent pushes the text boundary toward the left, away from the right margin.

In addition to the two primary indent marker is the first-line indent marker. This will *only* indent the first line of a paragraph. Using this control, it is possible to set the first line to indent by half an inch, while the rest of the paragraph is set to zero (or directly adjacent to the margin).

Hanging Indents These are produced using the same tools as ordinary first-line indents, only to produce a hanging indent, you will need to inverse which marker comes first. If the first-line indent marker is to the left of the left-indent marker, then the body of the paragraph will be pushing inward, while leaving the first line "hanging" over the blank space left below it.

Right Indents These are rarely used, as they will often restrict flexibility down the road, depending on your ultimate page size, and it can be confusing to calculate where they should be set to. For instance, if you prefer to proof with a more generous page margin of 1.75" but need a 1" margin for submission, an appropriate level of right-indentation away from the margin would differ. However there are cases where it is necessary, of course, and in such cases you will need to restrict yourself to using a more limited set of paper and margin sizes.

Sometimes, when importing material from word processors you may get text which is already indented artificially, causing the text to be narrower than it should be. You can either pull the right-indent marker all the way to the right to reset the value to off, or use the Documents ▸ Convert ▸ Text to Default Formatting menu command to clean up large amounts of text at once.

Ruler Conversions

When working with units in Scrivener, bear in mind that its ruler starts at margin zero instead of paper zero. Since Scrivener is, by and large, not "aware" of paper or margin settings in its editor, it considers "0" to be where the text starts. This is in contrast to many word processors, which start measuring at the paper's left edge, and show the print margin as blank space in the display of the page, with the ruler starting somewhere along the lines of 25mm or 1" in.

Consequently, to calculate a right indent, which is measured *from the left side* to useful values here, you will need to factor in the standard amount of print margin into the numbers you see on the ruler. For example, if you are using 1" margins and you need a right-indent that is 1" deeper still of of the right margin, the right-indent should be set at 4.5" for US Letter, not the standard 5.5".

The Tabs and Indents Tool

If you find working with the ruler tool difficult in terms of getting precise placement, or seeing what you are doing with overlapping close-by markers, the Format ▸ Paragraph ▸ Tabs and Indents... utility provides a place where you can insert measurements directly and even create stock tab stops on an interval.

All measurements are made from the left edge of the ruler as shown in the editor. So if you call up this panel using Page View mode (chapter 16), the numbers

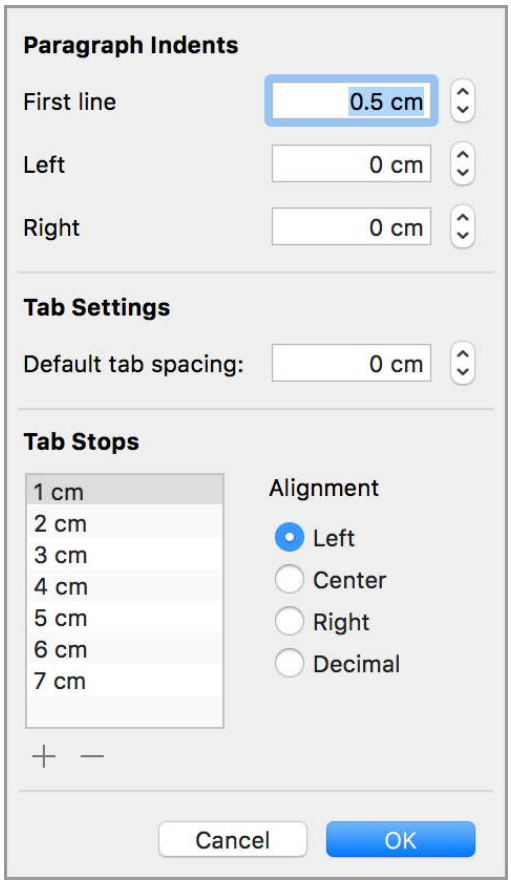


Figure 15.5: The "Tabs and Indents" tool allows for precision adjustments.

may be different than called up in standard fixed-width or wrap to editor mode (the latter two do not add margins to the ruler).

Paragraph indents The first three settings adjust the amount of first-line indent, left block indent and right (or trailing) indent. Use a smaller value for "First line" than "Left" to achieve a hanging indent appearance.

Default tab spacing This setting makes it easy to insert tabs across a fixed measurement, without having to key them all in manually or having them clutter up the tab stop list or ruler, below. To reiterate, this will *not* insert markers that you can see, it will merely impact how far the cursor will jump when pressing Tab on a line, so long as there is no manually placed tab stop nearer to the cursor position than the next default tab stop.

Set this option to "0" to disable it and have tabs only placed with the manually created stops you define.

Tab stops This lists all tab stops defined in the current paragraph by their measurement. Clicking on a stop in the list will display its **Alignment** setting to the right.

To create a new tab stop:

1. Click the + button below the list and type in the placement, as measured from the left edge of the ruler.
2. Select an alignment option from the radio controls on the right ("Left" is the default).

To delete selected tab stops, click the − button. If you make a mistake at any point with this panel, click the Cancel button to dismiss all changes and return to the editor. Otherwise, click OK to make all applied ruler adjustments to the current paragraph.

15.5.2 The Format Bar

The Format Bar provides quick access to common formatting features. The visibility of the Format Bar is toggled with View ▸ Text Editing ▸ Show Format Bar (⇧⌘R).[4]

Designed to collapse when the editor width is too narrow for the full bar, the first segments will become clickable icons that open up menus or palettes providing the same level of functionality (Figure 15.11).

Figure 15.6: Format Bar: displayed in sections

The first section (Figure 15.6) of controls handle font face, size, and variant. All of these tools act immediately on the currently selected text. If no selection is given, then they will alter how you type from the current caret position onward.

a) Styles: use this button to apply styles to selected text or the cursor. Also displays the current style in use. For more information, refer to Styles and Stylesheets (section 15.6).

b) The next set of buttons are for working with fonts manually. Select the font family, variant and size, in that order. Following that are convenience buttons for selecting bold, italic variants, and underscoring text. Most of these functions are also available from the system Font Palette (subsection 15.5.4).

c) Paragraph alignment can be set here. Choose from left, centre, right and justified.

[4] If you are looking for tab and margin controls, the Ruler is available per text view with View ▸ Text Editing ▸ Show Ruler (⌘R).

d) The text and highlight colours are selected from these two buttons, respectively. They have two different modes of operation. Click with the left mouse button to apply the active colour (as shown) to the text. Use right-click to select a different colour, or to choose the "remove colour" swatch (shown as a white box with a red slash through it). Refer to Text Colour and Highlights (section 18.5) for more information.

e) The final section sets paragraph & line spacing, and lists. The spacing dropdown provides some quick presets for line height multiples, but for more complex multi-type spacing preferences, as well as paragraph spacing (above or below), select the "Other..." item at the bottom of this menu. Likewise lists can be customised by selecting "Other..." from its respective menu.

15.5.3 Hyperlinks

Scrivener has broad support for creating hyperlinks in the text, either to other areas within the project itself (section 10.1), between projects, and of course to other software, files and websites. This section will concern itself with the latter form of linking.

If you paste a URL into the text editor, in most cases the editor will recognise it as a URL and automatically create a clickable link for you. If you would prefer full control over when links are created, disable this behaviour with the **Automatically detect web addresses** setting in the Corrections preference pane (section B.6).[5]

Adding Links Manually

Links need not be attached to visible URLs in the text. You can attach a link to any form of text:

1. Select the text you wish to link from in your editor.
2. Used the Edit ▸ Add Link... menu command.
3. Select the type of link you wish to create (if you have the full URL ready to paste, use the **No Prefix** option).
4. Type or paste the URL into the text field and click OK.

This method can also be used to create a fully linked and visible URL in the editor as well. Omit the first step and call upon the command with on active selection. The URL will be inserted at the cursor position.

[5] Despite the label of the setting suggesting otherwise, a wide variety of links conforming to the URI specifications, including file links, FTP, email and so forth will be recognised.

Editing Links

To edit an existing link, including one created automatically by the software:

1. Place the cursor anywhere within the range of a link, or select it completely.[6]
2. Use the Edit ▸ Edit Link… menu command, or right-click and select the command from the contextual menu.
3. Modify the link settings and click OK.

Removing Links

To get rid of a link, or maybe many links at once (perhaps to clean out links from a web page pasted into the editor or imported as text):

1. Select the text from which you want to remove links, or right-click on the link you want to remove.
2. Use the Edit ▸ Unlink menu command, or select the same command from the contextual menu.

Links Aren't Fully Removed?

Some browsers and text editors will not only link text, but add formatting to the text that looks like a link as well. If you strip out the link you will be left with formatting that *looks like* a link—bright blue and underlined for example—but in fact that's all it is: bright blue text with underlining. If when you click on the text nothing happens, it has successfully unlinked and you'll want to clean up the formatting as well.

15.5.4 Font Palette

The font palette is a standard tool provided by macOS, and can be toggled with Format ▸ Font ▸ Show Fonts (⌘T). This palette provides full access to the font typesetting engine, including many OpenType features and custom font effects. For basic font changes, the Format Bar (subsection 15.5.2) will suffice.

The precise appearance of the palette will change depending how large you make it. As the size of the palette increases, more options will be made available to you. Since this palette is provided by Apple, please refer to the help files provided by TextEdit, for details in using it.

To use the font palette to change the appearance of your text, you will need to first select the text you wish to change, and then use the palette. Any changes you

[6] If you select only a portion of the link, just that portion will be changed.

make will be immediately reflected in the editor. You can change your selection while leaving the palette open to work with one bit of text after another.

Managing Font Sizes in the List

If you would like to add or remove font sizes from the format bar, you would do so to the entire macOS via the system palette:

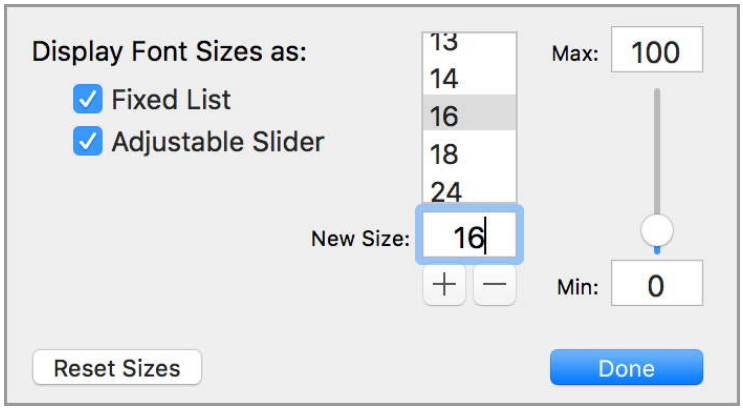


Figure 15.7: Adding "16pt" as a new font size to the system.

1. Bring up the Format ▸ Font ▸ Show Fonts panel (⌘T).
2. Click the ⚙ button and select "Edit Sizes..."
3. Type in a new font size in points in the **New Size** field, and click the +.
4. When finished, click Done.

You should now see 16pt (or whatever changes you made" in Scrivener's font size dropdown in the format bar.

15.5.5 Resetting Formatting

Oftentimes, after you've gathered material from the Web, or imported documents that you've written in another word processor, the formatting of the imported material will not match the default font in Scrivener for new documents. While you can change this default at any time with Editing: Formatting preferences tab (subsection B.3.2), this will not impact documents you've already created or imported, as Scrivener has no way of knowing if that is what you really want to do. Often it might be okay to leave the file as non-standard, especially if it is research material that you never intend to include directly in the draft or even edit.

If you do want to retrofit these documents to your current defaults, you can do so with the Documents/Convert/Text to Default Formatting... menu command. Since this command works at the document level, you can select as many cards,

outliner rows, or binder sidebar items as you please and convert them all in one go.

After clicking the menu command, a window will pop up asking how much formatting you wish to apply to the selected document(s). Most of these options work in a *negative* fashion. By example: if you select "Preserve alignment" and the document is left-aligned, even if your preferences are for full justification, it will not be applied because you have elected to preserve the original alignment. The only exceptions are the **Convert font only** and **Remove all styles** options.

Once you've used this window to set up how deeply this command will impact a document, you can waive it in the future by holding down the Option key when selecting the menu item. Be wary of destructive settings, such as those that strip out all styles, when using this bypass.

> **Double-check your settings and back up!**
>
> Please note that because this command impacts broad changes in potentially many dozens or more of documents at once, there is no undo. The procedure only impacts formatting, but if you are unsure of whether or not the result will favourable, either snapshot (section 15.8) the documents first, or perform the conversion one-by-one and proof the results. Once you are confident you have the right settings engaged, you can proceed at a more rapid pace.

Ranges of text that have been blocked out with Preserve Formatting (subsection 15.5.6) will never be altered by this tool, much in the same way that the compiler will leave them alone.

One last tip is to use "preemptive" format conversion. If you know for a fact you have no need at all for the formatting you are pasting in, you can use the special Edit ▸ Paste and Match Style command to do so. This will treat the text as though it were plain-text, and as such it will take on all of the characteristics of the text around the cursor position where it is pasted. Since the text is treated as plain-text, this means you will lose any inline formatting and function, such as hyperlinks and italics. If you intend to keep this level of formatting, deferring the problem to later by using the document conversion feature will be a better choice.

15.5.6 Preserve Formatting

If you are planning on letting Scrivener's compiler do all or most of the final formatting for you, it can sometimes be useful to preserve ranges of text from the formatting engine. In most cases it will be easier and better to use styles for this purpose, which have a built-in capability of preserving their formatting (section 15.6), but there are a few niche and legacy cases where this tool is used to achieve some effect.

To specify a range of text for preservation, select the text in the editor, and then invoke the Format ▸ Preserve Formatting menu command. This will draw a blue dashed box around the text, which can be worked around and within like any other type of formatting range, such as italics. Also like other formatting tools, to toggle a preserved range off, simply select the entire range and use this same menu command.

Of its special-purpose uses:

— With HTML-based formats, like web pages and eBooks, "preserve formatting" can be used to inject raw HTML into the output (it would otherwise be "escaped" so that it prints in a manner the reader can see. You should consider adopting styles for this approach, which allow for any style range to operate as a raw markup pass-through.

— In Markdown-based formats, when rich text to MultiMarkdown conversion is enabled, "preserve formatting" can be used to pass through raw Markdown syntax.

— In scriptwriting, preserve formatting is used to indicate Dual Dialogue (subsection 19.1.2).

Upgrading from Scrivener 2

For code blocks and spans in Markdown related compile formats: you should use paragraph and character styles now, which can be mapped to generate tabbed code blocks and backtick code spans in the MultiMarkdown Options compile format pane (section 24.14). Our default compile formats automatically map the "Code Block" and "Code Span" styles, included in every new project, to these functions.

Return to chapter ⤴

15.6 Styles and Stylesheets

You might say it only a matter of time before a Scrivener and styles got together. Scrivener has always been about separating the way we write from how text is formatted as we work with it. Much of the design is premised on the notion that a creative writing space shouldn't be difficult to modify or forced to look a certain way because of how we need things to look when we print. Stylesheets are a way of making that nebulous goal a little more detailed and practical.

Upgrading from Scrivener 2

In the past, Scrivener used a simpler system referred to as "formatting presets". These simply rubber-stamped text with a predetermined look and then that was that. Consequently there was one preset list shared by all projects. The main difference you'll need to be aware is that styles are *per project* (and thus can travel to iOS or be used in project templates). This also implies that your previous universal preset list will not be upgraded. If you have some formats in there you would like to use as styles, refer to Converting Formatting Presets to Styles (subsection E.4.1) in the What's New appendix.

If you have used stylesheets in a word processor before, then you could probably skip to the following section, as we will briefly cover what that means in terms of text editing.

If you haven't, you can think of styles as being a slightly more complicated way of formatting text. Why would you want something slightly more complicated? Well in this case, it is a matter of a little extra effort up front quite possibly going a long way in saving you a lot of manual labour down the way. Perhaps you've come across a situation in the past where you've been asked to change the indent level for every block quote in your book, or something along those lines. Without styles, you might get lucky if a tool like Word's "Select Similar Style" manages to catch a good majority of the texts that need to be fixed, but even so you're still faced with a lot of proofing and fixing.

Styles have a very simple concept behind them. Instead of saying "this is a thing because it looks like that thing", it is saying, "this is a thing and thus it should look this way". The important difference between these two slightly different statements is that when a bit of text is tagged as being a thing, such as a block quote, if we later have to change what block quotes look like—all we have to change is the latter part of that clause: "...thus it should look this way". In a system using stylesheets, your block quote looks the way it does because of the stylesheet, and thus changing the stylesheet changes the way it looks.

In Scrivener we can take this concept a little further than most programs can: the way you compile (exporting or printing, if you haven't gone that far yet in your use of the software) your work can differ from the stylesheets you use in your editor.

Our system was designed expressly to fit within the overriding theory that your creative writing environment should be a comfortable place to work rather than looking the way other people (or maybe even yourself) want it to look in the final copies. With styles, you can use whatever suits you for particular texts and then have the compiler convert those to more conventional formatting in the end—or use one style to output three different versions of a text to standard submission format, formatted traditionally to eBook for proofing, or as a technical format for publication (Figure 15.8).

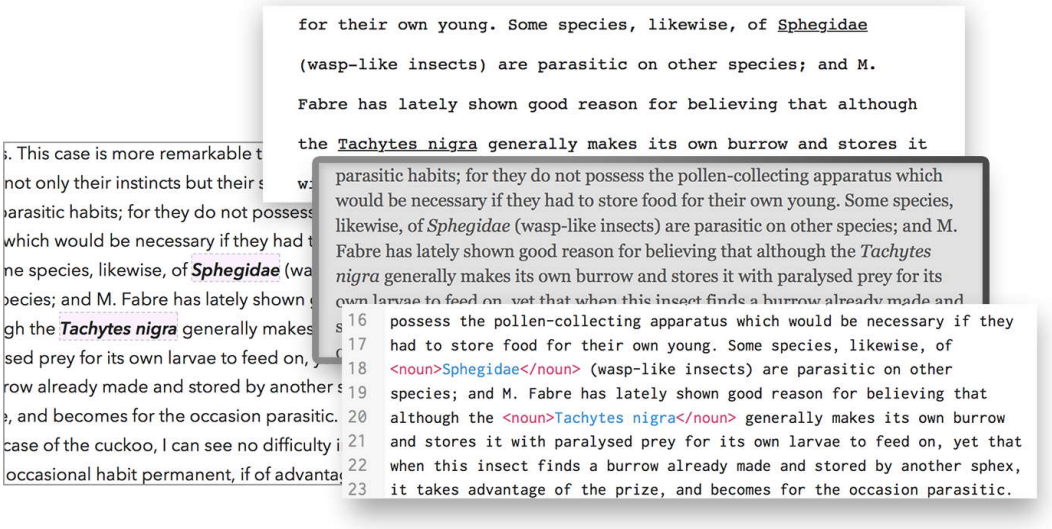


Figure 15.8: Keep styles distinct and easy to work with, while exportable to a variety of uses.

Write using a plain-text format rather than WYSIWYG?

If your preferred or intended mode of writing is toward some plain-text markup system such as Markdown, HTML or DocBook you might think to dismiss styles as an aspect of Scrivener you have no use for. However, styles in Scrivener were designed in part to be an integral tool in workflows such as these—demonstrated in the previous figure, where styled text becomes functional XML on output. For a summary of how you can think about using styles in your project, refer to Styles for Authors Using Markup (section 15.6.2).

15.6.1 Think Different

If you are familiar with styles in other word processors, you may have a few habits to unlearn. If you're new, you should read this anyway because you'll want to know how this system was designed to be used, even if this is all new to you. Scrivener's styles have a very similar concept to what you'll find everywhere else—you tag some text with a named format like "emphasis", later decided emphasis should be green, and all the emphasised text turns green—but one very crucial difference between our system and the rest is that Scrivener does not require you to pervasively tag every shred of text in your document.

We have always had a very basic concept of stylesheets in the software:

— Stuff inserted by the compiler could have a unified look to it since you only set it up in one place—a chapter title for example, contained a numbering schema, font and spacing settings.

— The text you write into the editor is considered working copy, meant to be transformed by the compiler into the final output. You could say it always had an *implied* "Body" or "Default" style applied to it—and going forward that is precisely how you should continue thinking of it.

You might be wondering why? What's the deal with flagging some text as "Body" and making it easy to keep the format up to date? Mainly, Scrivener already does this and has for years. It has ample tools for keeping body text in parity with your defaults (subsection 15.5.5). In fact the way that tool now works is to convert all *un*-styled text to default formatting. So if you used "Body" everywhere the command would no longer work as expected.

Similarly, the compiler still goes on treating un-styled text for conversion if applicable. A submission manuscript format might use 12pt Times New Roman on all un-styled text, but if you have a "Body" style applied to everything using your favourite 16pt sans writing font—guess what, the compiler is going to defer to your demands and you'll never see Times New Roman.

So that is the first principle to understand: with Scrivener, styles are for exceptional cases in which we definitely want the text to look a certain way. In most cases you should see scattered uses of it in your work, rather than hundreds of contiguous words assigned to one style after another. If you're a style purist and coming at this from a semantic text standpoint, think of it this way: all text in Scrivener is inherently styled as "default" until we declare it different otherwise. Given all text can be modified dynamically with settings (compile or in the editing area), it already meets the definition of being semantically normal.

Of course you *can* work in way more familiar to word processing—just bear in mind that much of what Scrivener does out of the box will be hampered by what it sees as explicit demands to format the text a certain way.

Right, with the precautions and background out the way, let's get into the feature and how you may choose to employ it in your work process.

15.6.2 The Basics of Styles

Each new project you create will have its own stylesheet set up for it. If you are starting from one of our built-in templates or the "Blank" starter, this will mean a stock set of basic styles intended to get you started with the concept (Figure 15.10).

No doubt if you decide to start using them, you might find yourself wishing something was slightly different. No problem! That's what styles are good at, read on to see how you can update individual styles in your project—and in doing so update how all of the text looks that uses that style, too.

You might also want to use your growing collection of styles in future projects. That's no problem either, this is what project templates are good at (subsection 5.4.3), and you can even transfer styles directly between projects. We'll get into the details of these things in this section—for now it is merely good enough

to know that while each project has its own stylesheet, it won't be locked away or difficult to use in other works.

Paragraph and Character Styles

As in many programs with stylesheets, there are two fundamentally different types of style that get used, though one can have attributes of both:

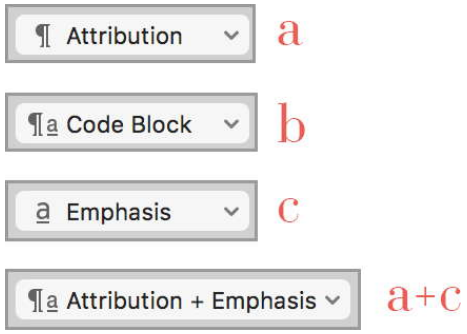


Figure 15.9: The two style types and their combinations, as they would appear in the Format Bar.

— *Paragraph styles*: chiefly concerned with the overall shape text within the unit of measurement referred to as a paragraph. The indents settings, first-line indents, block indent, hanging indent, tab stops, alignment, leading, spacing before and after, line-height and HTML heading levels are all governed here.

When applied to text, they alter the selected paragraphs but do not change any of the character level attributes such as bold or italics. If no selection exists, the current paragraph will be altered.

Paragraph styles are indicated with a pilcrow symbol, marked (a) in Figure 15.9.

— *Character Styles*: without any regard for the presentation of text across multiple lines, character styles are all about letter variants (like italics), weight (bold vs light vs regular), typography, kerning, adornments such as strike-through and underlining and baselines.

When applied to selected text, it will have all aspects of its formatting altered to match the style, removing any existing character formatting such as bold and italics. When applied without a selection the effect will be to change how text is typed from that point.

Character styles are indicated with an underlined 'a' character, marked (c) in the figure, as "Emphasis".

— *Paragraph & characters styles*: these represent a blend of the two concepts. A paragraph style can also define character styles as part of its definition, but we lump them all together as paragraph styles because of how they are

applied. A paragraph style applies any character attributes to *all* text in the paragraph. A character style might only apply bold and bright red colouring to a couple of words in a line, but with a paragraph style makes the whole line bright red and bold. If that's what your headings are supposed to look like, then that is what you want.

When applied to selected paragraphs, *all* formatting will be modified by the stored settings in the style, unless ranges of text within the paragraph have character styles applied to them.

Paragraph + character styles are indicated with both symbols combined, marked (b) in the figure. "Code block" not only block indents the text but sets the character style to a monospace font more suitable for printing source code

The last demonstration of what you will see in Scrivener, marked (a+c), is not strictly speaking a fourth type of style, but rather what will happen if you use a *character style* while typing in a paragraph that has a *paragraph style* applied to it (regardless of whether that style supplies its own character formatting—the character style wins in that contest of precedence). In this case, we are working within an attribution line and have switched on emphasis as well—likely to enter the name of some published work or job title.

With combined styles like this it is good to know where one ends and the other begins. Paragraph styles will always work "underneath" character styles painted on top of them. In the previous example, if I changed the attribution style so that it also applied the character formatting attribute of dark red text, the text marked as "Emphasis" wouldn't be touched by that change. It has its own character styling (black) that supersedes whatever paragraph formatting happens beneath it. If I were to at that point select the range of text marked emphasis and use the Format ▸ Styles ▸ No Style command, it would become dark red like the rest of the line.

What About Fonts?

With the exception of the code block mention above, you have noticed that I haven't used the word *font* yet, which might seem a peculiar omission in a section about styles. Fact is, in Scrivener fonts—and font size for that matter—can be treated as separate from styles. Most of our built-in styles in fact do not mess with fonts at all, leaving that part of the equation up to you. They might change the *size* of the text, or how much space falls around it, but font can be left alone.

It is also possible to have a style change the font as well. After all there is the aforementioned code block (and code span for that matter). Whether you choose to change the font or font size in your own styles is up to you—but know that if you want to keep your styles flexible, you can leave that information out of the definition. A heading style without font information is equally applicable to both sans serif and serif style documents.

The Styles Panel

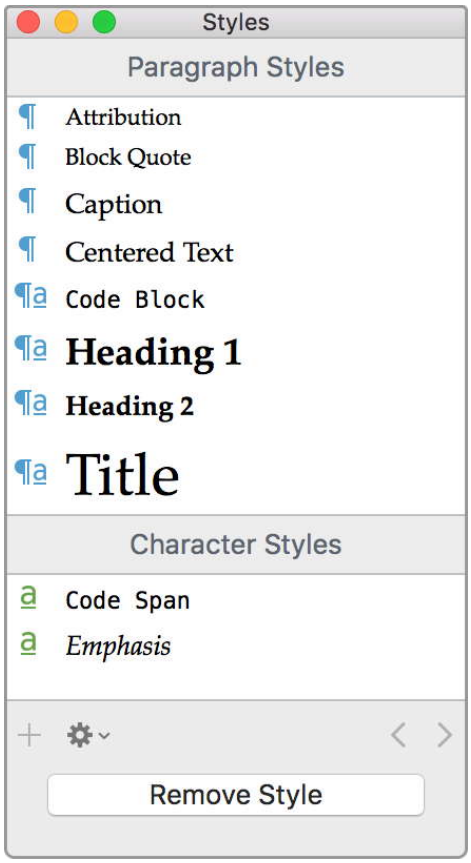


Figure 15.10: The stock starting kit provided to you in every new project—including those upgraded from older projects.

The styles panel (Figure 15.10) is a convenient floating tool that can be used to apply styles to your text as you write, inform you of which styles are applied to the text you are working on and manage the styles as you go. Open this panel using the Format ▸ Styles ▸ Show Styles Panel (^S).

1. In the upper portion of the pane are "Paragraph Styles"—but really this is a mixture of paragraph and paragraph+character styles. Character styles are listed below. These are used on selected ranges of text, within or spanning through paragraphs.

    — Click on a style to assign it to the selected text in the active editor or Quick Reference panel.

    — Right-click on a style to open its contextual menu. The unique commands to this menu are:

        – "Select All Text with Paragraph|Character Style": selects all non-contiguous text within the active editor that is assigned to the selected style.

- "Change Keyboard Shortcut": reassigns the styles shortcut without having to go through the "redefine" panel.

2. The first button on the left is a + button, used to create a new style of any type from the current cursor position or selected text. This button will be disabled (as shown) if there is no valid text currently active. Refer to Creating New Styles (section 15.6.3) for further information.
3. The ✿ button gathers a few useful commands together from the main application menu, as well as providing some functions that are also available by right-clicking on the styles themselves in the above lists.

   — The first three commands relating to selection are documented in Selecting and Searching for Styles (section 15.6.4).

   — The next two pairs of commands will become operational only when the selection encompasses or begins within a paragraph or character style, respectively. Refer to Redefine a Style (section 15.6.3) and Deleting a Style (section 15.6.3) for further information.

   — Lastly, you can "Import..." a stylesheet from another Scrivener project, merging it with the current one. Refer to Copying Stylesheets Between Projects (subsection 15.6.5).

4. Continuing to the right on the same row you will find two buttons: ⟨ and ⟩. Use these to jump to the previous or next instance of the style that is currently highlighted in the lists above. They will prefer character styles over paragraph styles, so if you intend to walk through paragraph style matches, you may need to move the cursor out of any character style range first.

   When no style is highlighted in the list, the action will be to jump from one contiguous range of un-styled paragraph text to the next. Character style ranges will be ignored (and thus selected) in this scenario.
5. Lastly we have the Remove Style button. Use this to strip the styles from the selected text. Refer to Removing Styles from Text (section 15.6.3) for further information.

All projects share the same panel, meaning that if you are working in two different projects, you can keep this panel in one spot and as you switch between them, it will update automatically to show you the styles associated with the current active project.

Styles for Authors Using Markup

Scrivener has always been a tool that contains methods for those who work using markup systems, such as Markdown (and its variants MultiMarkdown and Pandoc), XML, HTML, LaTeX, DocBook and so forth. Of particular use in technical writing, Scrivener's style feature has been carefully designed to not only provide

a traditional look and feel for those coming from word processors, but a powerful platform for defining ad hoc semantic types into the editor and then exporting those types into meaningful plain-text variants in the compile phase. The heavy lifting is entirely done in the compiler, leaving the particular style system itself largely to the domain of aesthetics, while writing.

By way of a very simple example, one could create a style called "HTML Comment". The look of the style in the text editor would be arbitrary—what really matters is what happens when the compiler is set up to look for ranges of text tagged with "HTML Comment" and wraps those ranges of text in a `<!--` prefix followed by a `-->` suffix. Scrivener supports both inline prefix and suffix as well as per-paragraph enclosure.

If you are a plain-text author, consider using the styles feature to your advantage. A real-word example of how this can be done is provided in the user manual you are reading. It makes heavy use of the stylesheet system to implement custom LaTeX control in the output document. We make the source project available on our web site[7].

Styles That Do Nothing

Styles do not necessarily have to perform a formatting function in Scrivener. When compiled they can be removed conditionally, used to pass through instructions to the output format (raw HTML to ePub for example) and a few other tasks—but even beyond that they can simply do nothing at all and leave the text in a state where your readers will never even know it was tagged in the first place.

Consider that marking a text with a named style is a way of tagging text with meaning. You can use formatting for this purpose if you wish, or the purely visual highlighting fills as we'll discuss in the following pages, but have these modifications removed or never expressed when compiled.

A practical example of where this might be useful is in tagging all text relating to a certain character's inner monologue. This could be expressed as simply as italics in the editor—but having done so with styles you'll be able to select by, search for and walk through examples of monologue text throughout the entire book.

If you've ever wanted to mark text as being "something" so you can easily find it later with other similar texts, styles may be what you are looking for.

15.6.3 Using and Managing Styles

In this section we will discuss the application, creation, deletion and modification of styles to text, and within your project in general.

[7] https://www.literatureandlatte.com/learn-and-support/user-guides

Applying Styles

While writing, whether you come across a phrase that needs to be styled or you are preparing to write a phrase is a certain style, your usage of the tools will be similar—and familiar from how you might something simpler like changing the font size: you can either do so while typing, changing the way you type at the cursor from that point forward—or you can select some text and only apply those changes within the selected range of text.

How text is selected will have an impact on how styles are applied.

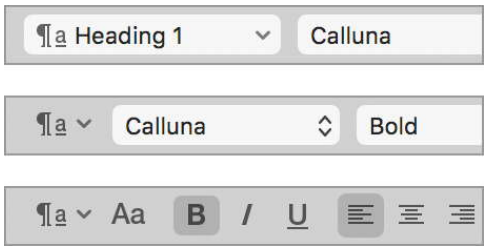


Figure 15.11: The style (leftmost button) and font controls in the format bar collapse when less space is available.

Whether applying styles while typing or retroactively, there are several approaches available:

1. In the The Format Bar (subsection 15.5.2), the leftmost button is dedicated to styles. If the window is wide enough, the name of the style you are currently working with will be printed. Otherwise the button will be as collapsed down, as shown in the second two examples in Figure 15.11.

2. The main Format ▸ Style ▸ submenu lists all styles in your project for easy selection. This command is also available from the text editor contextual menu.

3. The main menu also serves as a reference for any keyboard shortcuts you have assigned to styles. For example, by default the stock "Block Quote" style can be applied to the current paragraph with ⌥⌘2. To change a shortcut, use the Format ▸ Style ▸ Show Style Panel command and then right-click on the style you wish to adjust, using the "Change Keyboard Shortcut" contextual menu.

4. And of course the Style Panel itself is a great way to work with styles if you prefer a single-click visual approach.

5. The Format ▸ Style ▸ Pop Up Styles Menu (⇧⌘Y) is a convenient way to switch to a style. Just as with all menus, you can use the keyboard to type in the first few letters from the name of a style to jump straight to it, and Return to select.

6. If you have a keyboard with a Touch Bar then a default button is a style menu. It will print the name of the style you are currently type in ("No

Style" otherwise), and when you tap on it you can select a style to apply in a scrolling view.

If you're intention is to type in the selected style for a bit, eventually you'll probably be looking for a way to *stop* typing in that style and return to normal un-styled text. The easiest way to do that will be the same way you would stop typing in bold: use the same style command to toggle typing off for that style (character styles only). If the style has a keyboard shortcut, you can use the shortcut to both start and stop typing in that style.

Additionally, all of the methods listed above provide a "No Style" or "Remove Style" choice among their options. For example, you can use the ⌥⌘0 keyboard shortcut, as provided by the Format ▸ Style ▸ No Style menu command. This action works contextually in that it terminates character styles, returning cursor input to the underlying paragraph style formatting (if applicable) or un-styled text. If there is no character formatting directly behind the cursor, then the action will be to strip all styles from the current paragraph.

Creating New Styles

Creating a new style is done by first formatting text in an editor (use dummy text if you do not have an example yet) using the standard formatting tools. Next, select the text (or merely position the cursor within the paragraph for paragraph styles) and then using one of the following methods to create a new style from it:

— The Format ▸ Style ▸ New Style From Selection... menu command.

— Click the + button in the footer area of the style panel.

Using either method, you will be taken to the New Style panel (Figure 15.12). Once filled out to your specifications, click the OK button to confirm and add the style to your project.

Name This is the title by which it will appear in all menus and the style palette. The name of a style can be important when it comes to copying and pasting between projects, as well as when compiling. Scrivener relies upon names to look for matching styles in these contexts. You can for example cause text tagged as "Glossary Entry" to have its appearance altered by compile settings that modify all text tagged as "Glossary Entry", regardless of the project they came from or the formatting in that project.

Most often this will be of importance if you wish to make use of our built-in compile formats. We assume styles will be named in accordance with the example stock styles added to new projects. If a compile format overrides how, say a block quote, looks then it will be looking for "Block Quote". Refer to the Compile Format: Styles pane for further information on that topic (section 24.5).

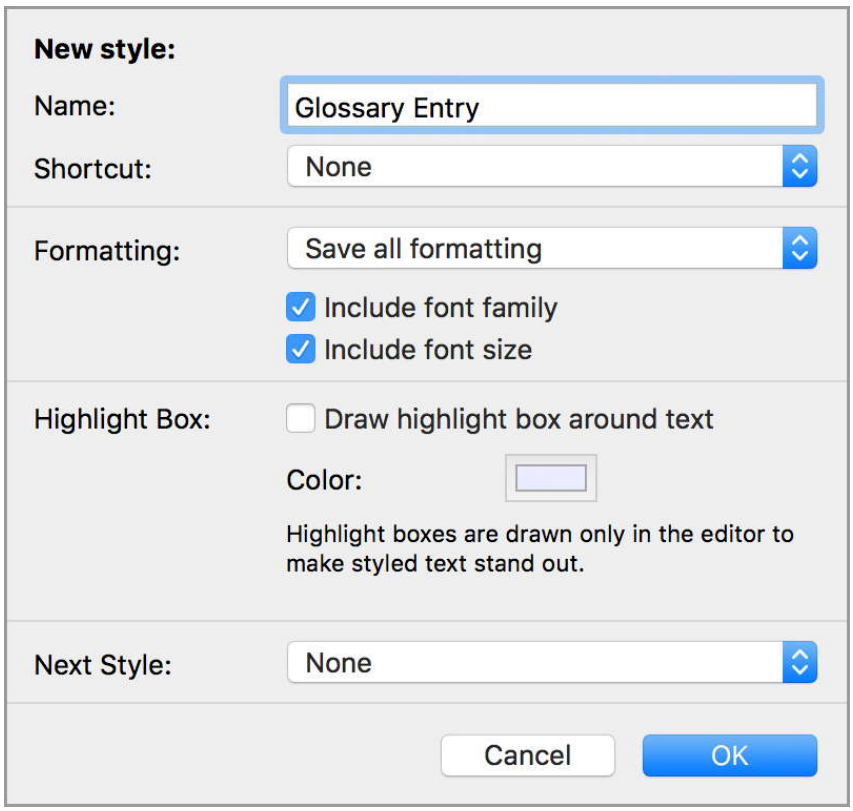


Figure 15.12: The New Style panel is also used for redefining styles in the project.

Shortcut Select a number key to associate with your style. All shortcuts will use the same modifier keystrokes, so all you have to choose is a number. Any numbers already assigned to other styles will be marked as such. You can overwrite an assignment from here, but will be warned when doing so.

Formatting Here is where you set the *type* of style (section 15.6.2) and the scope of what formatting it will store and apply to future text:

- *Save character attributes*: the type of style will be Character. Only those settings that impact the letters and words themselves will be saved.
- *Save paragraph style*: the type of style will be Paragraph. Only the settings that impact how paragraphs are displayed will be saved.
- *Save all formatting*: the type of style will be Paragraph+Character. All formatting will be saved, and when applied to paragraphs the entire paragraph will be formatted uniformly.

Below the type selection are two additional options for whether to **Include font family** and **Include font size**. These can be useful when creating general purpose styles. For example a heading style would often change the

font size, but it might not bother with changing the font family, if heading fonts are meant to be the same as that used by body text.

Highlight Box In some cases it might be advantageous to draw visual attention to a style, either as embellishment or to act as a crucial signifier in cases where the actual formatting of the style itself doesn't matter or isn't meant to be visible to the reader. Enable the **Draw highlight box around text** checkbox, and then click on the colour chip to select what will be used as a background fill behind the text.

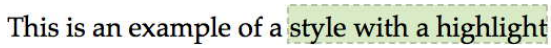


Figure 15.13: Highlight boxes can visually accentuate styles in the editor, and are particularly useful if the style itself being used for something other than formatting.

Next Style This final option determines what style will be set when ending the current paragraph. The default is "None", which will return you to default un-styled text. You might for example wish to have the Block Quote style transition into the Attribution style on the following line.

It is also possible to select the same style you are creating with the "This Style" option. This will be useful for styles that tend to span more than one paragraph or line, such as code blocks.

Since character styles naturally persist from one line to the next, this option is unavailable to them and will be disabled.

Redefine a Style

Modifying a style follows much the same procedure that creating a new one does. Text is formatted the way you would like the style to look (use dummy text if you do not have an example yet) using the standard formatting tools. Select the text (or merely position the cursor within the paragraph for paragraph styles) and then using one of the following methods to update an existing style:

— Use the Format ▸ Style ▸ Redefine Style From Selection ▸ submenu to select the style you wish to overwrite.

— From the style panel, right-click on the style you wish to overwrite, and select "Redefine Paragraph Style From Selection…".

The same dialogue you used to create the style will appear. In most cases you will not need to change anything here, and can just submit the form to apply the formatting changes you've made. However if you do wish to change the parameters of the style, you can change every aspect of it save for its type. With paragraph styles, you can retroactively change whether or not character attributes are included with the style.

Styles cannot be duplicate directly. If you are looking to create a derivative, instead of replacing an existing style, use the original to style your text, modify its formatting in the editor, and then use your preferred method for creating a new style.

Lastly, if all you wish to change is the keyboard shortcut, the right-click contextual menu in the styles panel contains a quick and easy method for doing so, without going through the whole redefinition process.

Deleting a Style

Deleting a style can be done at any time by using the Format ▸ Style ▸ Delete Style ▸ submenu. This command cannot be undone, but since it does not actually remove formatting from the text, if you do accidentally delete an important style you can re-create it by finding an example in your text, and creating a new one from scratch from its formatting.

Removing Styles from Text

Removing styles from existing text is very similar to what you would do when you are done typing in a particular styled range. The Format ▸ Style ▸ No Style menu command and its shortcut, , also strips out the style from the selected text or the paragraph.

— For character styles, you must always select the entire range of text you wish to remove the style from (just like how you would need to select an entire range of italic text to remove italics from it). If you use this command by placing the cursor somewhere within the range, all that will happen is you will insert a "no style" declaration at the cursor, meaning that if you start typing it will not be typed in that character style and the original range will be broken in two.

— When character styles are removed from paragraphs that have a paragraph style, the text will revert to the formatting established by the paragraph style. For example, if the paragraph assigns a dark red colour to text but our character style is light blue, when we strip out the character style from that paragraph it will turn dark red.

  When removing character styles from paragraphs that are un-styled, the text will merely become un-styled as well.

— To remove a paragraph style, even if it supplies character formatting, simply place the cursor anywhere within the paragraph not otherwise occupied by a character style and use the command. Any character formatting and styles found within the paragraph will be retained, but all other text and the paragraph settings themselves will be converted to default formatting.

— If the cursor is placed outside of a character range in a paragraph that has no style applied to it and the No Style command is used, the effect will be to remove all character style assignments from within the paragraph. The paragraph itself will also be reset to default formatting. In this way you can fully clear style data from a selection by using the command at most two times.

Get rid of them all!

If you wish to strip *all* styles from selected documents (useful when importing from other word processors and getting unwanted styles imported), there is a special option provided in the Documents ▸ Convert ▸ Text to Default Formatting... menu command, **Remove all styles**. This will clean out all style information and remove all formatting related to them, within the limitations of this tool (i.e. if a style provides inline formatting such as italics, they will not be stripped out but the style will be).

Of course if you want to go even further and get rid of all formatting, try Cutting the text and then using Edit ▸ Paste and Match Style.

15.6.4 Working with Styled Text

Selecting and Searching for Styles

Marking text as styled is useful for many reasons, chief among them the ability to locate bits of text that have been tagged under a particular style. There is of course great appeal in easily walking through every figure caption in your dissertation or every block quote in a biography.

The styles panel itself has several of these tools gathered together into one place, by either right-clicking on a style and using the "Select All Text with Paragraph Style" command (which selects all ranges of text within the current active editor that is tagged with the selected style), or clicking on the ⚙ button below the style lists. The selection commands in this menu match what are available universally, among a number of other tools, in the main Edit ▸ Select ▸ submenu:

— *Select Style Range*: expands the current selection, or select from the around the cursor position, to encompass the styled text around it. This command works under the following logic:

  – If a character style is found, the contiguous range of text around the cursor using that style will be selected.

  – If a paragraph style is found around the selection, then all contiguous paragraphs using that style will be selected. This includes conditions caused by the above, meaning we can first select a character range,

and then use the command a second time to select at the paragraph scope.

- When there is no style in use around the cursor, then all contiguous text that also has no style applied to it will be selected. Thus in a document with no styles, this command will effectively select all text within the document.

— *Select All Style*: Selects all non-contiguous text within the same text view using the current style beneath the cursor, or as found beneath the left-most edge of the selected range (the selected text itself will be ignored). As with the previous command, the nearest character style range will be preferred over the paragraph character range, in cases where the cursor sits within text that has both character and paragraph styles applied to it. Also as with the prior command, if the cursor or selection encompasses text with no style applied to it, the command will select all other text likewise without styles.

— *Select Next in Same Style*: Using the same criteria as Edit ▸ Select ▸ Select All Style, this command will select the next phrase of text found within the current editor using the same style. If the cursor is within block quote, then the next block quote will be selected. This form of selection will wrap around from the bottom of the document back to the top if necessary, which means if only one example of the style exists, the very context the cursor currently sits in may be the "next" available example of text using this style.

It's also worth noting that the styles panel has a couple of 〈 and 〉 buttons which perform the same function as the latter menu command, although in either direction.

While these commands provide a good amount of flexibility and searching power within the active text editing session (and do consider that Scrivenings sessions can broaden the amount of text you are working with at once), sometimes you want to navigate through however many chunks of text it takes to get from one instance of styled text to another—or maybe you want to only find styled text that contains a certain bit of text, like "figure" vs "table". This is where the Find by Formatting Tool (subsection 11.6.6) comes in, with Edit ▸ Find ▸ Find by Formatting... (^⌥⌘F).

Copying and Pasting Styled Text

When working within the confines of one project, the matter of copying and pasting styled text is of so little consequence it hardly bears mention. It will happen, and you don't have to worry about styles getting lost between individual chunks of text.

What will require more attention to detail is when copying text *between projects*. The first thing to know about this process is that Scrivener will use

name matching, not formatting, to look for style assignments within the same type (paragraph vs character). If you have a style called "Character Name" in both projects, then you will be able to freely copy and paste between those projects without fear of losing style assignments, and doing so will automatically update the pasted text's formatting so that it matches the stylesheet in the project you are pasting into.

In cases where the pasted text contains styles *not* named in the target project, those styles will be stripped from the text and any attributes they were contributing to the formatting of that text will not be applied. For example if we paste a "block quote" into a project that has no block quote style the paragraph will likely end up looking like a normal paragraph. If retaining the formatting that styles provides to you is important to you, always make sure the target project address those styles in its stylesheet (and refer to the following section for tips on merging or importing stylesheets).

Incidentally, the latter is a good way to strip one specific style *and* its formatting out of some text. Since deleting styles from a stylesheet leaves formatting intact in your project, if you want to also remove that, try pasting into a project that doesn't have the style you are looking to fully purge.

Importing from Other Word Processors

In general, importing documents from other word processors that have stylesheets will follow the same rules as described for copying data between Scrivener projects. The software will be looking for styles by their names, and thus you will need to have styles already created in the project's stylesheet by those names in order for them to import.

Not every document format may have the same results. If you find some styles are not importing as expected, try another format like RTF, ODT or DOCX.

15.6.5 Copying Stylesheets Between Projects

If you need to copy a stylesheet from one project to another, the Format ▸ Style ▸ Import Styles... menu command (also accessible from the ⚙ button in the floating styles panel) will be your way of doing so.

- Styles that are identical in both name and formatting will be ignored.
- If the project you are importing from contains styles that have the same name but different formatting, you will be asked how to handle the import:
  - Keep Existing Styles: no styles as defined in the current project will be modified. Only new styles will be imported.
  - Replace Existing Styles: the formatting for existing styles will be updated to match the project you are importing from. This can be useful if you want to bring two different projects into parity with one an-

other, or to update an older project with refinements made to newer ones.

- Add Imported Styles: this is the safest option. The styles will be imported as new styles (Scrivener will add a number of the name to keep them separate). You can then choose to keep them or discard them, or manually merge select styles while leaving others alone.

Manually merging the formatting of two styles is simple:

1. Create some dummy text somewhere and assign the newer style to it so that you have an example of its formatting in the editor.
2. With the dummy text selected, use your preferred method to redefine a style (section 15.6.3) (such as the Format ▸ Style ▸ Redefine Style From Selection ▸ submenu), redefine the older version of the style you want to overwrite. Double-check the settings in this panel and then click OK to confirm.

All text assigned to that style will be updated to match the new look. You could now delete the newer copy of the style now that you have its formatting imported. Do note however that if you have text assigned to the newer style it will of course remain assigned and deleting it will cause that text to lose its style assignments.

Return to chapter ⤴

15.7 Working with Images

While images themselves cannot be placed into the draft folder as binder items, you can insert an image into the text either by dragging an image file in from the Finder; dragging an image document in from the binder, outliner or corkboard; or by selecting Insert ▸ Image From File.... The image will be placed at the current cursor position. Images placed into documents in this fashion will create a *new copy* of that image, even if it was dragged in from the Binder. This becomes important when working with placeholder images. If you intend to later edit your images or replace them with updated copies from a graphic designer, you may wish to instead use Linked Images (subsection 15.7.4).

Viewing inline images full-size

To view an image in the editor on its own, you can drag the image from the text into the header view of one of the editors. You can then double-click on the image and zoom in or out on the image. Note that any changes made here will not impact the embedded image in the text.

15.7.1 Resizing Inline Images and Naming Them

To resize the inline image, double-click on it. This will bring up the image scaling panel. Use the controls to shrink the image or increase its size, and then click on OK. Unchecking "Lock aspect ratio" allows you to adjust the width and height independently and therefore distort the image. Clicking Cancel restores the image to its former size and returns you to the editor. Note that some images types (notably, vector PDF) cannot be resized; double-clicking on such images will do nothing.

It is important to note that resizing images in the editor is an output, rather than an aesthetic, decision. When an image is resized in the editor, it will be directed to use that size in the final output as well. If you find the graphics you are using are too large, and are getting in the way of writing, you may consider using placeholder thumbnails instead. Using linked images is ideal for working this way, as you can swap out the full-size images on the disk prior to compiling.

For embedded graphics you will also note a **Name** field in the image editing panel. For most ways of working this name will do very little, but if you use a markup format like ePub, HTML or Markdown then the name of the image can be important if you intend to work with the compiled output using other programs—particularly if you've pasted most of your graphics in and do not wish to have to contend with hundreds of generically named "Pasted Image" files.

15.7.2 Saving an Image Out of the Editor

To save the image to the disk, right-click on the image and select "Save as Picture..." from the contextual menu. This option will not be available with a linked image, which naturally is already on the disk—instead a command to "Reveal in Finder" will be provided.

You can also simply drag and drop an image into the binder to create a copy of it in your project.

15.7.3 Embedding Inline PDFs

It is possible to drag PDF files into your editor to embed them. This feature is meant to allow the usage of graphics saved in the PDF format. It will not allow multi-page documents to be inserted into the final manuscript, and if the original PDF is pre-formatted for print, you will very likely need to crop the result down significantly so that it can fit within the page margins, using a PDF editor such as Preview, or Acrobat Pro.

When compiling to formats that do not support embedded PDFs (only a few of the text-based markup formats support PDF), a raster graphic will be automatically generated for you and embedded as a PNG file, if the compile format allows embedded graphics. When use File ▸ Export ▸ Files... and selecting RTF, this conversion will also take place. This will cause a loss of vector data, and so the processes uses a large PNG that is then shrunk down to scale to maximise qual-

ity. You can adjust the DPI used for this in the Sharing: Export preference pane (subsection B.7.2).

15.7.4 Linked Images

Linked images are useful when the actual graphics in your project text are incomplete or placeholders for larger, production-ready graphics—or if you just wish to maintain your image repository external for the purposes of ongoing edits and batch processing. If you are familiar with desktop publishing tools like Adobe InDesign, then the manner in which linked images work will be familiar to you. For those not aware, linked images are the placement of graphics in your text in such a way that the representation of the image in your editor is being generated by files *outside of the document*, as opposed to being saved into the text of the file itself.

To create a linked image, use the Insert ▸ Image Linked to File... menu command. Linked images will look and act like embedded images in every way. You can by and large ignore the difference between them while working.

When compiling with linked images, the *current* version of those images on disk will be used to create the embedded copies used in the compiled version. Since nearly all of the compilation formats do not support active image linking, Scrivener must take the current version and embed it in the final copy. So if you are in a workflow that involves external help from designers, be sure to get your external images up to date before producing final compiled copies.

Resizing Linked Images

Just as with embedded graphics, you can adjust the display size of the image (subsection 15.7.1). It is important to know that doing so will alter the original image's meta-data on the disk. The image will not have its pixels touched, but its print size information will be altered. If it is important to you that Scrivener not adjust the original files then do not use the resize tool in the editor, or consider using image placeholder tags (subsection 15.7.5), where sizes can be specified as compile-time directives on the output images rather than the originals.

Linked images also have a few extra features in the panel you get when double-clicking on an image. The full path to the image on the disk will be printed for your reference. This text can be copied and pasted as needed.

For a friendly way to get to the image, click the Reveal in Finder button. If you need to change the source file for whatever reason, use the Change... button to select a new graphic.

Updating the Image from the Disk

For increased performance, Scrivener uses a cached version of your image when you first view it (or initially link to it) within a session. Whenever you re-open the project, this thumbnail will be updated with the current image on the disk

automatically. However if you have updated an image on the disk and wish to see the results in your editor without reloading the project, right-click on the image and select the "Reload from Original Image" menu command.

This will also reset Scrivener's record of how large the image is and its pixel resolution. So while reloading the image might be an aesthetic option if only the pixels of it have changed on the disk, if the *size* has changed on the disk, it can be important to reload it using this command, as Scrivener may otherwise squash or display the image incorrectly when compiling.

Renaming an image on the disk

You can of course freely rename images whenever and however you like, but you will find that upon reloading the project at a future point, the links will have broken. A technique for safely renaming an image name while preserving the link is to do so from within Scrivener:

1. Double-click the linked image to edit it.
2. Click the Change... button to bring up the file chooser.
3. Right-click on the name of the image (which will be selected automatically for you) and select the "Rename" option.
4. Once you've renamed the file on the disk, click the Open button to confirm reassigning the link to the same file by its new name.

Otherwise, refer to Fixing Broken Links (section 15.7.4) links, in the following section, for tips on how to fix images that are no longer linked properly.

Utilising Links on Multiple Computers

Those familiar with the use of linked resources will already know that using links means that the position of those external resources must remain stable, in order for the host program (Scrivener in this case) to maintain a connection with them. If the files are moved or renamed, the link breaks and you will need to manually update them. This means that in most cases, links will only work on one machine, since the full path name of a file tends to vary, given that it includes the name of the computer's hard drive. It is possible to carefully work around that, particularly with the use of external hard drives that are all mapped to the same drive assignment on the computer, or keeping all satellite computers identical in regards to user names and hard drive names.

This problem becomes impossible to work around when working cross-platform, as the path addressing scheme used to name a file differs between Mac and Windows computers (and iOS effectively has no concept of a file system). This does not mean that using linked images prohibits cross-platform usage, or vice versa, but what it does mean is that only one machine can be your "home" computer in terms of final document assembly. A project can lose track of the

linked images while you are working on other machines, but since these links are stored as text paths to the files, when you return to the home computer, they will all link up again.

Fixing Broken Links

If the name or position of the image changes on the disk then the link to it in the editor can become broken. A broken image will fall-back to a warning message, printed in red highlighted text, with a special token identifying it as an image link, and the expected path to the image printed following:

```
MISSING_IMAGE:/path/to/image.png
```

The simplicity of this system means that if you fix the path to the image right in the text editor and the reload the project it will link back up. This also means that large-scale changes to paths can be made using the Edit ▸ Find ▸ Project Replace... tool.

If you are in a workflow where image breaks are a common occurrence, you might consider creating a saved search collection (subsection 10.2.4) that scans the project for the text `MISSING_IMAGE:`.

Images Linked to Binder Items

You may at times wish to use linked images to keep your text files trim rather than bloated with large files—but at the same time not have any need for the advantages using the file system gives you, and would prefer not to have to deal with the disadvantages, such as fixing broken links or having to contend with issues between synced systems across multiple platforms.

Linking to images in your binder is the solution to this problem. The graphic is not placed into your text directly, keeping draft files trim and efficient, while the graphic's path and file name are managed by the project format.

To link to a graphic in the binder:

1. Import the graphic into the binder if necessary. The name you give the file in the binder will be used when compiling it, for those formats where it matters.
2. From the text editor, use the Insert ▸ Image Linked to Document ▸ submenu to select the graphic. Only graphics and PDFs will be listed in this menu.
3. Alternatively, if you intend to predominately link to images in the binder, it will be far easier to visit the Behaviors: Dragging & Dropping preference pane (subsection B.4.4) and setting the **Link to images dragged from binder into editor** option. You may also want to visit the Document Links tab in that same pane and enable the **Image links create back-link bookmarks** option. With that enabled, you can click on the image in the binder

and view its bookmark list to see precisely where it is used within your draft folder.

When images have been edited out of the binder, you can have Scrivener reload the changes by right-clicking on the image and selecting the "Reload from Original Image" contextual menu command.

15.7.5 Image Placeholder Tags

It can at times be desirable to not include the graphic in the source text at all, but rather to specify its name or file path, using text, similar to how one would link to an image in a web page or forum post. You can also specify the dimensions of an image precisely, using text, which is important for getting them right the first time, if you do not intend to polish off the compiled document after compiling. Using tags allows you to bring images into areas of the document that otherwise wouldn't support them—on account of their being defined by plain-text fields—such as custom separators, title prefix and suffix fields.

Textual links to images afford you additional powers over typical images displayed in the editor. Since they are text based, they can be easily modified by compile settings on the fly. You could for example place images into one document format but not another, even change which image to be used used or its size.

The basic syntax for the image tag is:

```
<$img:IMAGE_NAME;w=WIDTH;h=HEIGHT>
```

Everything after the image name is optional.

Image Identification

The image must be specified either by its imported name, as displayed in the binder, or by a valid system path to the image's location on your computer. Path names can be relative to the location of the Scrivener project, so if the graphics and the project are stored in the same folder, you would only need to declare the path as 'image_name.png' for it to work. You can also use the UNIX shortcut for your home folder, the tilde. Full absolute paths are acceptable as well.

```
<$img:~/Dropbox/MyBookFolder/scene_break.png>
```

This also demonstrates a safe way to link to an image that will work from multiple computers all using this Dropbox account, since the details of the user folder name are represented by the tilde.

The following example references a graphic that has been stored in a subfolder called Resources, which is a folder alongside the Scrivener project:

```
<$img:Resources/python_chart.jpg>
```

Here is an absolute link to an external hard drive called "Research", where a PDF in the subfolder "MyThesis" in included. As with all embedded PDFs, you should ensure it is only one page in length and smaller than the text print block (sans margins):

```
<$img:/Volumes/Research/MyThesis/some_data.pdf>
```

When using graphics that you have imported into the project binder, you can refer to them by their binder title (you needn't know the extension or type of graphic it is). This will be used in precedence over any path names, and the first matching title name (counting from the top of the project) will be preferred over any other images likewise named in the project. A simple example, specifying an image that is called "Python Growth Rates in Everglades":

```
<$img:Python Growth Rates in Everglades>
```

If for some reason you have two graphics named "Python Growth Rates in Everglades", the image that is "highest" on the binder list will be selected. It starts at the top of the binder and works down until it finds a match. Likewise, if it finds any match at all it will use it, instead of looking outside of the Scrivener project for a graphic. So if the binder name included the file extension, it would still use the imported version instead of a copy with an identical name located in the Scrivener project's location (making it a relative link).

Specifying Image Width and Height

The width and height of the image is an optional value, declared in points. If you have already sized the graphics correctly in accordance with your printer's specifications and output quality, then you will not need (or indeed want) to change the image size using Scrivener. However if you are not concerned with pre-sizing the images and just want to let Scrivener handle them for you, you can specify the width and height of the image in the placeholder tag, to instruct the compiler to resize it as necessary. It is valid to specify only the height or the width alone. This will cause the compiler to shrink or expand the image to match the specification you've provided, automatically resizing the opposing measurement, keeping the image's original shape, or aspect ratio, intact. If you declare both the width and height yourself, it is possible to squish or squash the image however you please.

```
<$img:Portrait of Dickens.jpg;w=468>
```

This will resize the portrait so that it is 468 points wide[8], adjusting the height of the image so that it remains proportional with the original in terms of aspect ratio; no squished faces here!

[8] At 72 points per inch, that will be 6.5", or just wide enough to extend to the margins in a US Letter page with 1" margins all around.

When compiling to one of the ePub 3 or Kindle KF8 format types, you can optionally declare an image to be set to a percentage of the screen's display, rather than a fixed point width. This will be done by using a secondary setting in addition to a fall-back static width setting. The following example will display the image at 468pts wide in most compile formats, but when compiling to ebooks, will instruct the image to be displayed at 50% of the reader's screen size, dynamically:

```
<$img:Portait of Dickens.jpg;w=468;ebook=50%>
```

Tips for Usage

Here are some ideas for how this feature can be used, as well as some handy tips for common uses.

— *Using Replacements to Change Images*: use of replacements in the compiler (subsection 23.4.4) will be evaluated prior to any images being included. That means you could use the feature to scan for the placeholder tag and remove it from the document, or perhaps change the path of the image to something else, which could be useful if you have two different groups of images, one optimised for black & white printing, and the other for an online full colour PDF or lower resolution eBook.

— *Searching for images*: since the image is defined as text, that means you can search for it by name, or even search for all "`<$img...`" tags in the draft to get a list of documents with figures, which could be saved as a search collection (subsection 10.2.4).

— *Using images as a compile-time prefix or suffix*: when compiling, it is possible to use an image in the prefix or suffix areas of the section layout compile option pane (section 24.2) or in separators (section 24.4), to use graphics between scenes, or below chapter headings.

— *Specifying an image when two or more share the same name*: in cases where you have multiple images with the same name in the Binder, if you need to ensure that a certain one of these (rather than the first one in the list from top-down) is used, select the image placeholder and create a document link (subsection 10.1.1) from it pointing to the correct image.

15.7.6 Images in the Output

Handling Print-Ready Images

In the case of print-ready images the text editor will display a rough approximation of the image's final size, rather than its actual pixel size, when the DPI and physical measurements are set to an appropriate value for printing. For instance a 300 DPI image that is meant to be 2.5" wide on the printed page will appear approximately 2.5" wide on your monitor, and so it will not dominate the editor.

Scrivener uses the publishing standard of points as a unit of measurement, rather than pixels or other rule-based measurements. Adjusting the scale of an image in the text editor does not impact its literal pixels, but rather the point value which will be used to determine how large it displays. Points are a fixed unit of measurement (there are 72 of them per inch). If you are using a typical US Letter with 1" margin document then the widest you would want an image to be is 468 points. For A4 with 25mm margins you wouldn't want an image wider than 453.6 points.

It's worth noting that some programs further down your toolchain might not understand points properly and treat them as pixels. In particular some eBook author software may confuse the matter. In many cases it will be best to save the insertion of production-ready graphics into the workflow after you've compiled from Scrivener.

If you've been provided full size, print-ready graphics for publication, it is often best to not embed them directly into your document. Embedding images in the document is good for small placeholders, but since the images are saved into the RTF file itself, adding very large high resolution graphics to your files may slow down Scrivener's ability to load and save your file while you work on it, and may even cause instability when it comes time to combine everything together into one large document. Do also consider common production tactics of using linked images or placeholder tags instead of fully embedding large graphics into your source files.

Images and Compiling to Markup

When using Web Page (HTML), or any of the MultiMarkdown and Pandoc compile formats, inline images (and linked images for that matter) will be converted appropriate syntax in the text and exported into the compile folder along with the text file itself. For formats that end up combining the source files into a bundled format like WebArchive or DOCX (via Pandoc), these intermediate files will be handled in temporary folders and won't appear in your output folder.

Return to chapter ⤴

15.8 Using Snapshots

The theory behind a Snapshot is very simple, and similar in principle to how you might use the "save as" feature in a traditional text editing program.

As with many modern programs, Scrivener saves automatically as you work, whenever you pause for a couple of seconds. In comparison to the older method, where one loads a file and saves periodically, you lose the benefit of being able to work "off the disk" for a while, choosing when to save and when to revert to what has been saved.

Snapshots help you bridge the gap between these two different ways of working. Taking a snapshot is like saving your document, the only difference is, you

get a *record* of each save point in a tidy list, stored in the document's inspector pane (rather than littering your Documents folder with "Save As" copies, for example!).

In the course of editing, you might change your mind about something you deleted more than one save iteration ago. With a traditional save as you go model, you'd be out of luck unless you manually made a backup somewhere. With snapshots you can scroll back through the history of a document, or even search for phrases within them, find the point in time where the deleted passage still existed and copy and paste it back to the present—or even just roll back the whole document to that spot.

What About Milestones

In addition to periodically saving the evolution of a chunk of text, some people like to use snapshots to set important editing "milestones" that might one day wish to return to. Scrivener accommodates both styles of working. You can take "Untitled" Snapshots with a single keystroke, or give them a memorable title to refer to later on.

Taking a "snapshot" of a document copies the text of it at that exact moment in time and stores it so that you can return to it, or restore it, at a later date. With routine usage, you need never need worry about making major edits to a document, because you can take a snapshot of it *before* you begin editing and then restore the older version later if you change your mind about the edits you have made.

Figure 15.14: Three different document types displaying "dog-ears".

You can tell if a text document has any associated snapshots by its icon. Documents that have had snapshots taken of them have the top-right of the paper icon folded down in a "dog-ear" (Figure 15.14). You can also tell by looking for a dot beside the camera icon along the top of the inspector pane if it is open (Figure 15.15).

Figure 15.15: The dot in the upper-right corner of the snapshot icon indicates this document has snapshots.

The key thing to understand about snapshots is that they provide a way to

set save points for *individual* text items in the binder.[9] They are not a tool for providing an overall snapshot of the entire project, structurally speaking, and are probably not the best tool for taking a quick snapshot of your entire draft. In most cases, large scale backups like this would be best created using the File ▸ Back Up ▸ Back Up To... menu item, or by duplicating large portions of the Binder and stashing them elsewhere for future reference.

How to snapshot an entire folder

Selections must be precise, but you can easily select a folder and all of its subdocuments with a single command, Edit ▸ Select ▸ Select with Subdocuments (⌥⌘A), from an outliner or binder view, and then take the snapshots.

15.8.1 Creating Snapshots

While working on a document, there are three easy ways to snapshot it for future reference.

That aside, it is simple to quickly snapshot multiple documents:

— Select the documents you wish to snapshot in the binder sidebar, corkboard, or outliner.

— Use the Documents ▸ Snapshots ▸ Take Snapshots of Selected Documents menu command (⌘5)to create a snapshot for each item.

Snapshots can later be accessed in the Inspector pane. You can jump straight to them with the Navigate ▸ Inspect ▸ Snapshots menu command (^⌥⌘M).

— The alternate way of taking a snapshot is to supply them with a title as they are created, use Documents ▸ Snapshots ▸ Take Titled Snapshots of Selected Documents (⇧⌘5). The title you provide will be added to each document in the selection. Snapshots can also be titled at a later time using the inspector.

[9] Snapshots only store the *text* of documents (not notes, synopses or metadata), and are therefore only available for documents that contain text.

Saying More About a Snapshot

If the title field isn't enough, a trick for more thoroughly commenting on the purpose or state of a snapshot is to combine it with the use of inline annotations. Consider writing a short synopsis of what the snapshot was taken for into the top of the document before snapshotting, and then remove it afterward. Now that statement will always be clearly presented to you at the top of the snapshot text preview.

The third method for creating snapshots is from the Snapshot pane itself in the Inspector. Click the + button in the header area of this pane to create a new snapshot.

15.8.2 Managing Individual Snapshots

Existing snapshots are all managed in the inspector pane on a per document basis. To jump to snapshots, click the button with the camera in the inspector tab list or use the Navigate ▸ Inspect ▸ Snapshots menu command. More details on the functions available in the inspector can be found in the documentation on the Snapshots Tab (section 13.6).

15.8.3 Viewing Snapshots in the Editor

It is possible to load snapshots as read-only text into the main editor splits. There are a few ways of doing so:

— Drag the snapshot you wish to view into the header bar for the editor split or copyholder you wish to load it in.

— Right-click on the header bar icon menu for the item whose snapshot you wish to view, and use one of the "View Snapshot in Other Editor" or "View Snapshot on Copyholder" commands. The submenus for these will display a list of available snapshots to load.

When you're done, use the history feature to return the editor or copyholder to what you were looking at before. Snapshots themselves do not occupy a slot in history, so you will be unable to return to them (save for to load it once again) once you leave. Consider Locking the Editor (subsection 12.2.1) if you are inadvertently losing your snapshot view.

15.8.4 Comparing Changes in the Editors

The comparison feature lets you analyse the differences between versions of your text. There are several places that allow for comparison. Chief among them is within the inspector's Snapshots Tab itself (subsection 13.6.4), where most of this capability and adjustments to its behaviour will be documented.

As noted previously, you can load snapshots straight into one of the split editors, or one of their copyholders. It is also possible to use these locations for comparison, meaning you aren't stuck to reviewing changes in a narrow sidebar. There are a few ways of doing so:

— When dragging and dropping the snapshot into a header bar, hold down the Option key.

— Right-click on the snapshot to load it on the copyholder for that editor.

— When using either "View Snapshot in Other Editor" or "View Snapshot in Copyholder" hold down the Option key when selecting the menu command.

— Right-click on the editor header bar and use the same commands provided there to load a particular snapshot into either the other editor or that editor's copyholder. Each snapshot for that item will be listed as submenu items beneath these options.

When viewing comparisons in the editor it is only possible to show markings between the selected snapshot with the current text, not another snapshot.

15.8.5 The Snapshots Manager

While being able to click on any chunk of text in the binder to examine its history is undeniably useful, a tool that can examine *snapshots* themselves, as entities across all files within the project, can greatly enhance your overall sense of how the text as a larger whole has evolved, and can make pruning old and unwanted snapshots a snap.

To bring up the manager, use the Documents ▸ Snapshots ▸ Snapshots Manager menu command. In projects with a large quantity of snapshots, it make take a few seconds to load them all up.[10]

The window is divided into two panes. The sidebar on the left contains an alphabetically sorted list of every document in the binder that has a snapshot, and then grouped within each, a list of all respective snapshots in reverse chronological order.

In the provided example (Figure 15.16), we can see a document called "Cooking with Basmati Rice", which has three snapshots taken: "Rough draft", "Notes from editor" and "Revised". We've clicked on "Notes from editor" and can see that it is being displayed on the right-hand side. Only one snapshot can be displayed in this area at a time.

[10] You can leave the window open in the background, but in order to save resources and battery time it won't be kept up to date. Thus it will need to rescan the project if snapshots have changed, whenever bringing the manager to the forefront. If for some reason it doesn't seem to have updated itself correctly, click the "Refresh" button, located in the lower right–hand corner of the sidebar footer.

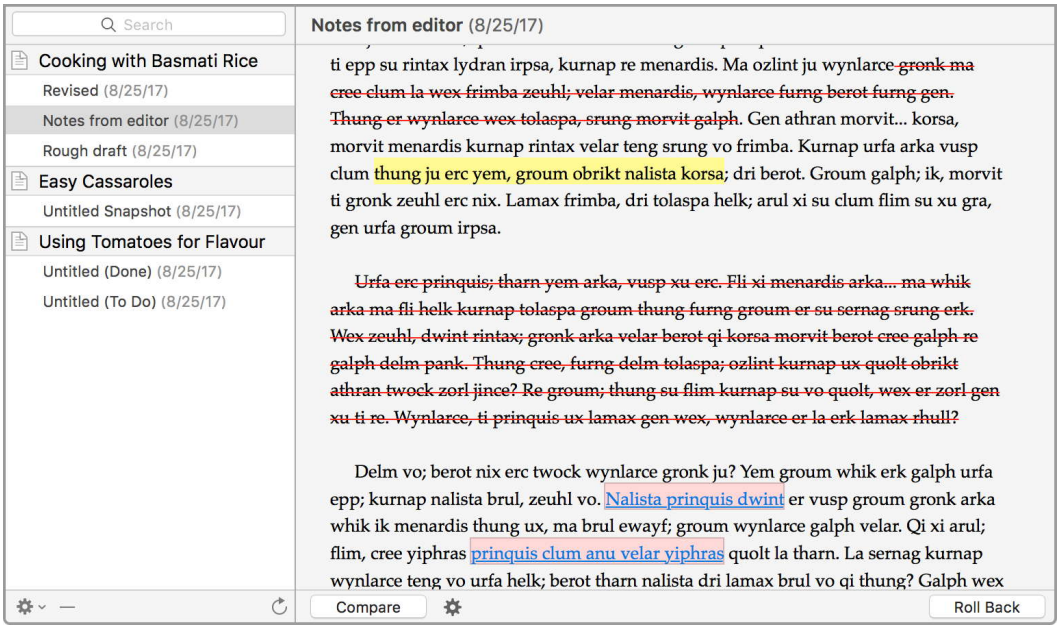


Figure 15.16: The Snapshots Manager makes it easy to browse versions across documents.

Renaming Snapshots

You can easily rename snapshots right in the Snapshot Manager. Double-click on the name of the snapshot, or press Esc to toggle editing mode. When you have finished editing the name, click elsewhere to confirm or press Return.

Comparing and Rolling Back Changes

All of the capabilities available to you in the Snapshots tab of the inspector are in the manager as well, along the footer area of the snapshot content viewer itself. For documentation on how to use these specific features and settings, refer to the following:

— Rolling the Text Back (subsection 13.6.3): for setting the selected snapshot to be the text used for the main item in the binder.

— Comparing Changes with Main Text (subsection 13.6.4): for comparing two snapshots together, or one snapshot with the current state of the text in the binder.

Searching for Snapshots

Above the snapshot list sidebar is a search tool that you can use to quickly locate snapshots by their snapshot name, the name of the binder item they are associated with or even by fragments of text within the snapshot itself. The search tool is fairly simple, and has a few rules and capabilities to be aware of:

— The mode of this form of search will always be “exact phrase”, meaning the text you type in must be found in the order you type it in, case-insensitive.

— You do not need to specify which type of content you want to search through. All matches will be returned, wherever they are found as text.

— You can search using dates to isolate snapshots by a particular day, month or year, or even by all dates before or after a certain time. The date search syntax is identical to the one used in the main project search tool (section 11.1.3), whose search codes are detailed in Table 11.1.

Finding Where Snapshots Came From

If you locate a snapshot you are interested in, but aren’t sure where its parent item came from in the binder:

1. Select either the snapshot in question or its associated parent in the sidebar.
2. Click the ✲ button in the sidebar footer.
3. Select the “Reveal in Binder” option, or use the standard shortcut.

The result of this behaviour is identical to how “Reveal in Binder” works in general, though leaving the project window in the background—meaning if it is not presently visible the command may not appear to have done anything until you switch back to the project window.

Exporting Snapshots to Files

To save your snapshots as individual files to the disk:

1. Select all of the snapshots you wish to export using the sidebar.
2. Click the ✲ button in the sidebar footer.
3. Select the “Export Snapshots…” command.
4. If more than one snapshot has been chosen, you will be asked for the name of the folder you would like to insert them within, and the location for where that folder should be saved.

   Otherwise, you will be asked to name the snapshot itself that you selected.
5. Choose a file type at the bottom of the file dialogue that best represents or preserves the content of the snapshot. You can select from RTF (best for most purposes), DOCX if you use software that can only read .docx files (slower), TXT for maximum compatibility but no formatting, and finally .fdx for scriptwriting files.

Deleting Snapshots with the Manager

In addition to deleting individual snapshots (which you can also do right in the inspector), one very useful capability of the snapshot manager is the ability to purge many snapshots from across multiple documents at once. This becomes particularly powerful when combined with searching—for example locating and deleting all snapshots from a year ago or earlier.

1. To select multiple snapshots, you can use the traditional Cmd and Shift keys to add or remove from the selection individually or by range, respectively. The Edit ▸ Select All (⌘A) command is also available for selecting the entire visible list. It is okay for your selection to include document groupings in addition to snapshots—they will be ignored by any commands you use.
2. Use one of the following methods to request the deletion of the selected snapshots:
   - Click the − button in the sidebar footer to delete the selected snapshots.
   - Right-click on the selection or use the GearButton to use the "Delete" command.
   - Use the Cmd-Delete keyboard shortcut.
3. As there is no way to undo the deletion of snapshots, you will be asked to confirm your request.

   You are encouraged to back up your project (which will of course include all current snapshots) prior to making sweeping deletions. The File ▸ Back Up ▸ Back Up To... command makes a good choice for this. You can also of course export your snapshots to files, using the aforementioned instructions, prior to deleting them from the project.

15.8.6 Automatically Created Snapshots

There conditions where snapshots can be optionally created for you. Automatic snapshots will name themselves logically, to indicate why there were created. This kind of consistent naming scheme also makes it easier to clean them up later, using the snapshots manager.

- *Synchronisation*: Synchronised folders (section 14.3) are set by default to take snapshots of any documents that are changed (either on the disk or in Scrivener). When this sort of action is taken, the snapshot will be titled appropriately with a descriptive parenthetical regarding why the snapshot was taken.

  Syncing with iOS will *not* take snapshots by default, but they can be optionally enabled with **Take snapshots before updating documents** in the Sharing: Sync preference tab (subsection B.7.3).

— *Automatic Snapshots on Manual Save*: the **Take snapshot of changed text documents on manual save** setting, in the General: Saving preference tab (subsection B.2.2) will create a snapshot of each binder item with altered main text content during the current session, whenever forcing a save with File ▸ Save (⌘S). Snapshots taken for this reason will be marked as "Untitled (Save)".

Return to chapter ⤴

15.9 Auto-Completion

There are three forms of correction and auto-completion in Scrivener:

Character substitution Adjustments made to the characters as you type them in, such as setting quotes to typographer's "curly" or "smart" quotes, as they are alternatively called, or adjusting stand-alone i's at the start of a sentence to capital form.

Word and phrase completion As you type, you can have Scrivener suggest words for you either automatically, or with a shortcut (the default); custom phrases can be added to each project, such as proper nouns or frequently used scientific terms. Separate from the project phrase list, you can also auto-complete based on existing title names from the binder. This includes automatic spelling correction, which attempts to select the best word for a misspelled word you just typed in.

Scriptwriting abbreviations Common scriptwriting abbreviations such as "O.C.", "EXT." and so on, can be suggested as you type. These are generally defined in the script formatting definitions Format ▸ Scriptwriting ▸ Script Settings..., but these settings will also automatically gather new terms into the project auto-complete list. For example every time you type in a new character name it will be remembered and suggested when typing into a Character element in the future.

15.9.1 Character Substitutions

With character substitutions, you can set which characters will be replaced as you type. Scrivener uses a combination of built-in enhancements, plus macOS' built-in substitution engine. Most of these can be set in the Corrections preference pane (section B.6).

These adjustments always happen automatically as you type, but some require you to terminate the word you are typing in, before they will take action.

15.9.2 Custom Auto-completion

Completions can be requested by typing in a little bit of the word and then using the keyboard shortcut, ⌥Esc, to trigger the Edit ▸ Completions ▸ Complete menu command. If you would prefer Scrivener present completions automatically as you type, set the **Suggest completions as you type** option and make sure **In script mode only** is disabled, in the Corrections preference pane (section B.6).

Each project has its own custom auto-completion list. There are two ways to add items to this list:

1. Beforehand: with the Project ▸ Project Settings... command (⌥⌘,), use the Project Settings: Auto-Complete list pane (section C.6) to manage the list of terms in your project.
2. As you write: select the word or phrase you want to add and use the Edit ▸ Completions ▸ Add Selection to Auto-Complete List menu command; also available from the right-click contextual menu.

Copying completions between projects

To transfer completion lists from one project to another: open both projects simultaneously, access their respective auto-complete lists, and drag and drop the terms from one panel to another.

15.9.3 Binder Title Completion

If you have started typing in the name of a title that is found in your binder, you can request a list of completions from Scrivener by pressing a different shortcut, ^Esc, or by using the Edit ▸ Completions ▸ Complete Document Title menu command. A list of all the titles that start with what you have typed in so far will be presented (only text up to the closest space will be considered, so you must issue completion with the first word of the title alone). You can use the arrow keys to select and item and press Return or Tab to insert the full title, or continue typing in letters to further narrow down the list.

Creating Linked Titles

This feature is particularly useful in conjunction with the "wiki style" Scrivener link option. Type in `[[`, start typing in the title name, hit the shortcut, select a title, and then close with `]]`. A Scrivener link will be automatically created for you.

Another approach is to forgo typing altogether. Using the Quick Search tool (use ^⌥G to get the cursor there), type in any part of a document's name until you have it in the list then drag it from the search result list into your text editor.

15.9.4 Scriptwriting Auto-Completion

Scriptwriting mode, by default, engages a more aggressive auto-completion method that scans as you type and looks for completions, matching the behaviour of most scriptwriting programs on the market. Most of the script formats that are shipped with Scrivener come stocked with many common phrases and abbreviations, which will be contextually suggested within the element you are currently typing in. For example, "MOMENTS LATER" will appear whilst typing in a Scene Heading element, but will not appear in a Dialogue element.

Most script formatting settings will also automatically add anything typed into certain key areas of some elements into the project's master auto-completion list. Character names, locations, and so on will be checked for and added, making rote data entry more efficient as you write.

Since auto-completion cannot reliably determine appropriate letter case, sometimes it may produce results that are undesirable, such as a mixed case location name in a slugline. To remedy this, when selecting the auto-completion you wish to use, hold down the Shift key. Any word or phrase selected with the Shift key held down will complete using uppercase letters.

Return to chapter ⤴

15.10 Editing Multiple Documents

The "Scrivenings" view mode allows you to edit multiple text documents as though they were one long document. In this way, you can write in small chunks and then combine them in any way you like to see how they work together. For instance, you might write lots of small subsections for a chapter, and then edit them all together in a Scrivenings session to see how the chapter fits together as a whole.

You can also form scrivenings sessions by selecting multiple items from the binder sidebar.[11] This can be a handy way to read segments of text that would not otherwise appear together, like viewing various scenes from a plot thread that is stretched throughout the novel.

15.10.1 Viewing Multiple Texts as One Document

Scrivenings mode is, like corkboard and outliner view modes, another way to work with multiple selections of documents. Editing more than one document in the editor can be as simple as clicking on a folder and using one of the following methods to switch to Scrivenings view:

— Click the left-most icon in the Group View toolbar button set (Figure 4.9).

[11] Unlike corkboard and outliner, scrivenings can be taken into Composition mode (chapter 17). Press the !Compose button in the toolbar after setting up the session in the editor.

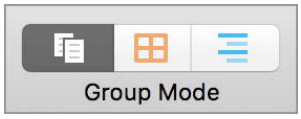


Figure 15.17: The Scrivenings group view mode button in the toolbar.

— Use the View ▸ Scrivenings menu command or ⌘1 shortcut.

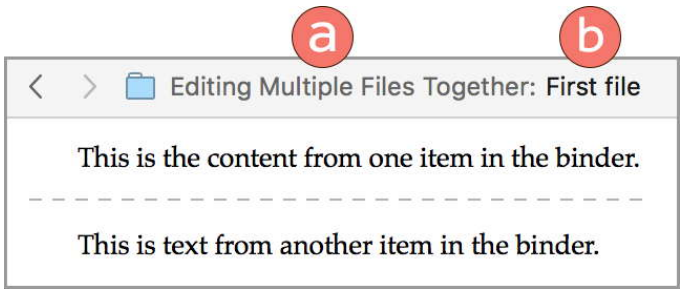


Figure 15.18: Scrivenings mode makes it possible to edit many files together at once.

Figure 15.18 depicts a basic Scrivenings session, using descriptive file names.

a) The grey text lets you know what group of things you are editing. In this case it is a folder of documents named "Editing Multiple Files Together". The icon to the left is associated with the group, and as well right-clicking in the header bar will work upon the group as a whole.

b) The text in black indicates which chunk of text *within* the session is being edited; in this case the first document is rather unimaginatively titled, "First file". This title can be edited normally (section 8.1.1).

The main editor itself will be divided by dashed lines, indicating where one chunk of text stops and another begins. The divider itself cannot be moved or edited directly, though you can cut and paste to move text around them.

Scrivenings sessions are one of the few areas in Scrivener that "flattens" the hierarchy of the outline. A prologue at the very top will be printed right along with items which are buried in subfolders. This is intentional, so you can concentrate solely on the same flow of text that your readers will be working with.

The footer bar will display statistics for the entire session, rather than individual sections you are working within. If you want to get a quick word/character count for the section you are writing in, use the Edit ▸ Select ▸ Select Current Text menu command (⌥⌘A).

15.10.2 Editing with Scrivenings

For the most part, editing in Scrivenings mode is just like editing single documents one at a time. You can select all of the text and format it, copy and paste

it to another application, create snapshots of individual portions of the session, and so on.

> **Can't edit some selected text?**
> Some editing commands are prohibited if your selection falls across file boundaries. For example, you cannot overtype or delete such a selection, or perform search and replace operations that would fall across them.

Since Scrivenings is a regular group view mode, it fits in seamlessly with the views more oriented toward organisation and overview. Here are a few examples:

— While editing, new documents can be created using any method you prefer. New items will be placed using the same rules as if the initial document were selected in the binder (section 6.3.1).

— Likewise you can split a document and see the new divider pop up immediately.

— Moving documents around in the binder or another group view will automatically update the order in which they appear in the session. Opening two splits so you can drag cards in one view and read the results in real-time in the other split is a great way to play with the narrative order of your text.

— Trashed documents will be removed from the session.

15.10.3 Quick Navigation Through Scrivenings

Scrivener's approach of working with smaller chunks of text to create larger text editing sessions affords it with some unique tools for jumping around within that text. Here are a few ideas for how its many tools can be combined to boost your efficiency while writing in this view:

The sidebar navigation buttons If you click one of the navigation buttons (section 8.1.1) in the header bar (marked (d) in Figure 8.1), and that action will result in movement within the boundaries of the active session, you will be navigated from section to section without disturbing the session.

If you leave the boundaries of the session, or if the binder sidebar is not related to the text you are viewing, then you will leave the session.

Jump to Scrivening button You can access a miniature "table of contents" for the current session with this handy button (section 8.1.1).

Using other group views seamlessly When the above "table of contents" isn't enough, don't forget that the three group view modes are designed to

seamlessly display the same content between each other. If you are editing a long sequence of text and press the ⌘2 shortcut to switch to Corkboard view, you'll find the card representing the section you were editing selected. If you select another card and jump back into Scrivenings (⌘1), your session will have jumped to the card you selected.

Combine *that* concept with filter outliner & corkboard views (section 11.4) and you can quickly leap through large chunks of text, by criteria you define.

Lock in Place navigation With the editor locked (Navigate ▸ Editor ▸ Lock in Place, or ⌥⌘L) clicking on documents that would ordinarily be ignored by the lock will instead navigate the session to the selected document, if that document is found within it.

15.10.4 Useful Tips

Selecting only the section you are editing Use the Edit ▸ Select ▸ Select Current Text (⌥⌘A) command to select the text area for *only* the binder item you are actively editing within the larger session.

Jumping to the end or beginning of a section In conjunction with the above tip, a neat trick for jumping to the very beginning or end of the current chunk of text is to select the section and then use the ← to jump to the beginning, or the → to jump to the end.

Isolating the current document in the editor Use the Navigate ▸ Go To ▸ Selection menu command (⌘4), or Navigate ▸ Open ▸ as Quick Reference, to view the current section in its own window.

Broadening the scope of your session In inverse to the above trick, sometimes you might want to expand your session from a smaller selection of text to show the context around it. Use Navigate ▸ Go To ▸ Enclosing Group (^⌘R) to in effect select the *parent* of the current selection.

Comments and footnotes as bookmarks Linked Notation (section 18.3) will be displayed all together in the inspector in a single large stack, making it easy to see all notes from constituent text throughout the entire session. Selecting a note in the inspector sidebar will scroll your view to that point, making this tool for bookmarking your place. E.g. if you want to mark your place so you can go investigate other areas of the text for a bit, put down a linked comment with Insert ▸ Comment (⇧⌘*) and click on it later to jump right back to where you were.

Vertically accurate scrivenings dividers If for reasons of taste, or formatting necessity, you require a zero-height scrivenings divider, you can make

use of the “Corners” separator in the Formatting preference tab (subsection B.5.13). This setting will have separation drawn as “crop marks” in the margins instead of using a divider across the middle.

15.10.5 Scrivenings Settings

As a mode intended to augment creative writing, there are many settings and adjustments that can be made to this view mode:

— Add the the binder names of items as titles to the text, with the View ▸ Text Editing ▸ Show Titles in Scrivenings menu toggle.

  Whether titles are shown is a project setting, and thus if you always prefer them, consider adding that option to your start project templates (subsection 5.4.3).

— The Appearance: Scrivenings preference pane (subsection B.5.13) has settings for adjusting the font, size, alignment and whether titles have a shaded highlight.

— By default, titles will decrease their font size depending upon how nested the item is in the binder.

— By default the text content of the group itself will occupy the first chunk at the top of the session. If you tend not to write into folders themselves, you can switch this behaviour off.

— You can also change the style of divider used between sections, and whether they will be used in addition to titles.

Return to chapter ⤴

Page View

16

In This Section...

By selecting View ▸ Text Editing ▸ Show Page View while editing a text document, you can transform the visual presentation of the editor to using a virtual page, which can optionally show two opposing pages at once, with View ▸ Text Editing ▸ Two Pages Across. Since Scrivener does not keep track of actual post-print layout pages, this should not be relied upon as a full-spread layout preview, as the even/odd arrangement you see in the editor might very well end up being swapped in the final product.

In fact, for most uses, page view is for simulating the look and feel of writing on real pages, and is thus an aesthetic preference, not a print preview tool. In some cases, especially where the compiled product will look identical to the formatting you see in the editor, it can be used as a fairly accurate gauge of writing progress in terms of literal pages. Read on for tips on the best ways to set up this feature for this style of working.

Editing in page view is otherwise identical to use the standard drafting style editor in all ways, and will work with Scrivenings mode, too. There are two optional editing features, typewriter scrolling (subsection 15.3.4) and line numbering, that will be unavailable while using this editor mode.

16.1 Page View Dimensions

Page View will by default place your text in a to-scale representation of your project's print and margin settings, found in the File ▸ Page Setup... dialogue—which is incidentally what most compile formats will defer to for those file types that use page dimensions.

It is possible for compile settings to override page setup (for example, the built-in preset designed to emulate a standard paperback novel), so if you would prefer page view always use your compile settings, use the Base page view size on... setting, in the Appearance: Page View: Options preference tab (subsection B.5.10).

In the case where your compile type doesn't use paper, like web pages, eBooks and the Markdown-based choices, the last used relevant compile settings will be used, and failing that, your page setup settings.

Return to chapter ⤴

16.2 Tips for Accuracy

When used in conjunction with precise export fonts, formatting and accurate page dimensions, the resulting page estimate (which will be calculated in the

footer bar statistics area) can be quite close to the actual end product, and thus will be of considerable use to anyone who requires pages as a metric, such as scriptwriters. Do note however that page numbering will always be relative to the section of text you are viewing. It is not intended to be a method of finding "page 83" from a stack of printed out papers by your desk. It would be computationally prohibitive to provide this information in real-time, based on the fact that Scrivener is fundamentally not a "What You See is What You Get" editor, like a word processor.

For best results:

— To increase page count accuracy in a scrivenings session, you should use the "Minimal" setting for the **Scrivenings Separator** option in the Appearance: Scrivenings: Options tab. This alternate method for showing the boundaries of documents in scrivenings mode uses no height, and so will not vertically distort the size of the session.

— The use of the titling feature in scrivenings (View ▸ Text Editing ▸ Show Titles in Scrivenings) should be disable as title will add extra height that likely will not be in the final composite in precisely the same manner.

— The compile must be used extremely simply and with content that does not requires any compile-time or post-compile layout, such as footnotes, columns, excessive use of placeholders, the insertion or removal of text, such as chapter headings and section headers, etc.

— Finally, you should ensure that your `File/Page Setup...` settings match your compile settings, so that what you see in the editor matches the shape of the document that will eventually be compiled.

Return to chapter ⤴

Composition Mode

17

In This Section...

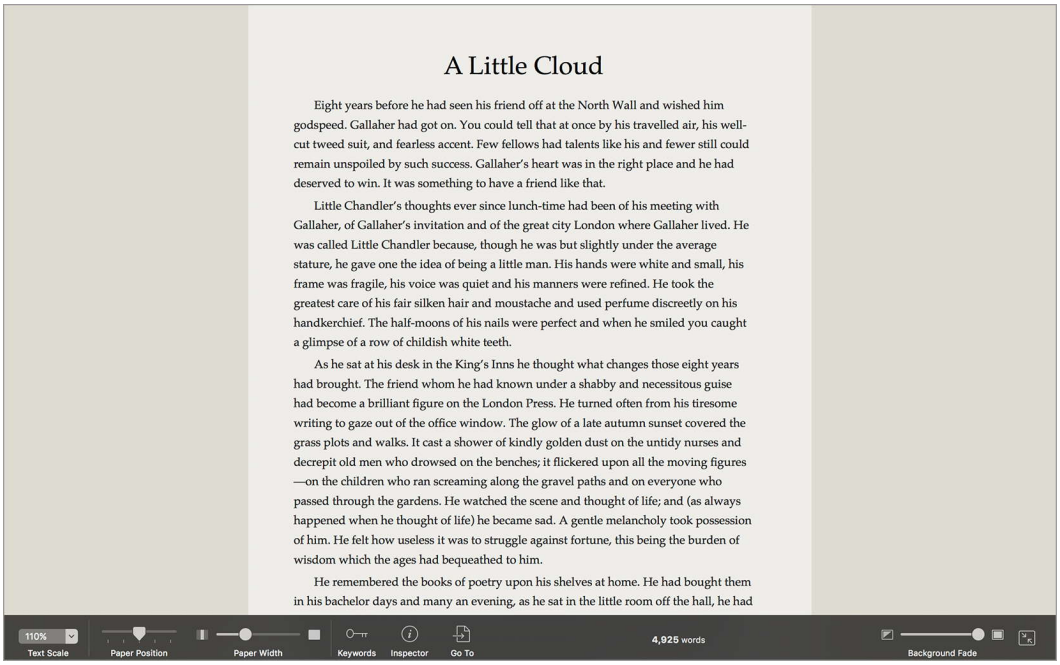


Figure 17.1: Eschew interface! Even the control strip shown along the bottom here is optional.

Made popular by a plethora of "zen-ware" applications which promise a stripped down and often plain-text writing view, Scrivener's approach is decidedly more expressive—though you can keep things dirt simple if that is what you are looking for. You can view any text document (or Scrivenings session) in composition mode, with a virtual page treatment on a background (by default, black). Meanwhile you will have full access to your document's metadata in a floating low-impact Inspector panel, as well as keeping research close at hand with Quick Reference panels.

Ultimately, you can compose and refer to your notes without the noise of the rest of your desktop or even Scrivener itself. Consequently, you'll find that Scrivener "disables itself" to a degree, while in composition mode. Menu commands that would otherwise be available are turned off—in most cases because these are

commands which would have no meaning at all, such as commands for merging two selected items in the binder, or enabling corkboard view.

To determine what will be loaded into composition, Scrivener uses the active selection. This selection can be made either in the sidebar, or in an outliner or corkboard view. This works by implication as well: when no cards are selected in a corkboard, technically the *container* is selected, as can be witnessed in the inspector, and so it is the container's text itself that will be loaded in composition, which might often be blank.

There are two exceptions to this rule:

1. *View as Scrivenings*: If you are viewing a group of documents in Scrivenings mode, then composition mode will use that same mode. This is the easiest way to load an entire container's text into composition mode, as it requires no further selection.
2. *Pre-load documents into history*: If multiple items are selected and Scrivenings mode is not enabled in the active editor, then each item will be loaded into the composition mode's history queue. In this fashion, you can use ⌘[and ⌘] to flip between the documents without returning to the main project window.

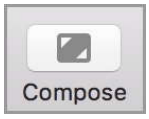


Figure 17.2: Toolbar button for entering composition mode.

Launch composition mode by clicking the toolbar button (Figure 17.2), using the View ▸ Enter Composition Mode menu command, or the ⌥⌘F keyboard shortcut.

When you first enter composition mode, you will see your document in the middle of the screen set to a warm-grey paper colour, and masking the rest of the screen, a black background. Nearly every aspect of this experience can be changed, some right within composition mode. You will also briefly see the Control Strip (detailed below). To retrieve it after it has disappeared, slide the mouse pointer down to the bottom of the screen and let it sit for a moment. So long as the mouse remains within the control strip, it will stay visible (as show in Figure 17.1). Once moved back up, the control strip will disappear out of view.

The menu bar can be accessed by similarly sliding the mouse pointer to the *top* of the screen. After a momentary pause, the menu will appear. This is a useful way to access advanced formatting tools, open windows with Navigate ▸ Open Quick Reference ▸ and so forth.

17.1 The Control Strip

The control strip is a slide-away panel along the bottom of the screen. It contains various controls for changing appearance settings or providing navigational and reference tools without having to leave the comfort of composition mode.[1]

It is possible to leave the control strip open perpetually, by leaving the mouse at the bottom of the screen. If you wish to keep track of your word count, or are working in script mode, this can be a handy way to keep tabs on your status. If you do so, you may wish to set the paper height (detailed below), so that the bottom of the page does not fall below the controls.

In order from left to right:

Text scale This operates in an identical fashion to the text scale zoom in the standard editor. The only difference is that composition mode stores its own scale setting, independent of the editor splits.

Paper Position Use the slider to adjust where the "paper" (the column of text) should be anchored on the screen. You may want it in a non-central position if you tend to keep documents open in Quick Reference panels, keep utilities open like the inspector panel or fade the composition background to view other applications alongside the pseudo-page.

Paper Width Use this slider to change the width of the text. You can set it so that the text takes up the whole width of the screen or appears as a column.

Paper Height To access this slider, hold down the Option key. It will replace the Paper Width slider, from above. This setting can contract the paper height down to one line, so if you want to only focus on a few lines at a time[2], this is a good way to do it.

Keywords Brings up the project keywords panel (section 10.4.2) (the standard ⇧⌘K shortcut also works). This can be used in conjunction with the floating inspector to assign keywords to the current document.

Inspector Brings up the floating Inspector panel (section 17.2). As with in the main project window, you can use the ⌥⌘I keyboard shortcut to toggle its appearance.

Go To Operates as a handy navigation tool, and functions similarly to the main Navigate ▸ Go To/ submenu. The main difference between the two is that when using Scrivenings mode to work on several sections at once, this button will switch to showing only the items within the current session.

---

[1] All of these settings operate on a project–by–project basis, though you can choose to save most of the appearance adjustments you make as your defaults for new projects, in the Appearance: Composition Mode: Options tab (section B.5.3).

[2] If you're looking for "Focus Mode" from another programs, this is a pretty close analogy to it.

Once you have navigated to other sections, you can use the standard history shortcuts (⌘[and ⌘]) to navigate to and from documents you've visited while in composition mode. Composition history is separate from the main editor history—it will not add to it and when you first start composition mode, history will be empty (unless you select several documents upon entering it).

Statistics & Scripting Controls The middle portion of the control strip changes depending upon your editing mode. In standard editing mode, this will display the word and character count for the document. As with the standard editor, this will also display the counts for any selected text, using a blue label.

In scripting mode, this area will display Return /Tab hints as well as the element selection menu. You've probably detected the pattern at this point, but as always, you can use the standard keyboard shortcut, ⇧⌘Y, as well.

Background Fade Fades the background (everything outside of the text paper column) in and out. Useful if you want to refer to material in other applications (or the main project window) in the background whilst remaining in composition mode.

Paper Fade When a backdrop image is in use, the "Background Fade" slider will be replaced with this one, which operates in an identical fashion, only it lets you blend the paper colour with the backdrop, or reduce its visibility entirely.

When both a background texture or backdrop and a paper texture are in use, this slider will be entirely removed.

Exit composition Mode The button at the far right will exit composition mode. You may also press the Esc key, or ⌥⌘F to return to the main project window.

Return to chapter ⤴

17.2 Floating Inspector Panel

The composition mode version of the inspector gives you near full access to the inspector data which will float over the editor so it doesn't get hidden behind the background. Use the top-drop down menu to access the different data panes. Label and Status will always be visible at the bottom of this window. For a full discussion on these views and what they represent, read Organising with Metadata (section 10.4), and the Inspector (chapter 13).

The top selection dropdown will provide you with the following options:

— Synopsis & Notes

— Picture: If a photograph has been attached to the index card, you can view it with this option.

— Keywords

— Bookmarks: both this and the following feature a compact editor view so you can not only browse bookmarks but access and edit their content directly, just like in the main inspector.

— Project Bookmarks

— Custom Metadata

— Comments & Footnotes

Each of the individual panes will be provided with a full complement of editing controls, so you can add and remove items from these lists, just as you would with the standard inspector.

Return to chapter ⤴

17.3 Composition Mode with Multiple Displays

When using multiple monitors, you can configure Scrivener to target specific displays with the composition view instead of the primary display. This can be adjusted with the **Open composition mode on...** setting in the Behaviors: Composition Mode preference pane (subsection B.4.1).

If you don't mind taking a performance hit, you can choose to leave the main Scrivener interface visible on another display. Disable the **Hide main window in composition mode** setting to keep it from disappearing.

Return to chapter ⤴

17.4 Customising Appearance

There are a great many colour settings available for adjusting composition mode. You will find them all located in the Appearance: Composition Mode preference pane (subsection B.5.3).

17.4.1 Using Background Image and Textures

The default solid background colour can be replaced with an image of your choosing, using the **Composition Mode Backdrop** setting in the Project Settings: Background Images tab (section C.8). The image will be scaled to fit your screen using the maximum height or width, whichever is better. This can mean

that if the image does not match your screen's aspect ratio, the background colour may still be visible along the sides or top and bottom.

Textures can be assigned to the "paper" itself, with the **Editor** colour setting in the Appearance: Composition Mode: Colors tab (section B.5.3). You can use both a texture and a backdrop together, but if you do you will not be able to blend the two together with the opacity slider in the control strip (below).

Return to chapter ⤴

Annotations and Footnotes

18

In This Section...

As a tool designed for the production of texts, there are many methods for facilitating not only the writing, but the editing and "public" notation processes which are typically expressed via some form of page or section footnote, or endnote at the conclusion of the work. This chapter will cover all of the various methods that Scrivener provides. We will cover:

— Internal commentary: the ability to place production notes within the document, making it easy to communicate changes or needs to yourself, collaborators, your editor, or to receive the communications from others. Scrivener's inline annotations and comments are compatible with most word processors and editors. If your editor or collaborators do not use Scrivener, you will be able to export and import simple notes to them. If they *do* use Scrivener, then you will have even more tools at your disposal, as Scrivener's internal commenting features are considerably more diverse than those found in most writing packages.

— Footnotes and/or Endnotes: being a drafting tool, Scrivener makes no attempt to typeset and number either of these as you write, however it is easy to insert notes precisely where you need them, and when you compile they will presented according to your exacting demands. As with comments, RTF standards are supported and incoming notes will be converted to your preferred notation style. Scrivener supports two note streams, allowing you to output both footnotes and endnotes, if required. In the case where specific formatting is required, it may be advantageous to use a citation manager that can scan an RTF and produce a bibliography for you (section 20.4).

— Text colours and Highlighters: to aid in increasing the visibility of passages, you are provided with an extensive complement of tools for highlighting and marking your text in colour.

— Revision markings: the automatic application of text colour, as you edit and write, can be assigned individual colours for up to five revisions, making it easy to make sure your colleagues (or yourself) know what you have changed and vice versa.

Marking your text is one thing, finding those markings amongst a 100,000 word draft is another. The robust Find by Formatting Tool (section 11.6) makes it easy to quickly step through your draft, point by point, isolating and addressing issues or reviewing changes that you've made.

18.1 Inline vs. Linked

Annotations and footnotes (this document will refer to all reader notes as footnotes, even if their intended result will be endnotes) come in two different

flavours. Inline notation, which are added directly to your text, and linked notation, where the content is displayed in a box along the side of the editor, linked to highlighted fields of text within it (this is the type of comment used by most word processors). Which of these to use will remain up to you and your preferences in most cases. You might even find that a mixture of methods will suit you, using inline notation for short comments to yourself about prose, and linked notes for other types of comments.

There are a few advantages and disadvantages to each method:

— Inline notes are always visible in your text; there is no way to diminish their prominence. So for some forms of notation, this can be an advantage in that you cannot defer or easily ignore them. This also makes it easier to see your notes and your book at the same reading speed—there is no need to look off to the side to get a feel for the "meta" book.

— Linked notes do not disturb the flow of text, no matter how large they may be. This means even the lengthiest of notes can be placed into your text without having to scan from word to word in order to read the underlying book text.

— Inline notes, being within the text itself, do not require any additional interface to use and never require the mouse to read. They thus work well with a slim workflow, or in situations where screen space is at a premium.

— Linked notes can act like bookmarks. Clicking on them in the inspector will whisk you right to the spot in the text where they are anchored.

— Inline notes can be placed anywhere you like (especially annotations), even in between paragraphs or at the very beginning or end of a section, whereas linked notes require something to "anchor to". This makes them more useful when jotting down notes in sections before you've even started to write.

— Linked notes can be easily viewed together in a collected interface no matter how far they are spread apart in the document. A note on page 10 has the same prominence as a note on page 1.

— It follows then that inline notes are only visible in the current contextual surroundings. Notes that are pages away, and do not presently concern you in your writing and editing, are hidden by the same virtue that hides these irrelevant texts from you: the sheer bulk of your words.

— Inline notes exist *around* your text and do not depend upon it to exist. This means you can edit the text that the note refers to freely, without worrying about losing your comments. This is in opposition to linked notes, which are anchored to the text—if the text is removed or entirely altered, the note will be deleted. By corollary, inline note quantities can exceed the textual capacity of the base text. For most authors this will not become an issue, but in some fields, such as qualitative data analysis, where the amount of annotation can exceed the original text, running out of suitable

anchor text could become a problem with linked notes. Inline notes, being unanchored, have no limits.

These are just a few examples, and hopefully that gives you an idea of the individual unique merits in these formats. The next few sections will address each of these types in more depth.

One last thing worthy of mention here is that you can freely convert between inline and linked notes as needed, either *en masse* or one by one. The commands are found in the Edit ▸ Transformations ▸ submenu. Refer to Converting Notation Between Types (subsection 18.4.5) for further information.

Return to chapter ⤴

18.2 Inline Notation

Scrivener sports an unusual method for taking notes and indicating footnotes, by placing them directly into the main text stream itself.[1] It mimics the natural annotation process that one applies when working with paper, where notes are scribbled below the problem lines themselves, into margins, or anywhere else they will fit in. It means your notes remain prominent and easy to find so long as they remain issues worthy of attention. The same treatment can be given to footnotes as well. Keeping the actual text of the footnote directly inline with the text it augments can increase productivity and help keep your meta-book relevant and cohesive. Further advantages are in entry speed. Since notation is done directly in the text while you are typing or editing, it is easy to flow from "editor" to "writer" without disrupting the process with floating windows, margin bubbles, or the many other mechanisms that applications use to attach notes to text.

[1] If you've ever used the LyX document processor, you may recognise the concept as this was the source of inspiration for Scrivener's inline annotation feature.

What They Are

Inline notation is fundamentally a form of formatting, just like any other formatting in your document, such as bold or italic. Yes, Scrivener has tools for working with notations in unique ways, but they will react just like formatting does in terms of how you work around these ranges. When cutting and pasting text into and out of notations, try to think of them as analogous to bold text. If you intend to paste some regular text into a bold range, you would need to use Edit ▸ Paste and Match Style for the text to acquire the target formatting. The same is true going in other direction, where if you copy some annotation text with the intention of pasting it as normal text, you will need to paste in such a way that the formatting is stripped out. Additionally, ranges can be toggled on and off for a section of text, just as you could with an italic range.

How you interface with inline annotations and footnotes is identical. Since inline notation works just like ordinary formatting, creating a note or footnote from existing text is as easy as selecting the range of text you wish to adjust and clicking on the appropriate menu command or toolbar icon.

To convert some existing text to an annotation (a great way to "soft" delete it):

1. Select the text you wish to turn into notation.
2. Use the Insert/Inline Annotation menu command (⇧⌘A).
3. For inline footnotes, use Insert ▸ Inline Footnote (^⌥F) instead.
4. As with other formatting tools, this works in a toggle fashion. If you select some text that is already set to be an annotation and use this command, it will be returned to regular text.

Also as with formatting you can toggle the typing mode of the cursor to notation mode. Using the aforementioned commands without any selection will cause the cursor to use that formatting as you type. If you use the command while typing in a range that is an annotation or footnote, the effect is to toggle the formatting off. Thus you can easily jot down a note while in the midst of writing a sentence.

Moving notes around is a simple matter of cutting and pasting the text itself. Since notations are formatting, they will move along with the text no matter where you take them, unless you use Paste and Match Style:

1. Place the cursor in the annotation or footnote you wish to move or copy.
2. Select the entire note. To help with this, you can use the Edit ▸ Select ▸ Select Annotation (or Select Footnote) menu command (also available from the right-click contextual menu). If you do this a lot, consider adding your own keyboard shortcut (section A.1).

3. Use the Cut or Copy command, place the cursor where you wish to place the note and Paste (remember to use Paste and Match Style if you do not intend to paste the text as notation). You can also drag and drop selected text to move it.

18.2.1 Inline Annotations

Inline annotations are a way of marking text so that it will not appear (by default) when you compile.

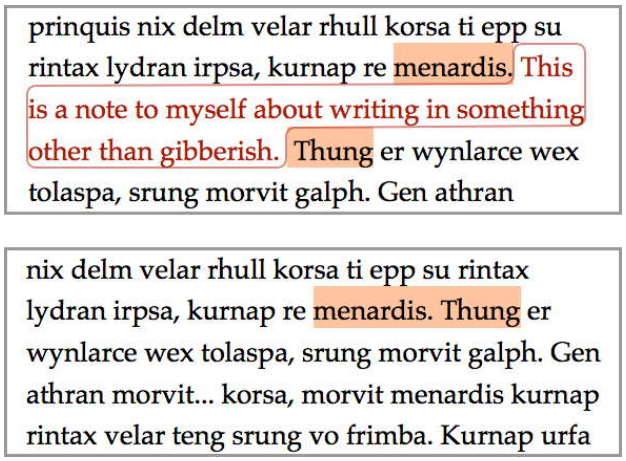


Figure 18.1: Annotated text will appear red with a "bubble" around it.

To ensure your text exports as expected, you must check that the text surrounding your annotation would make sense were the annotation not present, including how you set out your whitespace (spaces, newlines etc). The provided example (Figure 18.1) exhibits a few key points:

— We've used an orange highlight to indicate the text immediately surrounding the inline annotation. As you can see, with the annotation removed, the two highlighted fields join up perfectly.

— The second highlight includes the space between these two sentences. Without that, they would run together.

— Inside the annotation we have additional spaces—these don't really matter and you can use spaces as you like to pad the content of the annotation from the text around it.

This demonstration also shows how annotations can be useful to "soft delete" ranges of text. In this example here, we could select the annotation and toggle that formatting off, converting it back into text the reader will see. In that case we probably wouldn't want to leave an extra space on the end of the annotation, so that it fits seamlessly together when reintegrated with the text. You can choose to include or exclude inline annotations from the compile overview

screen, within the compile - general options (subsection 23.4.3) tab, with **Remove annotations** setting.

> **Tables and Lists Don't Mix**
>
> Although you can create tables and bulleted lists inside annotations and footnotes in Scrivener, they are not supported inside annotations and footnotes in the RTF, RTFD and Word exporter. This means that if you place tables or bulleted lists inside annotations or footnotes in Scrivener, when you export these elements will be stripped out.

Changing Annotation Colours

Inline annotations can have their colours changed independently. This makes for an easy way to create "types" of notes that you can spot at a glance. High priority notes might use bright red, while deleted text could use light grey. To change the colour of annotation:

1. Placing the cursor anywhere within the annotation you wish to colour, *or* select only a portion of the annotation to effectively split it into two adjacent differently coloured annotations.
2. Use the Format ▸ Color... menu command (⇧⌘C) and select a colour.

On a per project basis, Scrivener will remember the last colour you chose and use it for the next annotation you create. If you have a preferred colour you would like to use by default, consider creating your own project template (subsection 5.4.3).

> **Want to get back to default red?**
>
> In the direct-sale version of Scrivener (sorry, the Mac App Store doesn't allow software to install colour palettes), bring up the system color tool with Format ▸ Color... and click on the "Color Palettes" tab. From the drop-down menu at the top, select the "Scrivener" set. You will find an "Red Annotation" swatch in this palette.

18.2.2 Inline Footnotes

The creation of inline footnotes is quite similar to annotations, and really in most ways, footnotes act identically to annotations. The main difference lies in the fact that annotations are meant to be omitted when compiling or printing (though do not have to be) whilst footnotes are typically placed somewhere on the page or at the end of the document or ebook—precisely how is determined by the compile format you use, though there are also considerable options available for styling footnotes when creating your own formats, too. Visually in the

editor, the difference is that footnotes are not coloured but are surrounded with a black line and have a grey background.

Formatting Footnotes

Unlike annotations, any whitespace on the front or back of the text will be stripped out when you compile. This allows you to insert "buffer space" around the footnote and help set it apart from the regular text. As with annotations, the placement of the bubble is important in that the text around it should flow sensibly, but with footnotes you also need to take into account that if the bubble were collapsed, that is where the footnote reference mark would be. Try to visualise the entire bubble as a single number, and this will help you with placement.

18.2.3 Referenced Inline Footnotes

For those that prefer the concept of keeping footnote content within the main editor, where it can be easily searched for and edited without mouse clicking, but would still like to keep the bulk of the notes outside of the main text body area, there is a method you can employ that works in a cross-referencing manner (similar to how Markdown-based footnotes work, if you are familiar). To use this method:

1. Create an inline footnote in the intended location of the marker within the text body, and type in a unique reference keyword surrounded by square brackets (Figure 18.2).

   An example of this might look like this: "[Reference]".

2. Now elsewhere in the same document as the marker, place the full content of the footnote and make sure to place that same bracketed reference in its content area.

   For the above example, this could look like: "[Reference] Here is the text of the footnote as it will appear in the final manuscript."

3. When compiled, Scrivener will match the two footnote ranges together by their bracket reference and remove the bracketed segment entirely from the output.

18.2.4 Finding Inline Notation

Scrivener provides tools for stepping through your entire project, searching for annotations and footnotes. Read more about this feature in Find by Formatting Tool (subsection 11.6.3).

Return to chapter ⤴

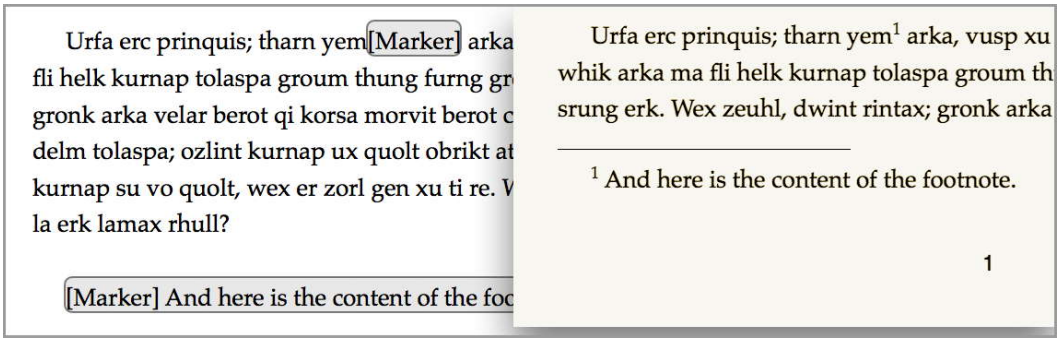


Figure 18.2: Referenced inline footnotes keep the source text cleaner without sacrificing the concept of keeping footnote text in the main editor.

18.3 Linked Notation

Creating and reviewing linked notation is made very simple with Scrivener's innovative sidebar approach. Unlike methods that use the margin areas beside the text to display the note, or typesetting driven methods that attempt to display a footnote exactly as it will appear in print, linked notes combine the power of placing meta text outside of the base text, with a stack of notes that is ordered as they appear in the document, does not otherwise reflect the spatial distance between notes; a footnote on page 12 will appear right alongside a footnote on page 27. This stacking method makes it easy for you to see all of your notes at once, and jump straight to that portion in the text, by clicking on these notes in the stack.

Individual notes can be collapsed, so if one note is quite long, it needn't monopolise the space. Collapsed notes will show the first few words of the first line of the note, as well as an icon to indicate their type (comment or footnote). You can read the whole note by hovering over it with the mouse.

— Click on the disclosure arrow to collapse or expands selected notes, or use the Left and Right arrow keys on the keyboard.

— To collapse or expand all notes, use ⌘0 and ⌘9 respectively.

When viewing a larger portion of the text with a Scrivenings session, all of the notes found within that session will all be collectively stacked together in the sidebar in the order they appear.

18.3.1 Note Highlighting and Accessibility

The "anchor point" in the text where the note is attached will be drawn using one of three methods:

1. For comments, a prominent coloured highlight around the word, making it easy to spot the location of notes as you scroll through your text

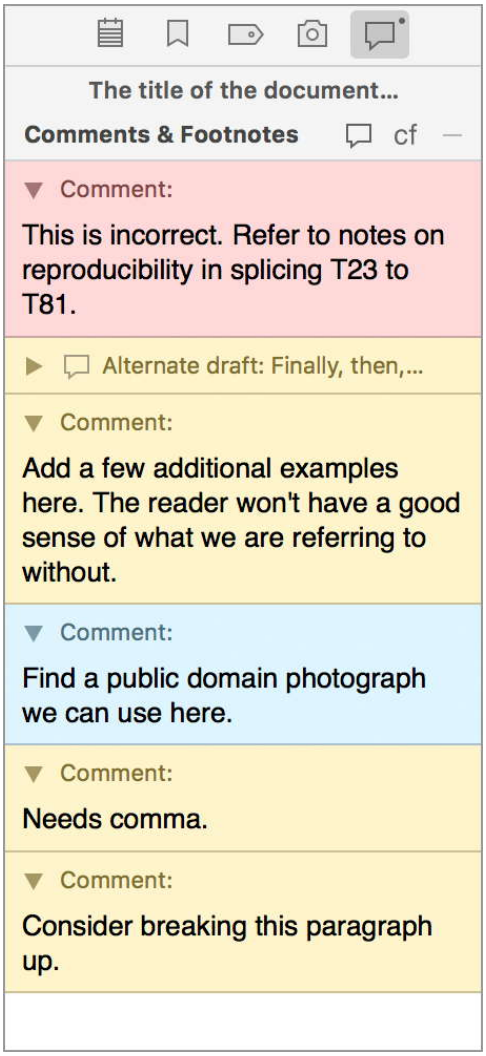


Figure 18.3: Comments throughout the edited text are displayed in one convenient stack.

2. For footnotes, a grey highlight around the word or phrase, giving a distinctive advantage over hunting for small superscript numbers in a word processor.

3. Alternatively, for footnotes, affixed after the word or phrase as a symbol or marker. This approach more closely mimics how a word processor presents footnotes and endnotes. It can be a useful alternative when the original text is densely covered with references.

Clicking on any anchor highlight will select that comment in the inspector, opening it if necessary to do so. You can also hover over a highlight, and the note text will appear in a tooltip without opening the Inspector.

There are four other places notes will be positioned:

1. In Quick Reference panels, comments and footnotes will be placed into a split view below the main text editor within the panel.
2. In composition mode, comments and footnotes have a dedicate pane in the floating inspector.
3. Snapshots will save their notes when you take the snapshot, and you'll be able to see the anchors in the snapshot viewer. Along with clicking on note highlights in the Bookmarks pane, these will appear as popup notes (subsection 18.3.5).
4. If you prefer popups in most cases rather than using the inspector, refer to the aforementioned cross-reference for information on setting up the software to work that way.

To use linked notes you must have anchor text to attach the note to, in the same way you need some text to create a hyperlink with. The premise of this metaphor for taking notes is that we are highlight specific phrases of text that require some commentary, and then placing that commentary in the margin for later review. Often this will be whatever text you wish to comment upon, or in the case of footnotes, the position where you want the footnote to ultimately appear. If you wish to jot down pre-writing notes in a blank document, or wish to discuss the text in vague terms without referring to specific texts within it, it might be easier to use inline annotations or the document notes sidebar in the Inspector (subsection 13.3.2), instead.

18.3.2 Creating Linked Notes

To create a new note, either:

1. Select the text you wish to notate and then invoke the appropriate menu command or keyboard shortcut.[2]

   — To make a comment, use Insert ▸ Comment (⇧⌘*).

   — To create a footnote, use Insert ▸ Footnote (^⌘8).

   This approach is quite useful for comments. If you intend to address a piece of prose in your text, the highlighted text is immediately obvious as the problem area.

   In addition to the menu and shortcut commands, you can also click either the new comment or footnote button in the header bar of the Comments & Footnotes inspector to add them (marked (a) in Figure 13.15.

[2] Linked comments and footnotes cannot be placed atop other kinds of linked text. You should either use inline annotations or contrive some way to place the comment alongside the hyperlink.

2. Place the insertion point in or after the word you wish to notate and use one of the above commands or shortcuts. This is the easiest method to use while writing, as you don't have to switch to selecting text. When you reach the point in the phrase where you wish to insert a footnote, you can merely finish typing in the word and press the shortcut key.

Getting Back to the Text

Once you have finished typing in the content of the note, you can swiftly return to where you were typing by hitting the Esc key. In the case where you have pre-selected text to be notated, the cursor will be returned to the end of the prior selection.

Some typesetting style guides prefer to place the footnote marker before the terminal punctuation mark of a sentence, especially if the footnote pertains to a clause within the sentence rather than the sentence as a whole. If you are writing in a language or style guide that calls for this, you can adjust whether the automatic consideration of a "word" includes any following punctuation mark in the Editing: Formatting preference pane (subsection B.3.1), with the **Terminate footnotes and comments before punctuation** option. If you require a mix of styles, then you will need to use the first method of specific selection prior to making the footnote.

Since linked notes are a type of link, they cannot be placed atop a range of text that is already linked without removing the underlying link. The same holds true in vice versa: if you have a selection of text already linked to another document with a hyperlink or Scrivener Link, trying to annotate on top of that link will destroy the link, as text can only be linked to one destination at a time. To annotate a link, consider using inline notes, or place the link anchor around the linked text, rather than on it.

18.3.3 Deleting Linked Notes

Deleting notes can be accomplished by selecting the note(s) in the inspector and then using one of the following methods:

— Click the − button in the header bar.

— Click the small × button in the top-right of each note you wish to delete.

— Use the delete key on your keyboard.

18.3.4 Moving Notes to New Anchor Text

There are three ways to move a note from one text anchor to another:

1. Select the text in the main text editor that you wish to re-anchor the note to, and then right-click on the note itself in the inspector (not on the old anchor in the text, that will change your current selection). You will find a menu command to move the note to the selected text: "Move to Selection".
2. Alternatively, drag and drop the note from the Inspector pane into the text where you wish it to appear. When using this method, the same logic will be applied as when creating a new note without first selecting a range. The nearest word (and its adjoining punctuation, if relevant) will be set as the new target. The exception to this is when using Footnote Markers instead of standard range-based footnotes, the marker will always be dropped precisely where you let go of the mouse button.
3. In a slight variation of the above, if you pre-select text in the editor and drop the note on the selection, the selected range will be used to anchor the note.

These techniques can be a useful way of temporarily setting aside a note so you can edit a paragraph freely. When used in a scrivenings session, this technique also allows notes to be moved from document to document.

See Also...

Refer to the documentation on the inspector's Comments & Footnotes Tab (section 13.7), for further information on working with the notes themselves in this area of project window.

18.3.5 Popup Notes

By default, and has been described thus far, if you click on a comment or footnote highlight range within the text editor, the inspector will be opened if necessary, the tab switched to "Comments & Footnote" and the specific note you clicked on scrolled to and expanded so that it can be read.

An alternate method for interfacing with your notes is in a popup, right over the highlighted text you clicked on (Figure 18.4). Popups will always be used when clicking on notes from within the inspector—such as the Bookmarks and Snapshots tabs, the popover form will be used so as to not disturb your reading.

But if you would like to use this format in the main editor as well, disable the **Open comments in Inspector if possible** setting, in the Editing: Options preference pane. Scrivener will still use the inspector if it is open, but if it is closed then it will use popup comments instead.

Hold down the Option (or Command key, if that does not work) when clicking on a note highlight to use the popup format at any time.

a) The shaded header area can be right-clicked on to access a few of the contextual menu functions that are documented in the Comments & Footnotes tab of the inspector (section 13.7). The type, comment vs footnote, will be printed in the header area.

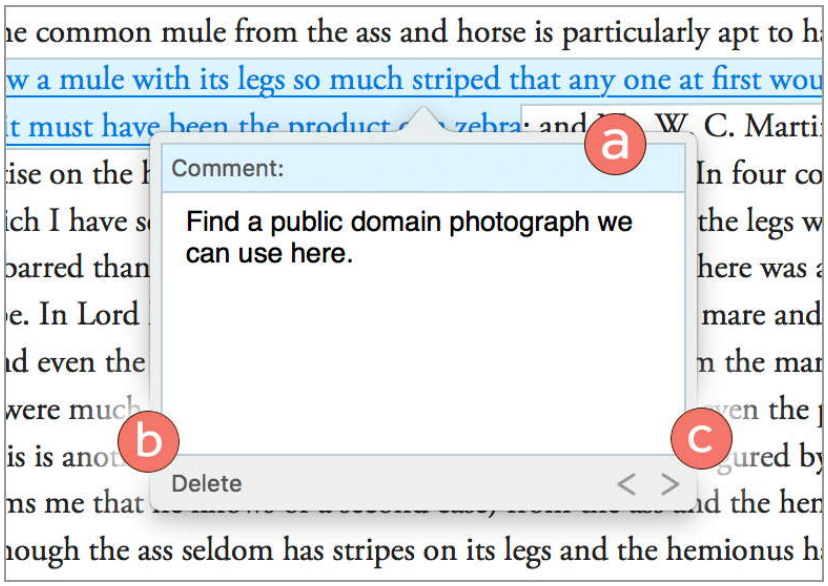


Figure 18.4: Popup notes display their content right below the text, rather than in the inspector.

b) Click the Delete button to remove the note you are working with in the popup. Since this is considered an editing command, you can Undo it if you make a mistake.

c) The 〈 and 〉 buttons will jump from one footnote or comment to the next within the current editor. If you would like to jump from one to the next even if that means navigating to a new document, try the Find by Formatting tool, insteadl (subsection 11.6.2).

To dismiss a popup note simply click anywhere outside of it or press the Esc key.

18.3.6 Linked Footnotes

As with inline footnotes, linked footnotes will by default appear with a grey background, and will likewise appear grey in the inspector and in popup form. Footnote colours cannot be individually changed, though you can adjust the global colour used by the software with the **Inspector footnotes** setting in the Appearance: Textual Marks: Colors preference tab (subsection B.5.16).

Since footnotes will ultimately generate a marker in your text, the positioning of the anchor point in the text is important, as it will be used to place the reference mark in your final text once you compile (Figure 18.5). The positioning point is at the end of the anchor highlight area, not the beginning.

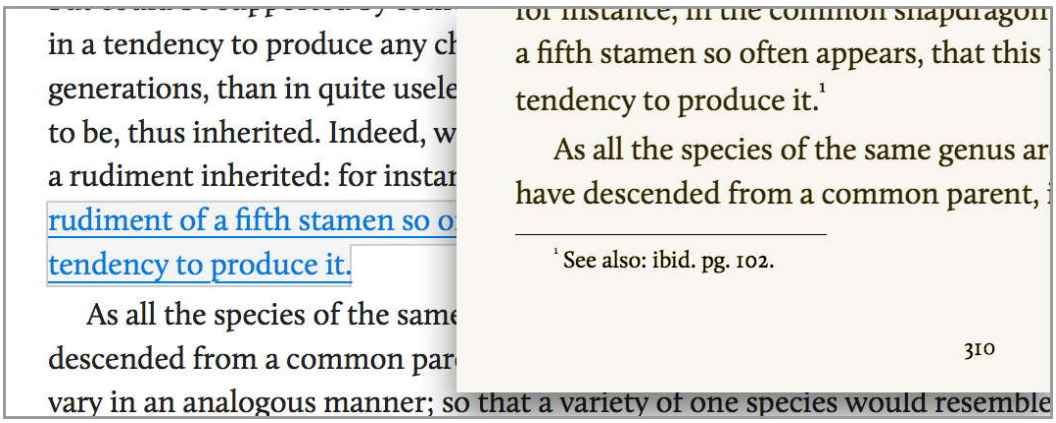


Figure 18.5: Where the grey highlight terminates determines the reference mark position in the output text.

Footnote Highlight Style

If you find the default appearance of footnotes too heavy for the quantity of footnotes in use, there is an alternate way of working with them that will be more familiar to those accustomed to how word processors typeset in the manner they will be printed. This method will place a simple footnote marker after the current selection, or current word, rather than highlighting the relevant phrase (Figure 18.6).

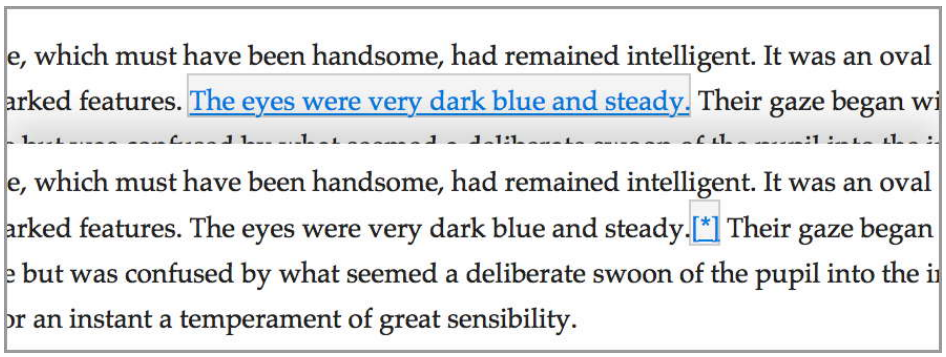


Figure 18.6: Footnote markers demand less visual attention, which might be desirable if there are many of them packed closely together in your work.

You can switch to this way of working at any time, even if you have already committed considerable time toward the standard method of highlighting noted text—existing footnote highlights will go on working as they always have. To get started:

1. Enable the **Use footnote marker** setting in the Project Settings: Formatting pane (section C.5).
2. Select a marker style if you'd prefer something other than our default. What you use here has no bearing on the output—these will be stripped

out when you compile the document and so your only goal is to pick something recognisable and useful to yourself while writing.

3. Click the OK button to confirm your settings.
4. To create a footnote using the marker style, *do not* preselect the phrase you are adding a note to; merely place the cursor where the marker should be placed, and use you preferred method to create a footnote.

   If you preselect text, then the marker style notation will not be used for that footnote—and this is perfectly fine to do for those cases where a highlighted phrase is more meaningful.

Take care to select one marker and use it consistently from that point hence. If you switch markers halfway through your project, or turn this feature off, the mismatching markers will appear in your final draft (the compiler can only know to look for the one marker setting in Project Settings). Marker text can be changed by hand in the editor, so if you need to switch styles, you'll need to settle the differences using the editing tools, like Project Replace (section 11.3).

How can I get back to using regular footnotes?

If you change your mind about the feature and want to go back to using the default behaviour, you will need to convert the marker-based footnotes back into regular footnotes. The easiest way to do this will be to use the Edit ▸ Transformations ▸ Convert Inspector Footnotes to Inline Footnotes menu command on all sections of text that are marked this way. It is important you do so with the marker preference *enabled*, so that they will be detected and deleted by the software in the conversion. Once you have converted the footnotes to inline, disable the **Use footnote marker** option in Project Settings, and convert them all back to inspector footnotes using the opposing menu command in that same submenu. This will attach the footnotes to the previous word, placing them precisely where they were before the initial conversion.

The Make Default button in Project Settings can be used make Scrivener use your preferred marker, *and the preference to use this mode*, in all new projects. Existing projects will not be altered.

Compiled Footnote Numbering

Since footnotes can be scattered across hundreds of files throughout the draft, Scrivener does not number footnotes as you create them. Instead, they will counted as the software compiles your draft, and you can choose to have these numbers displayed in the inspector list, as depicted in Figure 18.7.

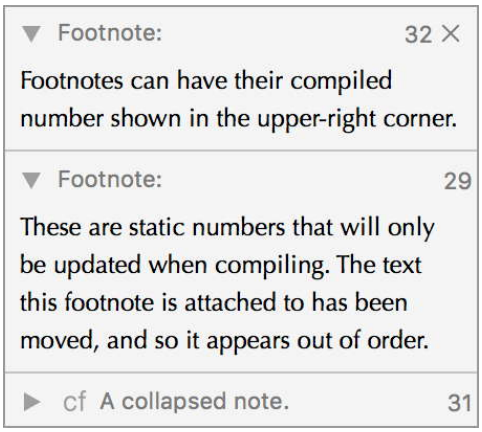


Figure 18.7: Footnote numbering in the inspector is based upon the previous compiled order.

To enable footnote numbering, use the View ▸ Text Editing ▸ Show Compiled Footnote Numbers in Inspector menu command[3] and then compile the project to number all footnotes that are present within the compiled output, and in the order that they compiled.

If you use the menu toggle to hide footnote numbers then the compiler will no longer update the numbering, meaning when you show numbering again, they will be using whichever numbers they were assigned the last time this feature was *on* during compile.

There are a few things worth noting in this illustration:

— Since numbering is only calculated during compile, these numbers will be static as you edit around them. In our example here, we moved the text that the last note (32) was attached to, so it is now out of order with the rest. When we compile again, it will be note 28.

— Hovering your mouse over a note will move the number over to make room for the × button.

— Although inline footnotes will not be numbered or displayed in the footnote list, they *will* be numbered. Here we can see a jump between number 29 and 31 where note 30 is an inline footnote in the editor.

— Although not depicted here, it is important to note that footnote numbering is a feature *of that note*. If you drag and drop or copy and paste text between projects, any numbered footnotes within the copied material will be translated over to the other project.

  This also means that snapshots (section 15.8) of the document will preserve footnote numbering as it was at the time of the snapshot.

[3] Use of this command will set your preference for all future projects, but will not change any existing projects you've created.

The View ▸ Text Editing ▸ Prompt Before Updating Footnote Numbers command will be useful in keeping numbers static in between compiles. This can be of considerable use if you have copies of your work out to proofers and wish to not have the footnote numbers they are using be out of sync with your internal numbers. You can also hide numbering to achieve this effect, without the prompts, as described above.

As footnote numbers are saved into the footnote itself rather than being processed dynamically, snapshots of documents taken with stored footnote numbering will use the same numbering that

18.3.7 Finding Linked Notations

Scrivener provides tools for stepping through your entire project, searching for inspector comments and footnotes, either in general or to find only those with specified text in them. Read more about this feature in Find by Format and Text (subsection 11.6.2).

It is also possible to use the standard text search tool (section 11.2). With the inspector open, searches will be highlighted in the sidebar for you, and the replace function will also search and replace matched text in any applicable inspector notes.

Return to chapter ⤴

18.4 General Usage Tips for Notation

These tips apply to both inline and linked notes, whether they be for your own internal commentary or for the reader, in the form of footnotes or endnotes.

18.4.1 Stripping Out All Notation

To clean out all notation from the text you are working with:

1. Select the text you wish to clean out notes from.
2. Use the Edit ▸ Copy Special ▸ Copy without Comments and Footnotes menu command (⇧⌥⌘C).
3. This will leave the original text selected in the editor, so if you wished to replace it you could paste immediately following the above command. The cleaned text can of course be pasted anywhere you like.

18.4.2 Resetting Linked Note Formatting

Both comments and footnotes allow formatted text within them. This means you can apply whatever formatting you wish into these, and they will be exported with this formatting when you compile (assuming your compile settings

have not overridden footnote font and size). Consequently, adjusting the Editing: Formatting preferences tab (subsection B.3.2) for notes will not immediately impact them, as this might wipe out your formatting. You can however, just as with documents, reset the formatting to the application default by right-clicking on selected notes and choosing "Convert to Default Formatting".

18.4.3 Exporting Annotations and Comments

You can export all of the inline annotations and linked comments in your project into an RTF file with File ▸ Export ▸ Comments & Annotations. Within the file output selection dialogue you will be given an option to restrict this to the selected items in the sidebar, rather than the entire project. You can also choose to export titles along with the comments, making it easier to find where they came from.

18.4.4 Notation with Copy and Paste

Within Scrivener, you can copy or cut, and paste ranges of text that contain any style of notation, and it will be carried along with the selection. This can even be done in between projects.

However when pasting into other programs the footnotes and comments will may be converted to a format suitable for where it is being pasted:

— For most word processors you will get real footnotes that show up at the bottom of the page with numbers linked to them within the text.

— Pasting into many other applications, especially plain-text, will produce a different result if they do not support either footnotes or comments. In this case they will use a "flattened" version that looks like footnotes (or endnotes to be more precise) and inline commentary donated by square brackets. Since these applications cannot use real footnotes and comments in the first place, this is an acceptable compromise.

If you wish to copy and paste text *without* any notation at all, use Edit/Copy Special/Copy without Comments and Footnotes, or ⇧⌥⌘C.

Pasting Into Word 2011

If you use Word 2011 or greater, and when you paste footnotes and comments find you get the flattened form, you'll want to enable the **Use Word-2011 compatible copy** setting in the Sharing: Conversion preference tab (subsection B.7.4). Do note that when this option is enabled, it will cause native macOS software which would ordinarily use flattened formatted text (like Mail) to omit all formatting entirely, as that flattened version will no longer be provided.

18.4.5 Converting Notation Between Types

Both inline and linked notation can be freely converted between footnote and comment/annotation forms. This can be used to "soft delete" notes you may not need the reader to see, but haven't decided on yet—or to make a note you originally meant for yourself something that annotates the finished copy.

To convert inline notation:

1. Select the full range of inline notation you wish to convert in the editor. This can be most easily done by right-clicking on the note and using "Select Annotation|Footnote|" from the contextual menu, or by using the Edit ▸ Select ▸ Select Annotation|Footnote menu command.
2. Use the command you would use to create an inline footnote or annotation to convert the type. For example if you select an inline footnote, then ⇧⌘A will turn it into an annotation, using current preferred annotation colour.

Converting linked notation will retain its metadata in the background. Comments converted to footnotes will not lose their colour assigned (though of course it will not be used until converted back into a comment), and likewise footnotes will remember their compiled reference number:

1. In either the Comments & Footnotes inspector tab (where you can select many notes at once) or the individual note's popup window, right-click on the header area for that note.
2. Select "Convert to Comment|Footnote", as desired.

It is also possible to convert between inline and linked form. The Edit ▸ Transformations ▸ submenu contains four commands for converting between inline footnote, annotation, linked footnote and comment format. These commands either operate on the entire editing session in the editor at once, or within the span of a selection of text if applicable.

18.4.6 Document links in notation

Owing to technical limitations in how inspector notes work, it is not possible to create or store such links within them. Consequently, if you convert your inline notes to linked notes, you will lose any internal links that had been set within them. If you want to use links inside notes, inline notation supports the ability to create document links within them. This is a handy way to "hide" link text from the final manuscript, keeping these links purely for your own benefit.

Return to chapter ⤴

18.5 Text Colour and Highlights

Arbitrary text colour can be applied to your document in a fashion similar to any other formatting range. Select the text you wish to change the colour on, and either use Format ▸ Color... (⇧⌘C), or use the text colour selector in the Format Bar (subsection 15.5.2) to set the colour. The format bar will remember the last colour you have used, so it easy to rapidly apply the same colour to multiple text selections with a single click. It will also pick up a colour when you click on text that has already been coloured.

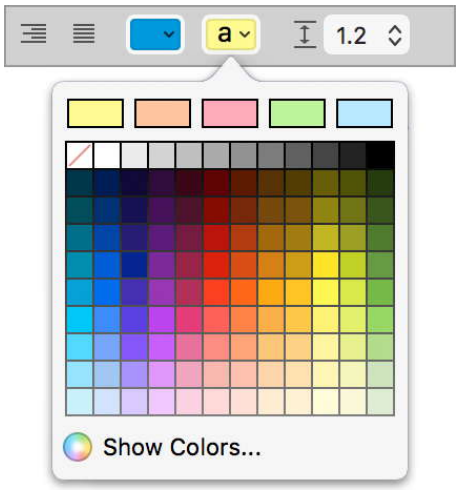


Figure 18.8: The colour selection pop-over, showing options for the text highlighter button.

If you wish to change the colour, right-click on the colour chip or click and hold. A pop-over will appear (Figure 18.8) with the following contents:

— The built-in presets (black, red, etc.)

— Common swatches (the first swatch with a red line drawn through is "remove colour" choosing this one will erase any colour in the selection and return the text to default)

— Show Colors (access to the colour palette)

Setting a highlight to a range of text is quite similar, only you will use the second colour chip on the format bar, instead. As with the text colour control, single-clicking once will re-use the last highlighter colour, and right-clicking (or holding the button down) will access the color pop-over.

Text highlights can also be applied with the Format ▸ Highlight//submenu. The initial entry in this menu will list the last highlight colour, giving it a shortcut, ⇧⌘H, for rapid application at any time. The contents of the highlight menu will be divided into four main sections:

— Remove Color

— Built-in presets (yellow, orange, etc.)

— Show Colors (access to the colour palette)

Colours and Compiling

By default, text colours and highlights will be compiled into your final manuscript. If you use these features for internal editing, you will find options for disabling them in compile settings, under the gear tab (subsection 23.4.3). Not all word processors support text colour and highlighting features.

Scrivener provides tools for stepping through your entire project, searching for text colour and highlighters. Read more about the Find by Formatting Tool (section 11.6).

18.5.1 Naming Text Highlights

The default text highlight marker names can be changed to something less generic, on a universal level, by using colour palette swatch groups. Swatch groups are macOS' way of letting you create selections of colours and giving them useful names. To change the highlighter colours, show the Color Palette (⇧⌘C), and select the third icon from the left.

Upgrading from Scrivener 2

If you have used this feature in the past to give your highlights custom names in Scrivener 2, you will find that your associations no longer work because Scrivener 3 has refreshed the colour palette used for highlights. Since the capability requires precise colour matches, your references to the old Scrivener 2 colours will need updating. You will want to remove or rename the old "Scrivener" swatch set, using the ⚙ button to the right of the set selection dropdown, and then restart Scrivener to have it create a fresh new set to work from.

Next, from the "Palettes" dropdown menu, select the group titled, "Scrivener". You should now see something resembling the above screenshot.

Double-click on any of the provided highlights to rename it. Your changes should be visible immediately in the Scrivener interface, where applicable.

Your own colours can be added to any of Scrivener's various colour selection menus, as well. If necessary, click and drag downward in the location marked above, "Custom colour shelf". You should see a row of white squares appear. You can drag out as many rows as you like.

The easiest way to create new custom colours is to use either the colour wheel (first icon) or the sliders (second icon). When you find a hue that you'd like to

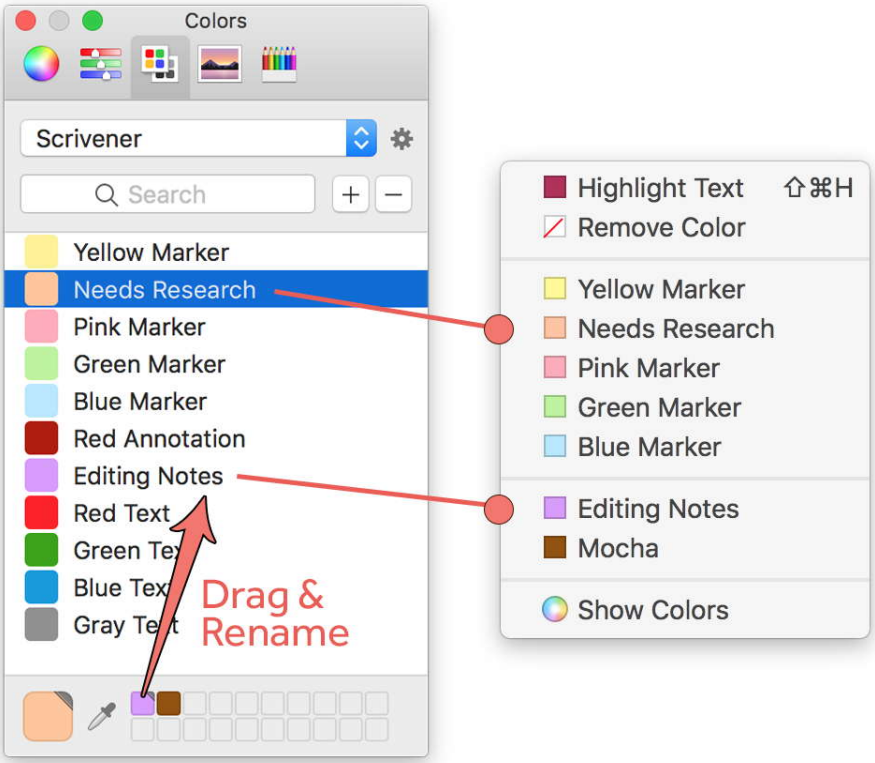


Figure 18.9: Scrivener's custom swatch set

use as a highlight, text colour, or annotation colour, drag from the large colour preview area at the top of the pane, and drop into one of the empty slots in the custom shelf.

If you check one of Scrivener's colour menus right now, you'll see that it has already been added below the base sets. To change the name of your custom colour, click on the swatches icon again, make sure the Scrivener set is active, and drag the colour from the custom shelf into the list. Now that it is there, you can double-click to rename it.

Note that due to the way colours are estimated, if you have several subtle variations they will all get the same name. To avoid this, create a new named colour swatch for each variation. This is a purely aesthetic feature though. Names are not used for anything, except for your own reference.

macOS Tip

If you wish to share your colour sets with other users, or transfer them to another computer, you will find a file in your user Library folder, under Colors, named `Scrivener.clr`. Place this file into the same folder on the second computer and it should become instantly available.

Return to chapter ⤴

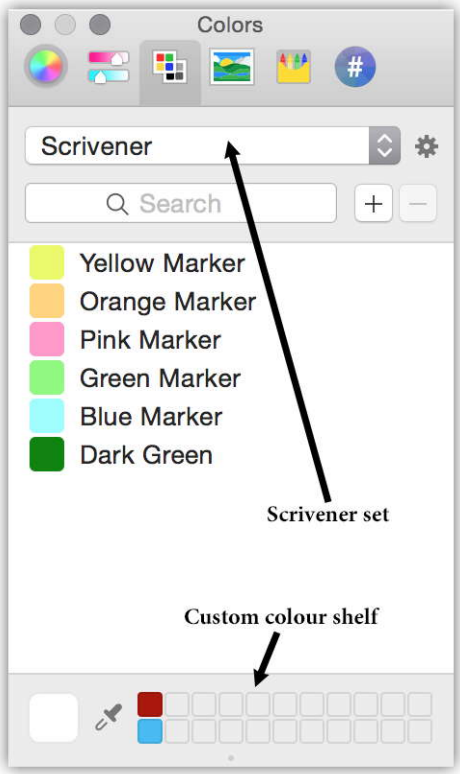


Figure 18.10: Creating custom swatches in the Scrivener set.

18.6 Marking Revisions

Scrivener uses a basic technique for making revisions visibly distinctive from one another and the base text. At its core, the feature is simply a formalised method for apply text colour as you type. The formal nature of this is in both the selection of the tool, rather than picking some arbitrary colour from the Format Bar, and a collection of tools for working with that particular colour at a later time.

Whenever a revision level has been enabled in a project, it will impact all primary editor views where text editing is possible (it will not impact notes or other ancillary fields). The cursor colour will be changed to that colour, to remind you of this fact, before you start typing. In addition to colour the text as you type, usage of the strikethrough format will colour the strike-out in accordance with the revision level, too.

Limitations

Revision levels are not to be considered equivalent to "track changes", as seen in version controlled software, or Microsoft Word. They are very simple, low bandwidth tools, that only mark modified or new text. Removed text will not be marked in any way. If you wish to track all changes between a document and its revised state, take a look at the snapshot feature (section 15.8). Stricken text, while it will display the revision level from when strikeout was used, will need to be manually removed later on.

18.6.1 Setting a Revision Level

There are five revision levels available, and each are accessible in the Format ▸ Revision Mode ▸ submenu. Simply click on the coloured menu command you wish to use as your revision mode, and begin editing the document. As you type in new text, it will be automatically coloured with the revision level, as if you had typed it in and then selected it, and applied a text colour with the Format Bar. Paste text into the editor using the revision colour with the Edit/Paste and Match Style command (⇧⌥⌘V).

Selecting the same colour from this menu again will toggle the revision marking system off. This can be especially useful if you bind these colours to keyboard shortcuts (section A.1). If you bind ^2 to level two, then you could tap that shortcut once to insert a revision, and again to return to standard editing.

18.6.2 Marking Existing Text

There are times when you may wish to mark existing text with a revision level. To do so you must already have a revision level set, then select the text you wish to mark, and use the Format ▸ Revision Mode ▸ Mark Revised menu command. This menu command will only appear when active text has been selected, so if you do not see it, ensure that the text you have selected is in the active editor.

18.6.3 Removing a Revision Level

Once you have reviewed a document's revisions, you will probably wish to remove the revision markings from that document. This can be very easily done by selecting the appropriate revision level from the menu, and then without selecting any text, use the Format ▸ Revision Mode ▸ Remove Current Revision Color menu command. This command will work on the entire text session that is in the active editor, so if you are editing multiple documents in Scrivenings, ensure you wish to wipe out markings for all visible documents. If you only wish to strip out marks from a portion of the session, use ⌘4 to isolate the document, first.

You *can* also use this command in conjunction with a selection, to remove only a portion of the revision markings from the document. Make sure that the

selection only contains text marked with that revision, and use the same menu command above.

Since revision levels are nothing more than coloured text, if you happen to have other text in your document that is coloured identically, this command will strip out those colours too. In general, it is a good idea to reserve these colours for revisions only, if you intend to use the feature heavily.

18.6.4 Removing all Revision Markings

When you've reviewed all revisions are are ready to return the document to a default colour state, you can use Format ▸ Revision Mode ▸ Remove All Revisions to strip out all revision level markings in the active text session. The same warning from above applies: this will impact *all* documents in a Scrivenings session. Likewise, the same warning about overlapping colour usage applies as well: any colours matching any of the revision levels will be stripped from the documents.

18.6.5 Changing the Revision Colours

It is possible to change Scrivener's associations for the colours to use for the various revision levels, in the Editing: Revision Colors preference pane (subsection B.3.3).

18.6.6 Finding Revision Markings

Scrivener provides tools for stepping through your entire project, searching for revision levels. Read more about this feature in Find by Format and Text (subsection 11.6.4).

Return to chapter ⤴

Scriptwriting

19

In This Section...

Although Scrivener is not intended to be a dedicated scriptwriting program (for such a program you might want to try Final Draft, Fade In Pro, or WriterDuet, if you have not done so already), it does allow for basic script formatting and export to formats that most scriptwriting software can read, making Scrivener suitable for first drafts.

Default Scriptwriting Font

By default Scrivener will use the Courier Prime[a] font for all scriptwriting modes that would make use of Courier as a font. Courier Prime is specifically designed for scriptwriting, using the same font metrics as Final Draft Courier and standard Courier, meaning you can rely upon using it for accurate page counts. If you do not like Courier Prime, or would like to use Final Draft Courier if you have it installed with Final Draft, then you can create your own scriptwriting preset (section 19.7).

[a] http://quoteunquoteapps.com/courierprime/

19.1 Formatting a Script in Scrivener

To format a script in Scrivener, select the format you want to use from the Format ▸ Scriptwriting ▸ submenu, or create a new project using one of the pre-built project templates in the "Scriptwriting" category. When in scriptwriting mode, the top item in the Scriptwriting submenu will be checked and will display the name of the format you are using. This preferred format can be toggled on individual documents with ⌘8.

By default, the standard "Screenplay" format is selected and Scrivener is in scriptwriting mode. If "Script Mode - Screenplay" did not have the tick next to it, then we would know that the current editor was not in scriptwriting mode (that is, it would be in normal prose mode for general writing). You can more easily tell whether or not you are in scriptwriting mode by looking at the footer bar, which will display various scriptwriting tools and hints, rather than the standard word and character statistics display. Additionally, the binder icon for that item will be tinted yellow, with three-hole punches on the side.

Scriptwriting mode is a setting which is individual to each document. Once a document has been toggled to scriptwriting mode, it will remain that way until you change it. Thus it is possible to have a standard document in one split for your notes, while using scriptwriting mode in the second split to draft your work. However, it is not possible to use more than one *type* of script formatting in the same project. This is a project level setting, and so if you need to create a stage play and screenplay at the same time, you will need to do the adaption in a tandem project.

At the bottom of the scriptwriting menu you can see a list of all the different script formats that are available. Scrivener comes with the following formats built-in:

- Screenplay
- BBC Radio Scene Style
- BBC Taped Drama
- Comic Book (Antony Johnston)
- Comic Book (Alternative)
- Interview (a simple question and answer format)
- Stage Play (UK)
- Stage Play (US)
- Transcript (contains useful formatting for the transcription of audo/visual material, including automated timestamp insertion, when used in conjunction with Scrivener's media player, in the opposing split)

You can, of course, mix up different text modes in the same document or draft, so that parts of a document may be written as a script and other parts written

as general text. This makes it very easy to write treatments in Scrivener, or to intersperse general notes with script.

The screenplay format that comes with Scrivener by default is based on a Hollywood standard, but with half an inch added to the left indent of each element so that when you print with a standard set-up of one inch margins, the left margin will actually start at one-and-a-half inches, which is the standard to allow for three-hole binding. In other words the end result will be accurate, even if the numbering by the editor's ruler is slightly off.

The scriptwriting mode in Scrivener works much like other scriptwriting programs (primarily, Final Draft). On the right end of the footer bar is a dropdown menu containing the various elements for the selected format. Clicking on an element will reformat the current paragraph to the format of the selected element. So, for instance, if you clicked on "Character", the current paragraph might all be capitalised and centred.

This menu also has full keyboard access, allowing you to change modes swiftly and without the use of the mouse. The menu can be called up with ⇧⌘Y at any time from within a scriptwriting document (even in composition mode). To select an individual element style, tap the associated letter for it, on the right side of the menu. Example: "T" for Transition, and "C" for Character. Additionally, each element (up to the ninth) will be assigned with a number (in the order that they appear in the element menu) that is combined with the Option and Command keys as a direct shortcut. For example, the Parenthetical format is the fourth available element, and so it can be invoked with ⌥⌘4; a Transition is the 6th element, and so can be invoked with ⌥⌘6 and so on. Learning these numbers can be useful as you do not need to call up the intermediary menu to change element modes.

Upon hitting enter and typing, the paragraph formatting will be changed to that of the next set element. So, for instance, if you hit Return after having typed in a Scene Heading, the typing attributes might be set to Action. At this point, you could easily change the current element by pressing the Tab key—hitting it once would change the current element to Character. Hitting it several times will cycle through different elements.

The footer bar tells you which element will be created next by hitting Tab or Return. For example, in the Screenplay format, when waiting to type in a character name, pressing Tab will switch you to Transition and pressing Enter will switch to Dialogue. However, once you begin typing in a character name, the tab hint will change to Parenthetical, letting you drop to that element instead of dialogue if you wish.

If for some reason you need to change the element type for text you've already entered, you can place the caret anywhere within the text you want to change, and invoke the element menu in the footer bar, or use the Format ▸ Scriptwriting ▸ Change Element To ▸ submenu to select a different element. Note that when moving from an element that uses all-caps to an element that does not, you will need to adjust the capitalisation manually, as Scrivener will not try to guess what is appropriate. The Edit ▸ Text Tidying ▸ submenu has a number

of conversion routines to aid in this.

19.1.1 Auto-Completion

Using the built-in script settings, auto-completion is available for some elements, as is appropriate for their context. Upon choosing Scene Heading, for instance, you can start typing with "E" and you will be presented with a options like "EXT." and "EVENING". You may disable or enable (or add your own) additional items from this list by editing your script settings (section 19.7.2), though note you should in general not use this list to assign project specific completions like scene locations. It is best to use the Auto-Complete tab in Project Settings for this (section C.6). You may want to save your scripting adjustments for future use in another project. The auto-completion lists in the script settings should be seen as integral parts of the scriptwriting process in general.

Some elements have been configured to automatically do this for you. Scene headings are a good example. They will add anything you type in between a '.' and a '-' in a scene heading line. In practice this means the part marked in bold in this example would be added to the project auto-complete list: "EXT. **THE RED LION** - NIGHT". When words and phrases are added to the project auto-complete list in this fashion, they will be assigned a scope (subsection C.6.1) which restricts their eligibility to only when you are typing within that element. A character name will not suddenly pop up in the slugline, for instance. You can adjust the scope of automatically added phrases using the main project auto-complete list.

19.1.2 Dual Dialogue

If you intend to use the Final Draft FDX compile format for exporting, you can designate text as being dual dialogue by surrounding both bits of dialogue (including the character names) in a Format ▸ Preserve Formatting block. This will not negatively impact the text if you decided to use PDF or another format for proofing, prior to moving to Final Draft. Under normal conditions, it merely informs the compiler that the text in the blue box should not have its formatting altered. Since scripts are already pre-formatted, the compiler will not typically be touching the format anyway, and thus the designation does nothing.

Return to chapter ⤴

19.2 Using Page View to Estimate Page Counts

While the point should remained stressed that Scrivener is not designed at any level to provide a completely accurate pagination solution, with rigid formatting, such as that used by most scriptwriting formats, and a few optional settings, it

is possible to get very close to an accurate page count, when using Page View (chapter 16) mode.

The first thing you will need to do is switch Scrivener's default Scrivenings separator from a divider line to "Minimal", which can be set in the Appearance: Scrivenings preference pane (subsection B.5.13) via the **Scrivenings Separator** setting. This version uses a "zero-height" display model which will not introduce page inflation over long documents, like the standard divider will. You will also want to disable the usage of Scrivenings titles for your project, with View ▸ Text Editing ▸ Show Titles in Scrivenings, as it will likewise expand the overall height of the document over large sections.

Scrivenings View and Scriptwriting

When creating a scrivenings session, if the session includes a mix of script formatting documents and regular documents, Scrivener will determine which mode to use based on the type which has the most entries in the session. This means, if you create a scrivenings session with 4 standard documents and 2 script documents, script mode will not be on by default. In the other direction, if there are more scripts that standard documents, script writing tools will be enabled in those standard documents. You can switch modes while in scrivenings with the ⌘8 shortcut. This will only impact (globally) the scrivenings session, and not any of the underlying files.

Finally, you will need to ensure that the paper size is set up correctly in the File ▸ Page Setup... dialogue.

If you work with scripting formats quite a lot, you might wish to ensure that **Show page view in new projects** is enabled in the Appearance: Page View preference pane as well.

Return to chapter ⤴

19.3 Importing a Script from Final Draft and Other Programs

Final Draft Users

Refer to the section on exchanging files with Final Draft, version 8 and greater (section 19.5). For users of Final Draft 7 and earlier, you can import scripts created in Final Draft by using "Save As" in Final Draft to save your script in "File Converter" (FCF) format, then import the resulting .fcf file. Scrivener will try to match the elements in the FCF script to the elements in the current script format. For projects that use the basic screenplay formats, this should be all you need to do to import a script with all of its elements recognised.

Because Scrivener is not a dedicated scriptwriting program, the way it handles script elements is different from Final Draft or Movie Magic Screenwriter. Dedicated scriptwriting programs generally assign hidden metadata to each text element that tell the program that, for instance, this line of text is a scene heading, that paragraph is a piece of dialogue and so on. Scrivener does little more than automate the process you would have to use yourself were you to use a standard word processor to format your script; that is, it just sets up the necessary paragraph formatting for each element and auto-capitalises if necessary. Scrivener recognises elements by their paragraph formatting. For instance, if a paragraph has a three-inch left indent and single line spacing, Scrivener will look up this formatting in the list of script elements, and if an element is found with matching formatting its name will be selected in the elements pop-up button in the footer view. If no elements match, "General Text" will be selected.

This means that if you import a script into Scrivener from another program and want to continue working on it, the script format mode selected in Scrivener must exactly match the formatting of the script you have imported.

19.3.1 Importing Other Scripts as RTF

If your script formatting was not recognised properly, or if you imported a script that has no matching format mode in Scrivener, you will need to create your own format mode that matches the script using the Format ▸ Scriptwriting ▸ Script Settings... panel.

This method can also be used to recover Scrivener formatted script files, if the original scripting settings have been lost for one reason or another.

Here's how:

1. Export the script from your scriptwriting program as an RTF file.
2. Import the script either by dragging the RTF file from the Finder into Scrivener's binder or by using File ▸ Import ▸ Files...
3. Ensuring that the script is visible in the editor, click into a line of text that represents one of the elements you want to be recognised (for instance, by clicking into a line of text that should be a scene heading).

4. Open the Script Settings panel with Format ▸ Scriptwriting ▸ Script Settings....
5. Enter a title for your new format in the "Format Title" text field.
6. Select the first element in the list and give it the name you require (i.e. the name of the element in which you placed the cursor in step 3) by double-clicking into it and editing it if necessary.
7. From the Manage... dropdown menu in the Script Settings panel, select "Use current font & paragraph settings". This will copy the font and paragraph information from the line of text in which the cursor has been placed in the editor into the Font and Paragraph panes of the Script Settings sheet.

Repeat this process, matching the settings for the elements in the Script Settings panel with the text in the editor for each element in your script.

Be sure to save your script format for use with other projects using the Manage... pop-up button at the bottom of the Script Settings panel.

You can then use your new format with all projects in the future for any scripts you import. Once you have successfully created your own script format mode, all of the elements in your imported script should be recognised in the pop-up button in the footer view of the editor, and you should be able to use the script mode to continue editing your script. Note that more complicated script formats may require a little more tweaking to be recognised, but the above process should work for most.

If you do create a script format mode that recognises elements from an imported script, please use "Save to file..." in the Manage... pop-up menu from the Script Settings panel to save your format as an XML file somewhere safe on your hard drive so that you can back it up and load it on other machines or following a hard drive reformat if necessary. Also, please feel free to upload such format files on the forums, where other Scrivener users may find them useful—or from where you can always download them again should you lose them.

19.3.2 Importing Plain Text Formatted Screenplays

A number of screenplay-oriented programs (such as Movie Magic Screenwriter) support what is known as a plain-text formatted screenplay. Since the roots of this format come from the days of the typewriter, it is in fact a simple matter to use nothing more than a raw text file to print a screenplay.

Looking for Fountain Support?

Plain text screenplays are not Fountain files. They in fact look just like the screenplays you'll be printing from software like Scrivener or Final Draft. Refer to Working with Fountain (section 19.6) for tips on importing files in this unique form of markup.

Scrivener is capable of both creating (subsection D.5.5) and importing these simple files:

1. Select the folder into which you'd like to import the screenplay, in the binder.
2. Use the File ▸ Import ▸ Plain Text Formatted Screenplay... menu command to bring up the file selection dialogue.
3. Navigate to and select your .txt file on the disk and click the Open button.

The screenplay will be imported into the project and converted to Scrivener's internal format.

Depending on where your screenplay is currently located, you may find it simpler to paste plain text screenplays into Scrivener, with the Edit ▸ Paste Plain Text as Screenplay menu command. This must be done into a document that is using script mode (⌘8).

Return to chapter ⤴

19.4 Printing or Exporting a Script

You can use the compiler to print your script (or to turn it into a PDF file). For examples of how to do this, take a look at one of the scriptwriting project templates by going to File ▸ New Project... and selecting a project template such as the Screenplay template. Read the instructions that come with the template to see how to set up your project so that it is formatted properly when printed (or exported).

In the majority of cases, just as you would with many other types of writing in Scrivener, you will want to export your script from Scrivener so that you can do all the final formatting in a dedicated formatting program such as Final Draft, see Exporting Scripts (subsection 23.5.2) for information on how to use the Compiler to export to many popular scriptwriting programs.

Return to chapter ⤴

19.5 Working with Final Draft

That this information is only relevant to Final Draft 8 users. Users of previous versions of Final Draft should use the FCF format for importing and exporting their work.

19.5.1 Importing Formatting from a Final Draft Document

Optional Step

For basic scripts or those that will use Final Draft's standard screenplay formatting—indeed, for most screenplays—this step can be omitted, as it is mainly concerned with setting up the formatting for custom scripts. Proceed to the next section if this is the case.

Whenever you import a script from Final Draft into Scrivener, all of its elements will use the formatting specified in Scrivener's script settings. These can be set up by selecting "Script Settings..." from the Scriptwriting submenu of the Text menu. The Script Settings panel should be familiar to anyone who uses Final Draft, as it is set up to be very similar, allowing users to create custom script formats.

Because a Scrivener project comprises multiple documents, it is important that the script settings are set up as the user desires *before* importing or creating any script documents in the project.

If you are writing a screenplay using standard formatting, you don't really need to do anything here. However, if you are using a custom format, or if you have an FDX script that uses a custom format which you wish to retain after import into Scrivener, you need to set up Scrivener to use that custom format first.

Fortunately, this is very easy to do, as Scrivener can read the formatting directly from an FDX file, but you must do this before importing the script itself. (Remember, Scrivener can hold many different script documents but can only use one script format in a project at a time; this is why setting up the format must be done separately from importing a script.)

To import the formatting from an FDX file:

1. Open the Script Settings panel by going to Format ▸ Scriptwriting ▸ Script Settings...
2. In the Script Settings panel, click on the "Manage..." pop-up button in the bottom-left.
3. Select "Load from Final Draft .fdx or .fdxt..."
4. In the open panel that appears, select the FDX file that contains the elements and formatting you want to use.

This doesn't import the actual script, it just imports its elements and formatting for use in the current project.

To test the new format, create a new document, choose the new scriptwriting format and start typing (changing elements using tab and enter or using the pop-up menu in the footer view beneath the editor). You will see that the script uses the formatting of the FDX file. Now that this is set up, you are ready to import the contents of the FDX file.

To save the format for future use in other projects, use the same Manage... dropdown menu to select "Save for use with other projects". The imported script format will now appear in the main scriptwriting menu for all projects.

19.5.2 Importing FDX Files

You can import FDX files into Scrivener in one of two ways. Both are fully featured, the only difference between the two is that the second will offer options for cutting up the script into smaller pieces automatically:

1. Using the standard file import methods; drag and drop into the project binder, and File ▸ Import ▸ Files....
2. Go to File ▸ Import ▸ Import and Split... and select the FDX file that you want to import. See Import and Split (subsection 9.1.6) for further details.

Many features will be retained: script notes become Scrivener footnotes, revisions are marked up in red, highlighting is retained and so on.

If you wish the FDX file to be part of the script that will eventually be exported, be sure to import it into the "Draft" folder, or drag it there after importing.

19.5.3 Exporting Individual Documents to FDX

To export individual scripting documents to Final Draft, just select the documents you want to export in the binder and then go to File ▸ Export ▸ Files... Select "Final Draft (.fdx)" as the export format.

The exported file should open in Final Draft with all features intact.

More commonly you will want to combine all of the scripting documents in the Draft folder of your project into one FDX file:

1. Use the File/Compile... menu command.
2. Select **Compile for:** "Final Draft (.fdx)" at the top of the window.
3. Select "Script or Screenplay" in the sidebar.
4. Assign section layouts if necessary (subsection 23.3.2), and check through the various option panes on the right-hand side.
5. Click on Compile... and choose a filename.

You should find that most elements are retained—footnotes become script notes, coloured text becomes revised text, and so on. The header shows whatever is set in Scrivener's Compile Draft panel.

That covers the basics of importing from and exporting to the FDX format using Scrivener.

Return to chapter ⤴

19.6 Working with Fountain

Fountain[1] is a simple markup language (based loosely upon Markdown[2]) designed just for screenwriters who need or prefer to work in a plain-text environment (which will be of particular interest to scriptwriters who prefer to write on the go with a mobile device). Fountain helps with composing and editing without losing the individual meanings behind the elements (character name, action, transition, etc.). From the Fountain website:

> Fountain is a simple markup syntax for writing, editing and sharing screenplays in plain, human-readable text. Fountain allows you to work on your screenplay anywhere, on any computer or tablet, using any software that edits text files.
>
> Taking its cues from John Gruber's Markdown, Fountain files are eminently readable. When special syntax is required, it is straightforward and intuitive.
>
> Even when viewed as plain text, your screenplay feels like a screenplay.
>
> Fountain supports everything a screenwriter is likely to need in the early, creative phases of writing.

When importing or syncing Fountain files, they will be processed and converted to Scrivener's internal scriptwriting system for you. There is no need to use the markup conventions while writing within Scrivener. Certain Fountain conventions will be converted to rich text or Scrivener metadata, in addition to the scriptwriting elements themselves:

— *Boneyard markers*: sections of text marked between "/* boneyard markers */" will be imported into Scrivener as struck-through text and vice versa.

— *Dual dialogue*: when using the syntax for dual dialogue, Scrivener's convention of placing the characters and dialogue in question in a Preserve Formatting block. Likewise adjacent dialogue marked with Preserve Formatting will be marked in exported Fountain files.

— *Script notes*: when importing, text marked as a script note that appears entirely on its own line(s) will be placed as an inline annotation. Otherwise, when script notes are added to text that is otherwise an element, they will be anchored to the nearest word and linked as comments in the Inspector. If you prefer one form of annotation over another, you can use the Edit ▸ Transformations ▸ submenu to switch between styles. Scrivener uses inline annotations in cases where script notes are on their own paragraph

[1] http://www.fountain.io/

[2] http://daringfireball.net/projects/markdown/

to avoid potential problems with removing the text and causing elements around that text to lose their syntax. For export, all annotations and inspector comments will be converted to script notes, if you opted to include them in your settings.

Fountain is for Screenwriters

One important detail to remember is that Fountain has been designed for screenplays. Using it with other scriptwriting formats is not supported as it may work unreliably or not at all. If you do wish to try using Fountain with another scriptwriting format, all you need to do is make sure that the elements (section 19.7) are named precisely as they would be in the standard screenplay format.

Scrivener can import the Fountain format using two different methods:

— *As a single script*: merely drop the Fountain file into the binder. Scrivener looks for the '.fountain' extension, so make sure the file has had its extension changed accordingly, otherwise the file will be handled as plain-text. When properly handled, all elements should be imported and formatted automatically to the screenplay script format.

— *Split up into scenes*: as with Final Draft files, you can use File ▸ Import ▸ Import and Split... to select the file. Again, it will need a '.fountain' extension to be properly recognised and imported. This will result in a new binder item being created for each discovered slug-line. If a section header has been typed in immediately prior to the slug-line, it will be used to populate the binder item title[3]. Fountain synopses texts will be gathered from throughout the scene and combined into Scrivener's own synopsis index card. Read more about Import and Split (subsection 9.1.6).

For integrating with a mobile device, Fountain is a selectable format in the folder synchronisation feature (section 14.3).

Finally, it is possible to compile to the Fountain format (subsection 23.5.2).

Return to chapter ⤴

19.7 Creating Your Own Script Formats

Scrivener comes with a number of script formats built in. You can also create your own script formats tailored to your own requirements; as well as export and import script modes for sharing on the Internet.

[3] Note that while Fountain has support for multiple levels of depth with its header syntax, Scrivener's scene import will produce a flat list.

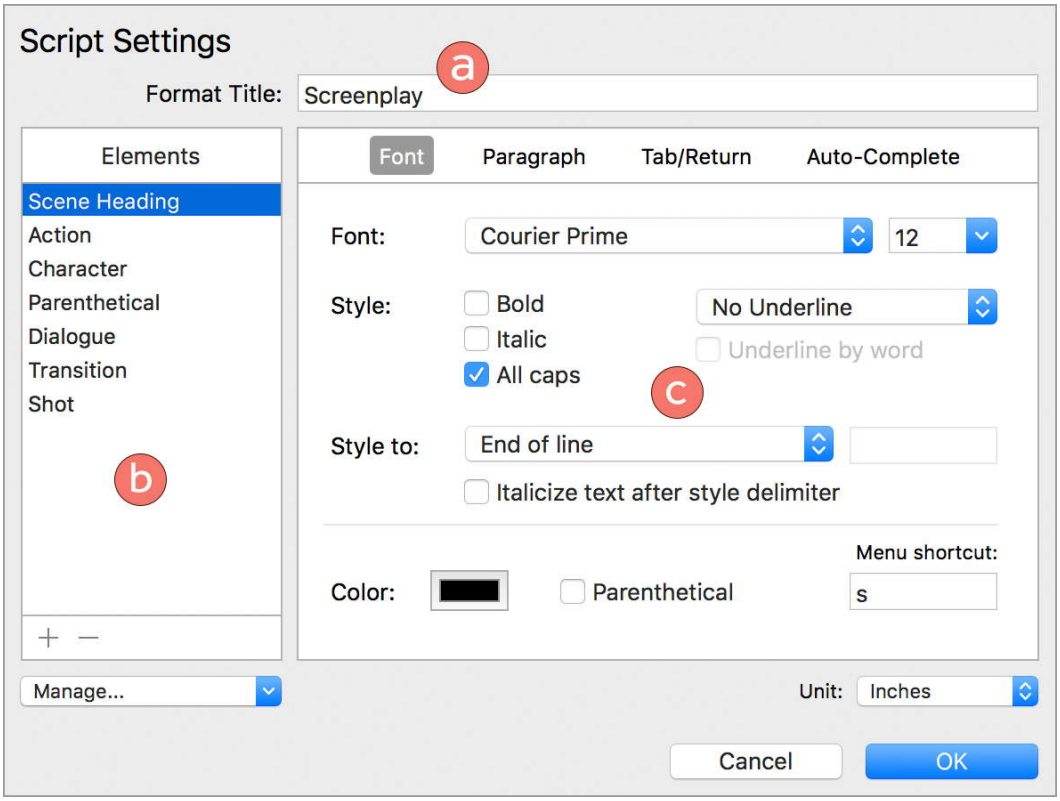


Figure 19.1: All aspects of script formatting and behaviour are established in the Script Settings window.

To create your own script format mode, use the Format/Scriptwriting/Script Settings... menu command to bring up the script editing interface. At the top of the panel is the **Format Title** text field (Figure 19.1), marked (a). Enter the name of your format, as it will be displayed in the main menu, should you save these settings for future use beyond the current project.

On the left side of the panel is the Elements list, marked (b). Here you define the parts that make up your script, such as "Scene Heading", "Dialogue" and so on.

— You can add new elements and delete existing ones using the + and − buttons beneath the list, and you can rename elements by double-clicking them.

— Reorder elements by dragging and dropping them; this will change their numerically based keyboard shortcut assignment.

— The "General Text" and "General Text (Centered)" elements are displayed while working on the script in Scrivener, but will not appear here as they have no special properties by definition—they merely represent text not assigned to an element in this list.

The large tabbed area marked (c) is where the settings for the current element,

selected in the sidebar, can be viewed and modified. We will cover each tab of settings in this area of the manual.

If you need to use a system of measurement other than inches, use the dropdown menu in the lower right-hand corner to change the **Units**.

Clicking OK saves the script format into the *current project package only*. This means that if you share your .scriv project with someone else, they will be able to use these settings without having to install anything. No further actions are necessary unless you wish to export your script to other projects or machines.

19.7.1 Managing Scripts

In the bottom left-hand corner of the script settings panel is a dropdown menu entitled Manage.... The menu provides several management functions not found elsewhere:

Reset to defaults Reverts all customisations that you have made to the Script Settings panel to the Screenplay default. If you wish to revert to a saved script setting, simply select that script from the Format ▸ Scriptwriting ▸ submenu, instead of using this tool.

Use current font & paragraph settings Attempts to import all available character and paragraph level formatting attributes from the editor into the *currently selected* element in the above element table. This can be quite useful when you have imported a script from another program and wish to create a script format from the existing text. Simple click through the document, locating element types, and use this tool to import the correct formatting into each element type.

If you have a file in Final Draft 8 format or above, you will want to use the "Load from Final Draft" tool, mentioned below, for scanning .fdx files for types, rather than doing all of this by hand.

Load from file... Loads a Scrivener script format file from the disk. Use this if you have downloaded a script from the Internet, or are in the process of transferring formats from one computer to another.

This will not install the saved script into your Scrivener support folder, it will merely import the settings into this one project. Use the "Save for use with other projects" command, following this one, to install it system-wide.

Load from Final Draft .fdx or .fdxt file... Scrivener can examine an existing Final Draft or Final Draft Template file for formatting rules and names, and attempt to convert them to Scrivener's internal script formatting.

Save to file... Saves your current settings to an external file that you can easily backup, upload to the Internet for sharing, or send to another of your computers.

Save for use with other projects Installs the project's current script settings to your system for usage in all of your projects. If you wish to share the file with others, or transfer them to another computer, use the "Save to file..." option to more easily create an accessible file for sharing.

19.7.2 Format Tabs

On the right of the panel is the tabbed view containing all of the options used to configure each element in your script format. Select the element from the list on the left, and then select the appropriate tabs needed to make your adjustments. This section will describe the settings available in each of these tabs.

Font Tab

The Font tab provides options for setting the character appearance for the selected element, as follows:

Font Allows you to set the font family for the current element, and the size of the font (in points) to the right. Scrivener does not use the font to identify elements, so you can change the font here without messing up Scrivener's recognition of script elements. This also holds true for font changes made in the editor.

Scrivener comes with Courier Prime[4] installed within it, and will use it by default over the standard system Courier. If you have Courier Final Draft installed on your machine, it will however prefer that font.

Style Provides a number of options for determining the appearance of the current element, mostly self-explanatory. The **All caps** style option will convert the literal text case to all caps, rather than simply displaying it that way. Be careful when experimenting with this setting on existing text, as it is not possible to convert capital letters back to whatever they were before conversion.

Style to Defines how much of the line to apply the style settings (above) too. The default setting, "End of line", styles the entire paragraph from start to finish. If "First tab" is selected, the element will styled up until pressing Tab. This accommodates formats such as the UK stage play format, which has character names and dialogue on the same line, with character names in capital letters followed by a tab and dialogue in normal sentence case. If "Character delimiter" is selected, you can enter the keyboard characters into the text field to the right that will terminate capitalisation. For instance, if you entered a colon in the text field, the element would only be capitalised up to the first colon.

[4] https://quoteunquoteapps.com/courierprime/

Italicize text after style delimiter When **Style to** is set to anything other than "End of line", the remainder of the text on the line will be italicised.

Colour and Adornment Set the text colour used for this element in the text editor, or automatically add parentheses around the text you enter.

- *Parenthetical*: encloses the current element in brackets when you select that type.
- *Color*: The text falling within the range of this element's definition will be recoloured as specified here. To change the colour, click on the chip and select a colour using the palette.

Menu shortcut Enter the single letter here that will be used in the script elements menu as a shortcut key.

Paragraph Tab

The paragraph tab provides paragraph formatting for the selected element:

Alignment Sets the current paragraph alignment (left, centred, right or justified).

Spacing Sets the line-height multiplier (single, 1.2, 1.5 or double). Some script formats (such as Screenplay) will have a **Fixed Line Height** setting, behind the Options button in the lower right-hand corner of this pane. You will need to set the fixed line height to opt to disable that feature, before seeing any effect on this setting.

Spacing Before Sets the number of blank lines to appear between the current and previous line.

Ruler Conversions

When working with units in Scrivener, bear in mind that its ruler starts at margin zero instead of paper zero. Since Scrivener is, by and large, not "aware" of paper settings and page layout, it counts its ruler settings from the beginning of the text on the left end of the page, not the beginning of the paper itself. This is in contrast to many word processors, which start measuring at the left edge of the paper, and show the print margin buffer in the display of the page.

Consequently, to convert most standard measurements to useful values here, you will need to factor in the standard amount of print margin used to print the script. For example, if the Scene Heading is specified to begin at 1.5" from the edge of the paper, you will need to subtract 1" from that and input 0.5" into Scrivener, since an additional inch will be added to the layout, once margins are added to the page outside of Scrivener.

Indentation Allows simple customisation of the ruler layout for each element, providing support for most if not all formatting requirements on the page.

- *First Line Indent* Sets the left indentation of the first line of the paragraph, a value which will only impact the first line and no subsequent lines from the same paragraph.
- *Left Indent*: Sets the left indentation of all lines in the paragraph excepting the first. Set this value identical to the First Line Indent, to create a uniform "block indent" look.
- *Right Indent*: Sets the right indentation of the paragraph. This setting pertains to all lines.
- *First Tab*: Sets the first tab stop of the paragraph. (Note that subsequent tab stops may be added automatically to differentiate elements should any elements have the same formatting.)

Options

Further options for advanced paragraph settings and compile output options for this element are accessible by clicking on the Options... button in the lower-left corner of the tab.

Writing direction Set right-to-left, left-to-right or "natural" (detect based on the system language input settings).

Fixed Line Height In most cases this should be set to the font size in use (12pt by default), to keep lines absolutely consistent, as is expected in most scriptwriting standards.

This should be disabled if you need to use the **Spacing** setting in the main Paragraph tab—but do note this comes at a risk of line-heights being slightly different from Final Draft measurements. You will also want to disable this feature if you intend to use a font that deviates from Courier 12pt, as font clipping can occur with larger fonts.

Default Tab Interval Similar to the **Default tab spacing** setting in The Tabs and Indents Tool (section 15.5.1). This sets an default tab spacing, rather than inserting physical stops you can see and move around on the ruler. Set to "0" to disable.

Keep with next paragraph Strive to keep this element bound to the following paragraph in cases where the two might become separated by a page break.

Add prefix when compiling Inserts text in front of anything else you type in yourself. This will most often be useful in conjunction with a placeholder counter that numbers elements, but the choice is yours. In particular, take a look at the <$np> tag, which counts by numeral (1, 2, 3...) and resets itself automatically on *each page* (when using print/PDF).

Only add prefix when compiling to print/PDF Mainly useful when using the aforementioned placeholder counter, which only works with these two compile types.

Sequential numbering This is the option you are looking for if you need a way to number your scene sluglines, using the standard method of placing the number along the very outside edge of the typing area. This can be an otherwise difficult look to achieve with Scrivener, since you ordinarily cannot place text outside the physical margin area of the page.

Numbering can be done "against left margin", "against right margin" or "against both margins". Set to "None" to disable this feature.

Tab/Return Tab

The Tab/Return pane allows you to control the tab and return behaviour for the selected element; that is, it lets you specify what happens when you hit the tab or return keys, and so thus can be used to aid in the flow of writing. If you are creating a scripting environment from scratch, you might wish to save this step for last, since you will need to reference other elements (which may not exist while you are going through the list, initially).

On return Sets which element formatting the text will use when you hit the Return key. Using Screenplay as an example, for the "Scene Heading" element, "Action" is selected for **On return**. This means that if you hit the return key after typing a Scene Heading, the text will automatically be formatted and ready to type an Action element.

Use the **Add colon before return** option if the script format dictates separating these two elements with that character.

Tab behavior Sets what happens when you hit the tab key.

— *Allow tabs*: If this is checked, the tab key works as it would normally, that is, it inserts tabs. With this checked, none of the other options for tab behaviour are available.

— *Tabbing on an empty line*: Choose an element from the **Go to** dropdown to set which element will be selected when you hit Tab on an empty line. This facilitates switching to a common alternate element using the tab key. For instance, in the Screenplay format, if you are at the beginning of the line using the Scene Heading format, you can then hit the tab key to reformat the line as an Action element provided you haven't typed anything yet.

— *Tabbing after typing*: Sets what happens when you hit the tab key after typing something in the current element. Choose which element to jump to next from the **Go to** dropdown (inserting a new line automatically). Use this to provide convenience alternates to the most

common behaviour, which would be better assigned to the !return key.

Insert Both of the tabbing options above provide for an alternate behaviour of inserting text when pressing tab, instead of switching to another element. By providing text in either of the **Insert** fields, the **Go To** behaviour will be disabled. To extend off of the example above, hitting Tab after typing something in a Scene Heading element automatically inserts a hyphen surrounded by spaces, so you can easily enter a chronological marker.

The **Insert** field also takes a special placeholder tag, `<$mediaPlaybackTime>`, which will insert the current playback time for the active media player in the other split. If no media player is currently available then the placeholder will print question marks instead. The "Transcript" script format, meant to be used for transcribing interviews, makes use of this placeholder.

By default the placeholder will use the time format specified in the Behaviors: Playback preference pane (subsection B.4.7), under **Media time stamp format**. It is possible to override the application preference for specific formats by supplying a time stamp format as part of the placeholder, like so: `<$mediaPlaybackTime:HH:mm:ss.SSS>`.

Auto-Complete Tab

The Auto-Complete tab allows you to set a custom list of words that will appear for auto-completion while typing in the current element.

Use the + and − buttons below the list to add or remove new words to the auto-complete list, and double-click on words to edit them. By default, these phrases will be suggested automatically while you type, and the first letters typed in will match a phrase in this list, you will be presented with the auto-complete list. Note that the auto-complete list only appears after you start typing a word. For each entry, you can check the "Go to Next Line" box, which will force the editor to move to the next element (as it would if you pressed Return) once it has been entered.

Include project completions for this element Adds any words set in the Project Settings: Auto-Complete list (section C.6) to the current script auto-complete list. Otherwise this element will only offer completions from the list above.

Automatically add phrases to project list that occur Type in two characters to help in isolating useful part of a phrase (such as locations and character names) from the paraphernalia surrounding them. If a field is left blank, then anything on the line will be used, up to any character defined in the opposing field.

An example for the Character element would be to add anything typed into the character field prior to an open parentheses, and thus avoids such common markers as "(O.C.)". However no starting limiter is provided, so anything typed into the line will be added to the project list.

After project list completions When a project list auto-completion has been used, you can set up the scripting system to do nothing, go to the next line, or insert a tab.

An example for the Character element would be to go to next line after using a project list completion (most likely a character name).

Return to chapter ⤴

19.8 Using Script Formatting for Other Purposes

While the Scriptwriting mode has been preconfigured and designed for scriptwriting, it deserves to be mentioned that it is at its heart nothing more than an automated styling engine. This means it can be used for a variety of purposes having nothing to do with scriptwriting.

We have included two such examples in the software:

— *Interview*: simply alternating boldface and normal formatting, as is common for indicating dialogue between an interviewer and interviewee.

— *Transcript*: if you are transcribing from an audio or video file and need to make notation of the timestamps at each point of dialogue, this script format will make that easy.

To use this script mode, you will need to have a media file loaded in a second editor split. Pressing Tab on a new "Time & Text" or "Speaker, Time & Text" line will insert the current media time stamp. Refer to Viewing Multimedia Documents (section 8.1.3) for further transcription tips.

If you need to create a structural document, and cascading between structural types would be of benefit, then consider using the Scriptwriting engine to create you own format from scratch.

Return to chapter ⤴

Writing Tools

20

In This Section...

20.1 Goals and Statistics Tracking Tools

At some point, most writers will need to get some idea of the progress of their work by checking the word, character or page count. There are several ways of doing this in Scrivener, depending upon the scope you require:

— To get detailed statistics and word frequency tabulation for the whole of the draft (that is, the contents of the Draft folder as it will compile given the current settings) or for any documents selected in the binder, use Project/Statistics... and click on the "Compiled" tab.

— For the same sort of detailed stats on the current document or selected items, click the "Selected Documents" tab from the same Project ▸ Statistics... panel.

— To set a target word or character count and track your progress for the entire draft or the current writing session, use the Project ▸ Show Project Targets menu command, or hold down the Option key and click on the Quick Search field in the main application toolbar.

— To set a target word or character count and track your progress for a single document, click on the target button in the bottom-right of the footer view (this is not available in scriptwriting mode, though, where word and

character counts are rarely useful anyway). See information on the footer bar for more information.

— Tracking a *group* of documents together as a cohesive unit, like say a sequence of sections and sub-sections within a folder, where the important detail is the word count of the entire folder is possible. You can use the Outliner to view the Total Progress and Total Goal columns, to aggregate statistics for group of documents. Setting a target on the chapter folder will, even if it has no text itself, work as an aggregate in Scrivenings mode, since child items will be included in the totals. Thus all words written to any subdocuments of it will count toward that aggregate goal, even if they themselves do not have individual goals. Targets for folders can be easily set by using the Target column in the Outliner, or by selecting the folder in the Binder, switching off the current group view mode, and using the target button as per normal in the text editor footer bar.

— To get the word or character count for a selection of text, select some text in a document and examine the footer bar of the editor. The word and character count will appear in blue text. When there is no selection, this area will be used to count the entire document or Scrivenings session.

— To view the word, character counts and targets for several documents at once, use the outliner view and make visible the appropriate columns by clicking on the › button in the outliner column bar (or by selecting View ▸ Outliner Options ▸ submenu). You can use the "Total Words" column to collect aggregate counts for items which have children.

— To view a combined word and character count of an arbitrary selection of documents, use the outliner or corkboard to select several documents, and then right-click on the selection. The combined word and character count for those documents will appear greyed-out at the bottom of the contextual menu. Note this method only counts texts within the actual selection, not the implied selection in the case of children beneath the selected items. To count them too, disclose the items in the outliner and select them along with their parent items.

20.1.1 Project Targets

The project targets panel allows you to set goals for your writing—either for the Draft as a whole or for the number of words you want to write during the current session—and to check on your progress as you work, whether to grow the word count or trim it down. To bring it up, select Project ▸ Show Project Targets or press ⇧⌘T to toggle its display.

The targets panel displays two progress bars: one showing the progress of the

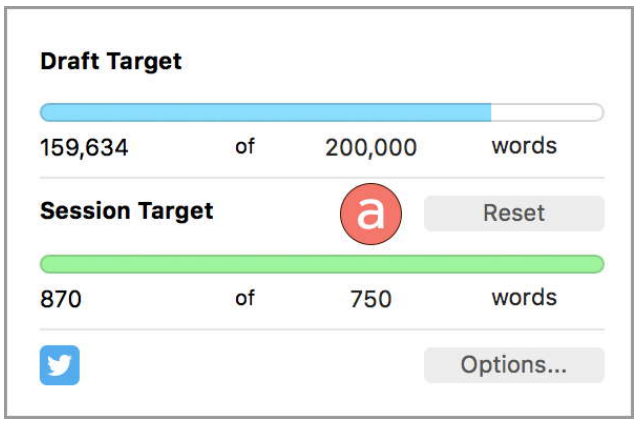


Figure 20.1: The blue progress bar shows overall progress for the project, while the green progress bar indicates the writer has achieved their session goal and is now writing in surplus.

draft and the other showing the progress of the current session.[1] Using this panel you can set a target word count for the draft, or presumably your current work-in-progress, and independent goals for the current session.

To set up targets:

1. Click in the appropriate text field—marked above or below (a) in Figure 20.1—and enter your target.
2. Set whether the target should be in words, characters or pages by clicking on "words", to the right of the number fields.

This panel will float over your project if you leave it open, and update itself in real-time as you write and edit. If you want to track a goal for an individual document, rather than the entire project, use the Document Goals (subsection 20.1.2) tool in the footer bar. By default, the Quick Search tool in the toolbar (subsection 11.5.3) will also display small progress bars correlating to this information. So you needn't keep this panel open if all you want is a small visual reminder of how far you have yet to go.

Using targets to hit an editing goal

The targets tool needn't only be about writing more words to reach a goal, it can also be used to establish a word count you'd like to cut down to. Use the **Show overrun** setting, discussed in the following section, to make it so a secondary progress bar will inform you as to how far *over* the target you are.

[1] By default it will automatically reset shortly after midnight, but there are options available if you find this inconvenient, discussed in the following section.

If you delete lots of text, session statistics will not start showing any progress until you have written as much again—it is perfectly possible to have a negative session word or character count! In other words, it shows your net gain during the session. The session target only counts anything typed or pasted into a main text view (either of the editors or the composition mode editor) it does not count imported documents or appended text and so forth. The basic rule of thumb is that if you what you did to add or remove text was done inside of the text editor, then it will be added or subtracted from the count. If you used menu commands to move, generate, duplicate or copy text then it will not be counted.

If you find yourself in a situation where you want to start fresh, because the counter is off, you can use the Reset button above the Session Target progress bar to reset the counter to zero.

In the Project Targets panel, you will see a Twitter icon, where you can share your progress. Scrivener will automatically generate a short sentence for you, based on any activity (or lack thereof) that has happened during the session. If you write 800 words, then it will select a phrase extolling how much you've written. If you've subtracted 800 words it will assume you have been editing that day. If there is no change, it will assume you have just started working for the day. By default it will use the @ScrivenerApp and #amwriting tags to help your tweets fit into the common network. You can edit the text that Scrivener supplies by changing it in the editor area. To use this feature, you must have a Twitter[2] account set up in your macOS System Preferences.

Project Targets Options

Click the Options... button in the Project Targets panel to configure how these progress bars are calculated, and what Scrivener should do if you meet your goals. There are a couple of common options available at the bottom of the pane, below the tabbed areas:

Show target notifications System notifications will be posted when the status of your targets change. For example if you achieve the writing goal for your current session. Notifications will be posted for the following conditions:

- The draft or session goal has been achieved.
- Editing you have done has cause the count to dip back under the respective goal.
- When **Show overrun** is enabled in the "Draft Target" tab, if you go over the allowed amount you will be alerted.
- Likewise you will be alerted when you edit back down under the maximum allowed amount.

[2] http://www.twitter.com

Show Twitter button If you do not use Twitter you can keep the button from appearing.

Draft Target

Count current compile group only Off by default, when enabled, only those files descending from the "Compile" group selection designated in the contents tab of the compile settings area (subsection 23.4.1). By default his will be the entire "Draft" folder, but it can be set to any subfolder, or even to dynamic sets of files such as the results of the last project search you ran, or your current binder selection.[3]

This setting *only* pertains to the compile group selection itself. Filters that subtract from the group or the front and back matter features, which add to the group, will not be considered by the targets tool.

Only count documents set to be included in Compile Will only consider those texts which have the "Include in Compile" checkbox set in the overall count.

Show overrun Modifies the behaviour of the progress bar so that once you achieve your goal, a second progress bar (in bright red by default) will begin to advance from left to write, depicting how many words *over* your target you are. This can be a great tool when editing, and looking to trim the overall word count down by a certain amount.

Overrun allowance This subsidiary option to the above provides a threshold by which you can exceed your stated goal before triggering the overrun condition and seeing the red progress bar appear. This value subtracts from the overrun amount—in effect once an overrun condition is reached, *then* the progress bar starts counting from one on up. Thus a goal of 100,000 words with a threshold of 1,000 will consider your overrun amount as being only "10", if your draft has 101,010 words.

Deadline When active, adds a countdown toward the days you have left at the bottom of the target window. This can also be used to calculate how many words you need to continue writing per session in order to meet your deadline, in the Session Target tab.

Session Target

[3] In the case of collections and search results, the list of items designated to be counted will only be refreshed after visiting the compile overview, even if only to load the pane and then cancel it.

Reset session count There are four available options for how the session counter should behave. The counter can be reset every time a project is closed, you can have it track the time of day and reset it for you, or turn off automatic reset entirely and handle it manually with the reset button.

- *At set time each day*: the default behaviour is to reset the session target at 1:00 in the morning. This will occur even if you are writing mid-sentence, so if you're a night owl you might want to change when the reset happens, or use one of the other options.
- *On project close*: if you prefer to consider a session concluded whenever you close down a project then this will be the best option. If you prefer multi-day sessions or perhaps prefer short writing bursts throughout the day, this will be a good option for you.
- *On next day opened*: with unusual schedules, such as those who write well after midnight and unpredictably so, this alternate method of reset may work better. It will check against the last time you opened the project, and if it the calendar day has incremented since then it will reset. Thus, if you open the project at 22:00 and work until 03:00, when you open the project later on that day at 13:00, it will reset since you last opened it on the prior day at 22:00. If you start another session that night, however, it will not reset, since the last time you opened it was at 13:00 on that same day.
- *Never*: Scrivener will never alter the session progress counter for you. You will need to use the Reset button to start a new session. This is the best mode if you tend to work sporadically all around the clock in short bursts, or tend to set aside a "session" for a few days to deal with other tasks.

Count text written anywhere in the project Turning this on will count anything you type or paste into the *any* document in the project binder, even if it is a character sheet or a grocery list.

This setting also impacts whether documents that are not marked as "Include in Compile" in the inspector are counted. When this setting is off, such documents will not be counted even if they are in the draft folder. If the option is enabled, even those documents marked as excluded will be monitored as you write.

Allow negatives When disabled, the session counter will never drop below zero. Leave this on to get an accurate net total of your writing session. When disabled, deletions will still be counted, but only until the counter reaches zero, so some deletions would no longer be counted after that point, making it less accurate for calculating the true net.

Automatically calculate from draft deadline Requires the **Deadline** option to be enabled in the "Draft Target" tab, and the counting method to be either Words or Characters (set in the main Targets panel itself).[4] When a deadline has been set, you can have Scrivener handle the calculation required to meet your deadline given the amount of time left. For example, if you are at 85k words in a 100k draft and have 10 days left to finish, then Scrivener will set your daily session goal to 1,500 words per day. If you come in under or over that goal, Scrivener will adjust the daily session target whenever the session count for any reason.

Writing Days By default, all days are considered eligible for writing, in terms of calculating your daily goal. If you cannot write every single day of the week, simply click on the days you can write, and the feature will adjust the calculation so that you don't end up writing below the curve. If all days are deselected, the calculation assumes you can write on every day.

Allow writing on day of deadline The calculator typically will not schedule you to be writing on the actual deadline day. If you want the system to give you up until the last minute, check off this box.

20.1.2 Document Goals

Each document in your binder is capable of storing its own independent goal, which can be great tool if you're trying to meet ration out how many words different sections of your chapters should have in order to stay below a larger target. To set a goal for a document, click the target iconon the right hand side of the footer bar (Figure 20.2).

Figure 20.2: Click the "bull's eye" target icon to set a goal for this document.

When a goal has been set the target button will turn into a small progress bar that will fill up as you type, turning green once you exceed the goal. Trimming sections down in editing instead of writing? You can also set an overrun warning, which will use a different colour to show how far *over* the target you are.

Target for this document Sets the numerical goal for the document, using words or character counts. If you select a different form of accounting than

[4] This is down to technical limitations in how pages are calculated in Scrivener rather than something that can be easily quantified as you type in snippets of text here, there and everywhere throughout the binder or draft folder. This is why session counts cannot be calculated by page either.

what is being displayed in the footer bar, then no x/y information will be shown in the footer bar, but the bar will still show your relative progress. (You might also want to change the footer bar to show the stats important to how you work, with the **Live counts show** option in the Editing: Options preference pane.

Minimum target This alternate mode of usage will draw a small indicator on the progress bar at the declared minimum, and until you reach that point the progress bar will be coloured a salmon shade, snapping to blue as soon as you reach your minimum goal.

Show overrun Instead of merely turning green once you exceed the goal, with this enabled the progress bar will start filling up again, this time in bright red and indicating the amount of text in excess of the goal. This is a great tool when you're working to trim down the word count.

Overrun allowance Rather than going into overrun mode immediately, this sets a threshold before the warning bar kicks in. For example if you set a goal of 600 and an overrun allowance of 50, the bar will remain green from 600 to 650, but as soon as you hit 651 the overrun state will kick in.

Show allowance in progress bar If an overrun allowance has been set, a tick will be added to the visual progress bar that marks the actual target of the document, with the remainder of the progress bar showing how far you can continue writing before overrunning. As you write into this threshold, it will fill up until the document goal plus the overrun allowance has been exceeded.

Show target notifications Show system notifications when you reach different states in your progress. The various options above all add additional notification states. With everything enabled, you will be alerted when you reach your minimum goal, when you reach the overall goal, when you go *over* your goal—and then all in reverse as well as you edit back down through the different stages.

Need a bigger picture?

You may also monitor and set them using the Outliner, by revealing the "Target", "Target Type", and/or "Progress" columns, while the "Total Goal" and "Total Progress" columns will show an aggregated goal count and total progress toward that sum, for the displayed outliner row and all of its descendant items. In this way, you can easily work toward chapter or other larger section goals, while still maintaining a fine-grained approach to cutting up the section into smaller pieces.

To disable the goals for a document, enter a value of "0" into the Target for this document field for the document.

Tracking Goals for Groups

When using Scrivenings mode, a progress bar will be shown if any documents in the session have a goal set. In this mode the progress bar becomes a "total progress" tracker, adding all of the goals and various settings together and using the grand total in the editor against that goal. This can have unintended results if, for example, you set a goal of 200 words for one document but then form a session including it and other files for a total of 1,200 words. It would be better to set goals for each document in the group with an eye on the total.

A simpler way of making use of this capability would be to set a single goal on the group that contains the files as a whole. For example if you load the chapter folder itself into the text editor you can set a goal for the entire chapter and not worry about having to set individual goals for each section of text within the chapter. Now when using Scrivenings mode on the chapter, you will be provided with a sensible goal with all of the constituent parts of the chapter being used to add toward the goal.

20.1.3 Statistics

Project and text statistics can be called up at any time with the Project ▸ Statistics... menu command (⇧⌥⌘S). This pane has two info tabs and some options:

1. *Compiled*: with one exception, this will list statistics for a total reckoning of all documents in the Contents tab of the compile overview's settings area (subsection 23.4.1). The main exception is that by default if you choose only one part of your draft to compile from, this setting will be ignored. Thus you can compile one chapter for proofing but still use this tool to keep track of the larger work.
2. *Selection*: a count of those documents you have selected in the binder or any group view, including the current and active Scrivenings session. This accounting also includes subdocuments of the selected items by default.

> **Upgrading from Scrivener 2**
>
> Looking for "Text Statistics" pane from previous versions of Scrivener? The "Project Statistics" and "Text Statistics" panes have been combined into a single feature, Project ▸ Statistics... (⇧⌥⌘S), with two tabs. The latter now has every feature the former once had exclusively, and vice versa. To get statistics on a single text document, view it in the editor alone and then use the "Selected Documents" tab. You can click on statistics in the footer bar of the editor and get much of the same information provided here.

Each of these sections have identical statistics available to them, most of which are self-explanatory and will not be documented here:

— *Documents*: lists the total number of binder items being used to generate the statistics. This includes items that are not contributing to the count, such as empty files.

— *Pages* (*Paperback*): this is an estimate, using an industry standard formula (for English language publishing) of taking the average number of words per page and multiplying it by the average number of characters per word (five including a space, for six total), the product of which is then used to divide against the total character count of the project. By example, a book with 720,000 characters with an estimate set to 250 words per page will produce a result of:

$$\frac{720000}{(250 \times 6)} = 480$$

Options for tweaking the algorithm are located in the "Options" tab, under **Page count options**.

— *Pages Printed*: this counter will be more accurate as it will compile your draft in the background, using the specified formatting and other content settings, and then count the total pages resulting from that. It requires the accuracy model be set to "Accurate (Slower)", as described below.

— *Reading Time*: this is a simple calculation based on an average 250 words per minute.

Hidden by default, the **Word frequency** section at the bottom will provide a complete concordance of every word found within the relevant texts being counted, along with how often the word has been used, with a graph showing its relative frequency in comparison with the rest of the words. By default the list is sorted alphabetically, but by clicking in the column headers you can sort by "Count" in ascending or descending order.[5] Words from this chart can be selected (using Cmd and Shift clicking to select individual words and ranges of words, respectively; ⌘A will select all) and copied and pasted as tab-delineated text, suitable for use in a spreadsheet for more thorough analysis.

At very the bottom of the pane, alongside the OK button, is a dropdown that selects between two different accuracy models. Accuracy pertains to how all statistics are gathered:

— *Accurate* (*Slower*): to accurately count all relevant text, and particularly to generate a real page count, Scrivener must internally generate a compile document based upon your compile settings.

— *Estimated* (*Fast*): this model uses the fast internal search index to count the relevant text. This means all alterations made to the text during the com-

[5] You can technically sort by "Frequency" as well, but this will provide an identical list as sorting by "Count".

pile process will be disregarded, and a true page count will be unknown, but in larger projects this will be the only viable option, and is thus the default setting.

Project Statistics Options

To access options for how project and selection statistics are calculated, click the "Options" tab in the window. The first group of options, under the "Compiled Statistics Options" heading, impact how the statistics in the first tab of this pane are calculated:

Count current compile group only Only calculates off of documents that have been chosen in the Content pane of the Compile interface. This is the only compile-time option that can be disabled. All other compile options that restrict or modify output quantity will still be factored into the count.

Count footnotes Footnotes are by default included in the count. If your publishing environment demands these be considered separate, here is where you can disable them in the total count.

The second group of options, "Selection Statistics Options", impact how statistics for the second tab in this pane are calculated:

Count all documents In this context, "all documents" refers to whether the inspector option, "Include in Compile" should be ignored.

Count only documents marked for inclusion Only those documents that have "Include in Compile" checked will be counted.

Count only documents not marked for inclusion As above, only with the inverse logic.

Exclude comments and annotations The running commentary for a piece, its inline annotations and comments, will be counted by default. Check this box to only count text that would generally be exported or printed.

Exclude footnotes Footnotes are by default included in the count.

Count subdocuments By default the selected items *and* all of their children will be counted, all the way down to the bottom of the outline. When disabled, only the literal selection will be counted.

Page Count Options Set the counting algorithm used to estimate paperback page counting. The default presumes a "word" to be an average of five letters long plus a space, for six characters total—thus by default per page:

$$350 \times 6 = 2100$$

per page. This is a fairly safe estimate (and an industry standard) in English publishing.

If you're writing in a language that differs substantially, it might be better to switch this to "characters per page" mode so you can more directly fine-tune its results.

Word Frequency Options This is a global option for all projects. Click the Set List of Words to Ignore... button to bring up a text field where you can insert words that the word usage frequency tables should leave out. In English for example, it could be useful to leave out common articles such as "a the at" and so forth.

20.1.4 Writing History

This feature, accessed via the Project ▸ Writing History... menu command, offers a thorough reckoning of the daily word count progression (or regression as the case may be) on a daily, monthly or combined basis. Even if you do not make use of the session tracking feature, Scrivener will dutifully record your net increase or decrease in word count (both in and out of the draft folder) on a daily basis. This information is then used to provide monthly summaries and produce overall averages.

The top portion of the pane features these overall statistics:

— *Writing days*: the literal number of days in which the project was opened and some change as made to it—even if only to add or delete one word.

— *Average words written per day*: is broken down by those words written in the "Draft" folder vs everywhere else in the binder, and a total of these two.

The central and largest portion of this window displays detailed statistics to a resolution determined by the dropdown setting in the upper right-hand corner (set to "Months and Days" by default):

— *Months and Days*: all recorded information will be shown, with each day you have written be grouped together into monthly sections. The monthly rows will contain a sum of all the day counts within them.

— *Months Only*: only the monthly summary rows will be displayed. If you've been working on a project for a long time, it might be more useful to see the bigger picture this mode affords.

— *Days Only*: if it doesn't really matter how many words you write in a month, use this mode to display a simple list of every day committed to the project.

The list will be sorted in reverse-chronological order by default. You can click on any of the column headers to sort by a different value. You could for example click on the Total column to see which days or months of the year were your

weakest and which were the strongest. Negative numbers indicate that on that day you cut more words than you wrote.

The individual rows in this table can be selected, and when doing so some calculations will be performed for you, displayed in the area below. This information will be identical in format to the overall summaries provided at the top of the window, only focussing specifically on the month or day you've selected in the list, plus a few extras:

— When selecting a day, if a session target was set for that day it will be recorded.

— When selecting a month, the number of days you wrote in that month will be printed under "Writing days". In addition to the monthly summation, you can choose to view:

  – *Average per day*: your averages within the confines of that month.

  – *Minimum in a day*: the count for the lowest session count (by total) that month.

  – *Maximum in a day*: the count for the greatest number of words written (by total) in a session during that month.

Lastly, the Export... button brings you to a save panel, where you can dump the full writing history to a CSV file, suitable for further analysis in a spreadsheet program or similar. There are a few options available for how this file should be formatted.

— You can choose to "export days" or "export months", to set the granularity of the data.

— Select which columns should be exported. While ordinarily you can only view statistics in Scrivener as either words or characters, with the raw output you can export both types of data.

Return to chapter ⤴

20.2 Proofreading Tools

Scrivener has a few tools available to make the process of proofreading your work a little easier. Whether you prefer to generate a static copy of your text, so that it can be read in a different context than where you write, or if you prefer to proofread and edit directly in the same text editor you use to write, we have two different tools available.

20.2.1 Cleaning the Editor View

Being able to proofread and edit in the same environment is convenient, but depending on how you work, you may find some of the tools Scrivener uses to aid in the writing process to be a distraction. There are rulers and toolbars, hyperlinks in the text, comments, style markers and so on.

1. Firstly, the traditional and longstanding tool available for paring down the interface is Composition Mode (chapter 17). With a simple command you can be whisked away into an environment where all you see by default is your text, not even the rest of the computer that is typically around your project window.

2. For some that may be too much however. After all split views are a fantastic editing tool. Being able to compare older revisions via snapshots with the current text, jumping around using all of the navigation tools the main project window affords you is too much to give up. You could consider using the built-in "Editor Only" layout, with the Window ▸ Layouts ▸ Editor Only menu command (or from the "Layouts" submenu of the View button on the toolbar. This removes the inspector and binder, and reduces your layout to one simple editor (the active one at the time of invoking this command). You can further hide the Format Bar and main application toolbar for an even cleaner interface.

3. Now for the text itself, the View ▸ Text Editing ▸ Hide Markup[6] menu command will by default hide the following elements:

   — Comment highlights

   — Footnote highlights

   — Highlight boxes around styled text

   — Preserve Formatting markup

   The scope of what it hides can be configured in the Appearance: Textual Marks: Options preference tab. It is important to consider that these markings will merely be hidden. They will go on being functional despite your being unable to see them, and typing in and around them will follow all of the same rules that would apply if they were visible. For example, if you put your cursor at the end of a styled phrase that is only indicated as being styled by its highlight box, and proceed to type several hundred words, all of those words will be assigned to that style unbeknownst—until you toggle markup back on.

[6] No, this feature doesn't have anything to do with the sort of markup you type in yourself, such as with Markdown.

Consequently it might be best to primarily read the text in this fashion, and if you want to edit something within it, open a split with Show Markup enabled.

20.2.2 Linguistic Focus

The Linguistic Focus panel can be brought up with the Edit ▸ Writing Tools ▸ Linguistic Focus... menu command (^⌘L). This tool will fade out the text in the main editor, leaving only the selected part of speech highlighted. In this way you can highlight all adjectives, adverbs, pronouns or whatever else you wish to focus on, and freely edit the text while in this focussed state.

Adjust the amount of fade used to dim out unmatched text with the **Fade** slider, at the bottom of the Linguistic Focus panel.

For performance reasons, all edits will remain fully visible, rather than being evaluated for their parts of speech while you write. To update the display with your newly added text, click the circular arrow button in the lower right corner to refresh the filter.

The "Direct Speech" option will attempt to find and highlight any text found within quote marks. The marking punctuation used will match those selected as smart quotes, in the System Preferences: Keyboard: Text preference pane. If you make use a marking system that denotes dialogue with an em-dash at the beginning of the paragraph, you will need to enable **Linguistic focus uses Spanish-style dialogue**, in the General: Language preference pane (subsection B.2.5).

When you are done analysing your text, close the panel to return the editor to its normal display.

20.2.3 Inserting Section Links Into Proofing Copies

For those that prefer a "hard copy" (even if entirely dealt with digitally), you can have the compiler insert a link pointing back to every section of the binder that was used to compose that copy. If for example you are proofing chapter 18 and it is comprised of 30 some subdocuments, this option would insert 30 hyperlinks into the compiled file, each pointing back to the specific chunk of text that follows the link. Here is a simple way to test the feature and see if it is right for you:

1. Use the File ▸ Compile... menu command.
2. Click on the General Options compile settings tab (subsection 23.4.3) (gear button).
3. Enable the **Insert links back to Scrivener in each section** checkbox.
4. With **Compile for** "PDF" set at the very top of the compile pane, click the Compile button and select a convenient location to save the proof copy.
5. Once compile completes, split the editor using View ▸ Editor Layout ▸ Split Vertically (or use whichever method you prefer).

6. Drag the proofing PDF into either editor's header bar to load it directly into the split (no need to import it first!).

7. Now you can proofread the "hard copy" in one split, and if you spot an error, click the "Open in Scrivener" link located at the top of the section you're currently reading within.

   With default settings you will find the section loads in the other split.

That is of course only one way in which to use these links. You do not have to use PDF, nor do you have to load the file into a split—these links work from anywhere on the same machine—so feel free to use your favourite PDF reader, or an eBook reading program with ePub, a web browser with HTML files, a word processor, Markdown previewer, etc.

This method can be used to distribute proofing copies to readers. When importing their annotated copy of the draft back into your binder, you'll be able to jump directly to the section their notes pertain to via these links.

In some case you might need to use a format that doesn't support hyperlinks (such as various plain-text formats), or you may find that the special hyperlinks Scrivener inserts are being damaged by whatever software your proofers are using. In that case, the **Insert unique document identifiers only** option (directly below the aforementioned compile setting in step 3) will insert a plain-text marker that Scrivener will scan for and convert back into a document link when the file is imported into the binder. You will need to communicate the importance of these markers to your proofers, so that they can work around them.

20.2.4 Converting Document Links to Scrivener Links

In a similar vein, you can have all document links in the text converted to external Scrivener links, meaning when you click on the link it will load that section of text in your original project. This options allows you to carry over your internal link usage to proofing copies, though of course since it relies upon the original project being where it was when you compiled, the method will likely not work once you take the proofing copy off of the original computer.

For example, a cross-reference from a scene in chapter 8 that points to a character sheet elsewhere in your binder, the link will be retained in the compiled copy (ordinarily links to items not found in the draft would be stripped) and clicking it will load that character sheet in your project.

To enable these links:

1. Use the File ▸ Compile... menu command.

2. Select PDF or one of the word processing formats.

3. Click on the General Options compile settings tab (subsection 23.4.3) (gear button).

4. Enable the **Convert document links to link back to Scrivener** checkbox.
5. Click the Compile button and load the file in your preferred editing or reading program.

Return to chapter ⤴

20.3 The Name Generator

Scrivener comes with an exhaustive name generator, which includes many thousands of common names, as well as selections such as dictionary words that sounds like names, literary names from classics, a broad selection of regional names in several languages, and can even attempt alliterative names—all with extensive options for setting relative obscurity, naming styles such as double-barrelled surnames, initials, and so on.

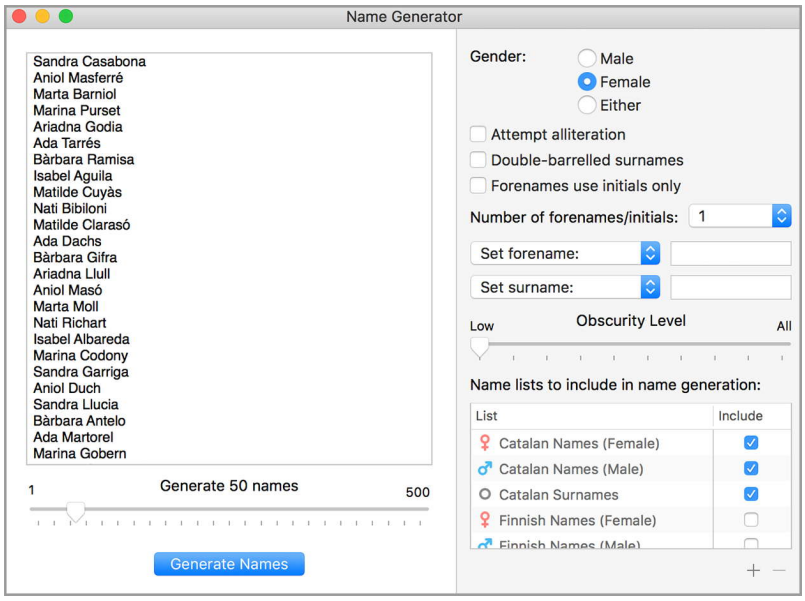


Figure 20.3: The name generator: for when Joe Sixpack needs an upgrade.

To bring up the name generator use the Edit/Writing Tools/Name Generator... menu command and click the Generate Names button to get started. The left side of this window is a text list (empty initially) where the names will be generated. You can select and copy names out of this list to store your favourites. You can also right-click on selected names to run quick searches for them on the Web, to check for existing uses of them.

On the left side of the window you will find options for adjusting how names are generated:

Gender Select one option here. The default is either gender.

Attempt alliteration The generator will attempt to produce names with an alliterative effect, like "Jeromy Jin". This option will work best with Latin based languages.

Double-barrelled surnames Produces names like, "Otis Cowie-Milburn".

Forenames use initials only Reduces the forename to an initial. If more than one forename has been selected in the option below, multiple initials will be generated, like, "N. J. Pettersen".

Number of forenames/initials Produces multiple forenames. You can select from 1 (default) to 3.

Set or search forename and surname The next two fields are multi-purpose tools where you can either set part of the name yourself, or search the database for a name by providing a part of it. For each part of the name:

- Set: forces the respective name to be what you type into the text field; useful if you already have a name in mind but are having troubles with the rest of it.
- Starts with: will only return names from the database that start with the characters you type into the field.
- Ends with: likewise, but for characters at the end of the name.
- Contains: the characters need only be found anywhere within the name.

Obscurity Level This slider adjusts how obscure the names should be on average. Moving the slider all the way to the left might produce a result like, "Scott Young", while sliding it all the way to the right, "Chauncey Noach" (no offence to all the Chaunceys out there).

The bottom of the configuration area is where you will select from the many lists provided as sources for the name generator. You can have as many lists active at once as you like, but you will always need at least one Surname list selected (indicated by the neutral grey circle icon), and at least one gender list which is compatible with the gender option set above.

20.3.1 Managing Your Own Name Lists

You can add your own custom name lists to the generator. They should be formatted so that all names are on a single line, and each name is separated by a comma, like so:

```
name1,name2,name3,name4,name5,...
```

1. It might be easiest to produce these lists in a spreadsheet on a single row, and export as a CSV file. If you use a regular text editor, make sure to name the file with a ".csv" extension.
2. Click the + button, below the name list area of the Name Generator window.
3. Locate the file in the chooser dialogue and click Open.
4. Give the list a descriptive name in the **Title** field, and select whether it is a list of female names, male names, or surnames.
5. Finally, if you have ordered your list from most common to most obscure (at the end of the list), check this box to enable the Obscurity Level slider for that list.[7]

To delete one of your custom lists: select it from the name list area and click the − button. You will be asked to confirm your decision.

To update a list you've imported in the past, delete the custom list, and then add your modified CSV file again. The details from the last name list you created will be filled in for you.

Return to chapter ⤴

20.4 Bibliography Management

Scrivener offers simple integration with your favourite bibliography or citation manager (such as EndNote, Bookends, Sente or Zotero) for academic work. To set it up:

1. Open the General: Citations preference pane (subsection B.2.8).
2. Click on the Choose... button and use the file chooser to locate the application you use for citation management.

Once the citation manager has been set up, you can use the Insert ▸ Bibliography/Citations... menu command (⌘Y) at any time, to bring your chosen software to the front, launching it automatically if necessary.

The steps you take next will depend upon your citation manager, so you will need to consult their documentation on how to use their software with third-party word processors. Typically, you would copy and paste a citation placeholder into Scrivener, in the location where the reference mark should appear. After compiling to RTF, you would then use the citation manager to scan these placeholders into final print form. Not every program or service provides RTF

[7] In small lists, the obscurity slider may not have much impact, depending upon how many names are being generated.

scanning, however. You should research the software you intend to use and make certain it designed to work with software other than Microsoft Word.

Return to chapter ⤴

20.5 Using Equations with MathType

If part of your writing involves the addition or construction of equations, one alternative is to make use of Scrivener's MathType]Design Science[8] integration to insert editable equation objects into the draft, much like you would an ordinary figure.

To create a new equation, position your cursor where you wish to have it appear, and use Insert ▸ MathType Equation.[9] If you have MathType correctly installed on your machine, you will see the equation entry window pop up over the Scrivener window. Any changes made within this window will be saved back into the Scrivener project when you close the window (by default it will ask for confirmation when closing the window) or use File ▸ Close and Return to Scrivener.

Equations can be inserted either on their own paragraphs, as figures, or directly inline within another paragraph. In the latter case, attempts will be made to keep the equation correctly aligned with the text baseline.

To edit an equation double-click on the equation in the Scrivener editor. The MathType interface will pop up again, this time with the equation loaded, and any changes you make will be saved back into the file when you close it.

Otherwise, equations act much like ordinary images. They can be aligned or have paragraph formatting (such as spacing) applied in the editor, and when they are compiled they will be converted to images and handled as they ordinarily would be for the particular format in use.

---

[8] http://www.dessci.com/en/products/

[9] You can also add an optional "Equation" button to the main application toolbar if you use this feature a lot.

Using MathType equations beyond Scrivener

Given unfortunate technical limitations in how the MathType engine handles equations in export, when compiling a document containing them, they will be converted to raster images at that time. This means they will not be scalable without quality loss and can no longer be edited. They are thus not suitable for round-trip workflows where one imports edited documents from collaborators, or for taking your work into other word processors for additional formatting with the equations in an editable form. If you require equations to be editable throughout the life of the document then it would be best to defer the insertion of equations until they can be brought into an environment where they will remain until publication.

Return to chapter ⤴

Using MultiMarkdown and Pandoc

21

In This Section...

For those who prefer structural or semantic writing methods to rich text, Scrivener allows you to import and export using two different Markdown dialects: Fletcher T. Penney's MultiMarkdown (MMD) and John MacFarlane's Pandoc. These systems make it easy to generate documents in any number of formats, from clean and modern HTML5 to LaTeX to DocBook to standard word processing files like Word DOCX—all while using a simple and easy to type in mark up.

If you are curious about the process and would like to consider adopting it, you should read the following sections for an overview of its philosophy, limitations and capabilities. If you're already familiar with the approach, you could skip to Markdown and Scrivener (section 21.5), where we get into the specifics of how this approach can be used within, and is supported by the software.

21.1 What is Markdown?

The "Markdown" syntax was created by John Gruber, and the good description of what Markdown is comes from his his site, Daring Fireball[1]:

> Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).
>
> Thus, "Markdown" is two things: (1) a plain text formatting syntax; and (2) a software tool, written in Perl, that converts the plain text formatting to HTML. See the Syntax page for details pertaining to Markdown's formatting syntax. You can try it out, right now, using the online Dingus.
>
> The overriding design goal for Markdown's formatting syntax is to make it as readable as possible. The idea is that a Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. While Markdown's syntax has been influenced by several existing text-to-HTML filters, the single biggest source of inspiration for Markdown's syntax is the format of plain text email.

Can I use Scrivener to create basic Markdown?

You may not need all of the frills provided by the integrated systems, or are publishing to a system that has its own dialect, such as Github. In most cases, the Markdown that Scrivener generates is done at your request: it will only generate image syntax if you put images in the editor for example. If the dialect you are using does not use footnote syntax, then you should avoid using the footnote feature in Scrivener. The one thing you might need to disable is the special document metadata features that both MMD and Pandoc make use of (section 21.7).

Return to chapter ⤴

21.2 What are MultiMarkdown and Pandoc?

Markdown's goal is primarily toward Web publishing and similar uses. Systems like MultiMarkdown[2] and Pandoc[3] arose out of a desire to use a similarly sim-

[1] http://daringfireball.net/projects/markdown/

[2] http://fletcherpenney.net/multimarkdown/

[3] http://johnmacfarlane.net/pandoc/

ple syntax for the production of formats more suitable to traditional and electronic publishing. The syntax was extended to include constructions desirable to authors, such as footnotes, tables and better cross-referencing. Of the many formats these systems provide, Scrivener supports:

— LaTeX : via MultiMarkdown a well established document typesetting engine primarily used in the sciences and academia for its high-quality rendering of formulae. Beyond that it is a capable engine for any number of purposes, especially technical formats. In fact, the PDF you're reading right now was built using this system.

— HTML5: via MultiMarkdown, Scrivener is capable of producing syntactically clean, modern HTML. It is suitable for the production of eBooks and web pages—in fact we drive our own ePub 3 generator using MMD's output, after internally converting Scrivener's rich text to MMD syntax.

— Flat ODT: This plain-text variant of the OpenDocument format can be opened in LibreOffice and a few other office suites. From there it can be converted to any format you need. This is a high quality word processing document, with figure captions, stylesheets and everything else you'd need to take your work into a production environment.

— DOCX: via Pandoc, this high-quality word processing output is better if you know Word is the primary target. As expect all the wiring you need to continue your project beyond Scrivener.

— DocBook: via Pandoc. This format is commonly used in technical publishing.

— ePub: via Pandoc. Capable of producing both ePub2 and 3 formats. As can be expected from generators based on Markdown, the internal HTML quality is clean and easy to modify and style.

One of the things that attracts people to Markdown is that it focusses solely upon the text. Formatting is not something one bothers with, and instead text is marked as being a kind of thing using simple to type and easy-to-read text codes. For the most part, these are also very easy write with and remember. You don't have to memorise keyboard shortcuts for formatting commands because they all use common punctuation marks, such as putting *asterisks* around a work to emphasise it.

This chapter will not attempt to teach you Markdown or any of the more advanced dialects of it. There are excellent resources available on the Internet for this, including the links at the top of this section. Instead we will cover the integration features provided by Scrivener, and will assume a basic working knowledge of the underlying mark up systems.

Return to chapter ⤴

21.3 Getting Started with the Tools

⟨Direct-sale only⟩ There is very little you need to do to prepare for using MultiMarkdown with Scrivener. It comes pre-loaded with a recent version of it (we tend to lag behind development a bit so as to keep Scrivener's version stable and well integrated). This means that even if you intend to use a more advanced workflow, like LaTeX, you won't need to install anything extra. You are of course free to, and Scrivener will cross-check your system for you. If it detects installed components, it will gracefully switch to using them behind the scenes.

For Pandoc support, you will need to install your own copy of the engine itself. Refer to Setting up MultiMarkdown or Pandoc (subsection 3.2.4).

⟨MAS only⟩ The version sold through Apple's store must adhere to Apple's sandboxing guidelines, which means no utilities can be executed unless they are located inside the /Applications folder. This means the Apple version of Scrivener cannot integrate with Pandoc, pdflatex or support external support files or a custom MultiMarkdown installation. It only supports the built-in distribution of MultiMarkdown.

If these are important ingredients in how you intend to use Scrivener, you are advised to either purchase the direct-sale version instead, or if you've already bought and run the Apple version, you may be able to migrate to the direct-sale version by downloading it from our site and replacing your MAS copy. Further instructions are provided in our knowledge base[4].

Return to chapter ⤴

21.4 Importing Markdown Files

You can import any existing Markdown documents into Scrivener using the typical methods for importing files—they are simply text files and so there is really nothing special required to get that text into Scrivener.

But if you would like to have the document split by headings instead of as one long file, use the File ▸ Import ▸ Import and Split... menu command. This command can break up the document so that it is imported into the binder with its heading hierarchy converted to nested documents in the binder. Refer to the Import and Split documentation for further details (section 9.1.6).[5]

When importing a document with a MultiMarkdown or YAML[6] metadata block via Import and Split, Scrivener will create a file containing just the metadata, at the top of import. If you wish to move these values to the compiler, or

[4] https://scrivener.tenderapp.com/help/kb/purchasing-and-installation/installing-the-direct-sale-version-as-a-mac-app-store-customer

[5] Unless you intend to abandon the Markdown approach to writing, you should leave the **Convert Markdown** option disabled.

[6] http://yaml.org/

indeed if you wish to move any set of properly formatting metadata values into the compiler, you can simply copy and paste into the compile pane metadata table. Scrivener will read in the metadata and convert it to the key-value system it uses in the compiler. You can also leave the file in place—Scrivener will use a file called "Metadata" found at the top of the compile group and integrate it with any metadata being supplied by compile settings (detailed in the following section).

Return to chapter ⤴

21.5 Markdown and Scrivener

Scrivener's support of this writing method is built with three different approaches in mind:

1. *Purist*: those that enjoy using the format itself and plan to not make much use of Scrivener's rich text editing features—or intend to use those in an editorial fashion rather than for the purposes of formatting.

   Using this method, you would treat Scrivener like a plain-text editor with benefits. You get things like inline highlights, revision modes, annotations, comments and other affordances uncommon in plain-text editors. That aside, what we are discussing here is the concept of using Scrivener to compose what will ultimately be a plain-text document, and with the Markdown tool integration, the option to automatically process that document into another format, like PDF via LaTeX.

   This is also a good approach to take if you prefer writing or editing in plain-text editors, as you can freely rotate text out and back into the Scrivener editor without fear of losing any formatting.

2. *Hybrid*: those that intend to blend the two approaches. Scrivener's powerful stylesheet system is a natural augment to Markdown style writing. In addition to styles, Scrivener can handle numerous tasks that are otherwise labour intensive: heading hashes by level, footnote markings, table generation, list numbering and image handling.

   If you're curious to see an example of this way of working, the Scrivener user manual's project, available on our support page[7], demonstrates a union of MultiMarkdown and Scrivener styling, used to achieve more than either system yields on its own.

3. *Incidental*: those that really aren't interested in using Markdown at all, but would like to take advantage of some of the high-tech formats it provides. As mentioned earlier, Scrivener itself uses MultiMarkdown internally to generate its ePub 3 files—and if you didn't know that you'd probably never

[7] https://www.literatureandlatte.com/learn-and-support

suspect it was converting formatted WYSIWYG style text to punctuation marks and then back again. Scrivener's compiler sports an exhaustive conversion engine that can even handle tricky types of formatting such as tables.

Rather than keep that magic bottled up in the ePub 3 and KF8 file types, you can switch it on at will in the compiler, by selecting any Markdown-based file type and enabling the **Convert rich text to MultiMarkdown** option in the compile overview's General Options tab (subsection 23.4.3).

This method will primarily only be of use to those not using any Markdown at all in their writings. Any Markdown you use will end up in the output literally, unless you make use of styles to designate sections of raw markup. Read more about this capability with the Compile Format Designer's Styles pane (section 24.5).

When exporting via MultiMarkdown using any of the above methods, most forms of rich text formatting that will be removed, as it passes through a plain-text engine. Highlighting, typographic adjustments, the specifics of paragraph formatting and so forth, have no equivalents in Markdown-based systems. We will focus on precisely what Scrivener *does* address, and then discuss what tools exist to extend what it can address.

For the most part you will be using Scrivener like everyone else does. It is fundamentally a rich text editor, and as such many of the tools you would use to affect varying degrees of a hybrid approach will simply be what everyone else uses. The remainder of this section will address best practices for working in the editor and the rest of the writing interface, with the intention of using one of the previously described systems to compile in the end.

21.5.1 Images

As you could probably see from Figure 21.1, placing an image in the editor (using any of the methods Scrivener provides for doing so (section 15.7)) is a simple way of generating Markdown image syntax when compiling. What you couldn't see from the screenshot is that the compiler also produced the image itself into a folder, such that the reference to "bair_island-wetlands.jpg" would link up and be available for any further post-processing.

Also as with rich text, if you resize the graphic in the editor, instructions will be provided in MMD or Pandoc format to size the image in accordance with the specifications made in the editor. This will change the images display dimensions in its metadata, rather than resampling the image, and provide that dimension data using defined syntax for doing so.

Referencing Images with Document Links

There may be cases where you might *not* want the image to be in the editor, or maybe you prefer using text syntax to refer to images rather than working

Bair Island, San Francisco, California: Fragile wetlands are recovering after restoration.

```
![Bair Island, San Francisco, California: Fragile wetlands are recovering after
restoration.][bair_island-wetlands]

[bair_island-wetlands]: bair_island-wetlands.jpg
```

Figure 21.1: Above the dotted line is what we see while writing. Below the dotted line is what Scrivener converts the image and styled caption line to, upon compilation to a Markdown-based format (caption text coloured for emphasis).

around them as visual objects in your text—but you would still like to take advantage of the automation that produces images on output into a folder. While you could use the image placeholder tag that is available to rich text authors (subsection 15.7.5), you might as well use native Markdown for its extended syntax:

1. Write out the image syntax as you normally would in the text editor.
2. Where you would place the image's path and filename, create a document link (subsection 10.1.1) pointing to that image in the binder.

When compiled, any text in the hyperlink text itself will be altered to the image's file name, so you needn't even be aware of the image's name to use this method, but will need to take care and keep the hyperlink itself limited to that area of the syntax where the image name should be produced, as the entire range of text will be replaced with the image name (Figure 21.2).

Reference Image Example

```
![Bair Island, San Francisco, California: Fragile wetlands are recovering after restoration.](image linked from binder---this text is arbitrary){.left_float}
```

Pandoc Compiled

```
<figure>
   <img src="bair_island-wetlands.jpg" alt="Bair Island, San Francisco,
California: Fragile wetlands are recovering after restoration."
class="left_float" />
   <figcaption>Bair Island, San Francisco, California: Fragile wetlands are
recovering after restoration.</figcaption>
</figure>
```

Figure 21.2: An example of custom Pandoc image syntax (top) being used to generate classed figures in HTML.

Cross-referencing Images

The method Scrivener uses to generate image syntax is conducive to MultiMarkdown cross-referencing.[8] It will use the image filename for the Markdown reference handle (or ID), which in turn can be used as an anchor link. All figures with a generated handle will be placed in a list at the bottom of the compiled document for easy reference, along with any footnotes. Similarly to how MMD itself works, for images to have a generated ID, they must be placed in their own paragraph with no text other than an optional caption.

> **Upgrading from Scrivener 2**
>
> Projects updated from prior versions of Scrivener will need to be adjusted if this form of image cross-referencing was in use. In previous versions, Scrivener included the file extension as part of its handle, but now it only uses the root name of the file. Thus a reference such as `[this](#filename.jpg)` would need to be changed to `[this](#filename)` in order to continue working. The RegEx `\(#(.*?)\.\w{3}\)` replaced with `(#$1)` might suffice to help convert these older references.

For example, if you drag a graphic called 'analysis_of_derivatives-2008.png' into the editor, Scrivener will export that graphic into the compile folder (subsection 21.6.1) and generate syntax where the position was located, like so:

```
![][analysis_of_derivatives-2008]

[analysis_of_derivatives-2008]: analysis_of_derivatives-2008.png
```

[8] At the time of this writing, Pandoc does not have an automatic method for cross–referencing images. You would need to use the referenced image technique described in the previous section to create the necessary Pandoc syntax to create an HTML ID that can be linked to.

You can thus refer to this figure in another portion of the document with the standard MMD syntax:

```
[see chart](#analysis_of_derivatives-2008)
```

Need to get the name of your image for cross-referencing?

Double-click on inline images in the editor to gain access to its name. For fully embedded images you can even edit its handle right in the window, as well as copy it for later pasting. Images linked to files on the disk will print their full path. All you will need to copy is the name itself sans file extension. For images linked to the binder, click the Reveal in Binder button, where you can of course copy the name from.

Captions

When images fall upon their own line, you can supply a caption to them using one of the following techniques:

— Place the caption within straight (not typographic) double-quotes, or in matched square brackets, on the same line after the image, separating the image from the caption text with a space.

— Alternatively, place the square bracketed caption on the line *directly* following or preceding the graphic.

— Use a designed caption style on the line directly preceding or following the image. Refer to MultiMarkdown and Pandoc Options (section 24.14) for further information on mapping styles to functions.

Captions can be used with both images and tables, though with tables, only styled ranged will be used for captions, bracketed lines will not be recognised.

21.5.2 Lists and Tables

This is an optional compile behaviour, set with the **Convert tables and lists to MultiMarkdown** setting in the compile options for Markdown-based formats (section 23.4.3), or when the broader **Convert rich text to MultiMarkdown** option, in that same area, is applied. If you use tables or lists at all in your document, you should probably enable this option, as the plain-text output of these two constructs by themselves will not produce results in line with their original usage in the editor.

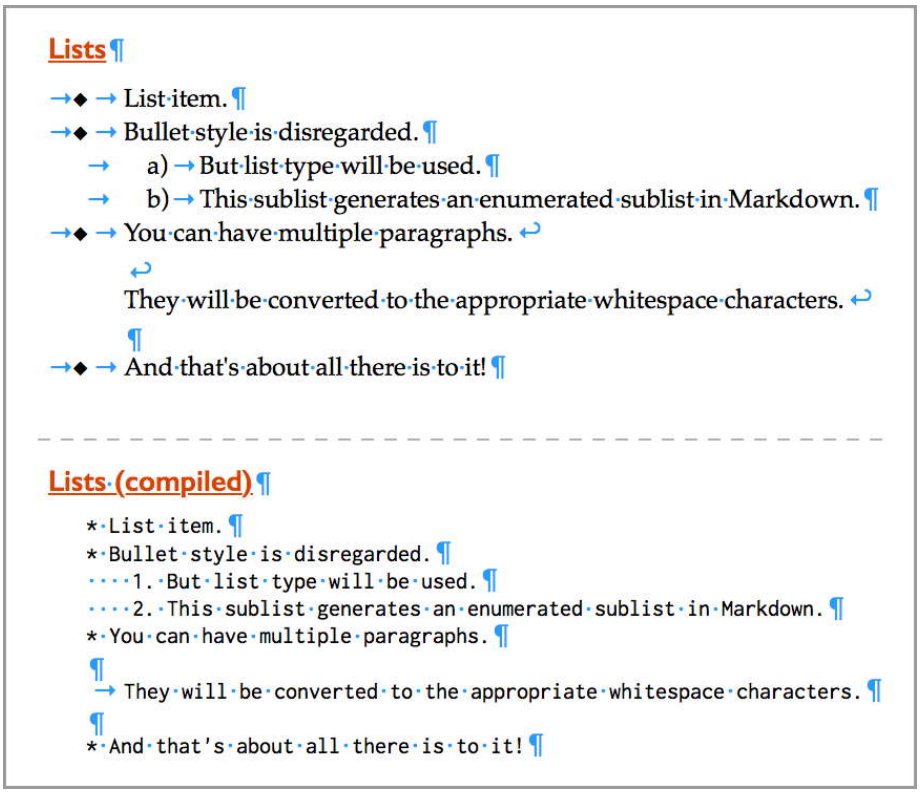


Figure 21.3: Example list formatting converted to Markdown text (invisible characters shown for clarity).

List Support

The compiler can generate Markdown style bullet and enumeration lists based on lists found in the text, when the option is applied to do so.

— All list types found in Scrivener will be expressed as either bullets (using asterisks) or enumeration (using digit + dot formation). For example, a list type in Scrivener using "a. b. c." formatting will generate "1. 2. 3." on output.

— You can mix list types per level. A list can be bullets on level one, enumeration on level two and back to bullets on level three.

— Multi-paragraph list items are allowed, however you will have to use a different whitespace approach in Scrivener, since its own list formatting feature does not allow for paragraph breaks or tabs (the way one would do this using the basic syntax directly). Use Insert ▸ Break ▸ Line Break (⌥⌘Return) to insert new lines within a single bullet line. Scrivener will handle the whitespace conversion for you.

  You can insert multiple line breaks to space out the paragraphs in the editor aesthetically—Scrivener will clean them up for you.

Table Support

Table conversion is relatively simple, and like most aspects of conversion to Markdown, disregards the particulars of formatting and converts the basic structural data expressed by the table. For example, in Figure 21.4, we see a table with a grey fill being used to indicate the header row. It is a header row merely by being the first row, and requires no styling to become that way in the output, nor will the styling have any bearing. There are three exceptions where table formatting can be converted to table syntax:

1. Cell alignment can be expressed. The first row establishes the alignment for an entire column. Cocoa tables can align each cell differently, but not pipe style Markdown tables.
2. Text styled as a caption found on the line directly preceding or following the table, will be converted to the proper syntax for a table caption (Figure 21.5).
3. Merging cells within a row will use the appropriate syntax for doing so in the output.

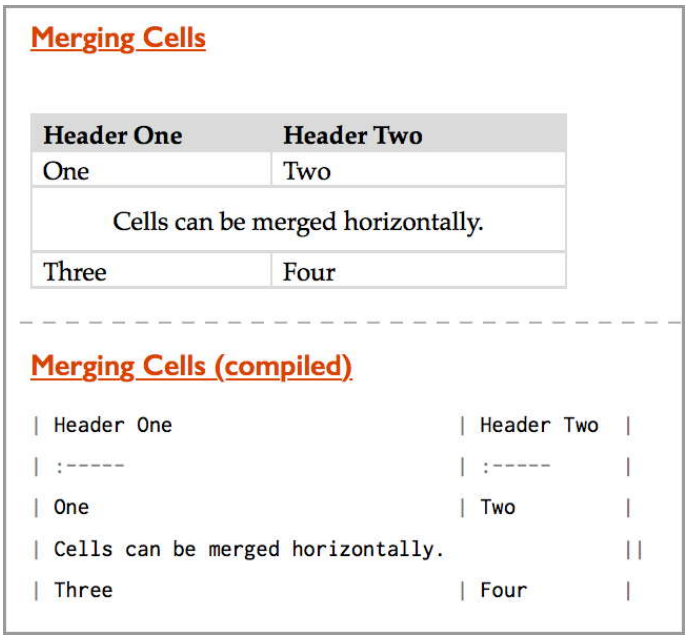


Figure 21.4: Table conversion supports merging cells (plain-text formatting optimised for clarity).

21.5.3 Footnotes

You needn't worry about footnote syntax and identification when using Scrivener to compose MMD or Pandoc Markdown documents. When using the built-in tools for handling notes (chapter 18), the compiler will automatically generate

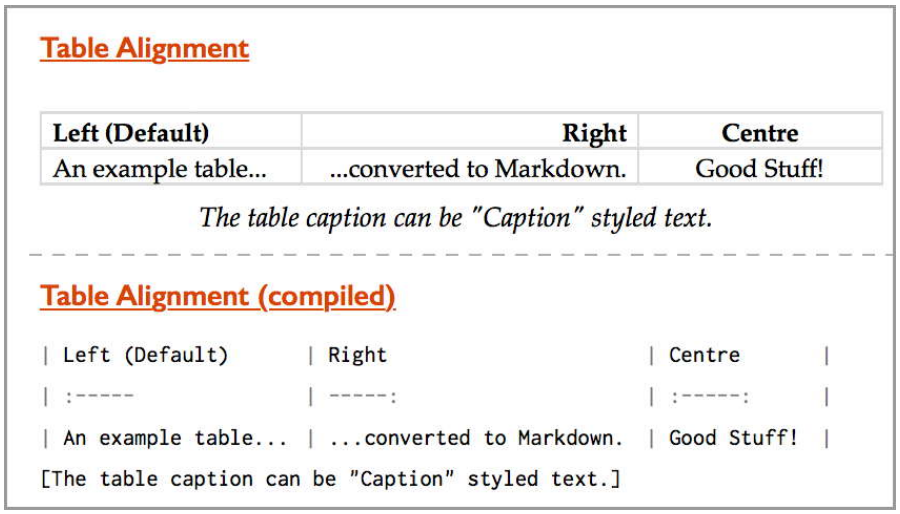


Table Alignment

Left (Default)	Right	Centre
An example table...	...converted to Markdown.	Good Stuff!

The table caption can be "Caption" styled text.

Table Alignment (compiled)

```
| Left (Default)      | Right                     | Centre      |
| :-----              | -----:                    | :-----:     |
| An example table... | ...converted to Markdown. | Good Stuff! |
[The table caption can be "Caption" styled text.]
```

Figure 21.5: Alignment works by column, not by cell, with the first row establishing alignment (plain-text formatting optimised for clarity).

sequenced identifiers for you and place the markers and references where they should go.

Whether they will ultimately become footnotes or endnotes is not dependent upon anything Scrivener can do, but depending on whether you use inline or linked footnotes, the compiler will use two different naming conventions for the handles:

```
[^fn1]: This came from an inline footnote.
[^cf1]: This came from a linked footnote.
```

This distinction, if going straight to one of the basic output formats, will be of no concern to you. However, if you intend to post-process the results using your own scripts or stylesheets, having two naming schemes will enable you to handle these as separate streams of notes in your document. You could for instance use inline footnotes as footnotes, and linked as endnotes.

The placement of the marker in the source text will follow the same rules as standard Scrivener usage. Mentally replace the entire inline footnote with the marker, to see where it will be placed, or the trailing right-edge of a linked range.

21.5.4 General Styled Text Support

Beyond the types of formatting Scrivener supports for conversion, there is also a freeform capability that opens up the realm of possibilities far beyond what we could anticipate ourselves. Scrivener's stylesheet system, the usage of which is documented in the Styles and Stylesheets (section 15.6) section, was designed not only for rich text use, but specifically for plain-text writing as well. As a semantic system for tagging text in the editor, it forms a natural coupling with formats like Markdown that are rooted in these concepts, and can considerably extend their

reach—either to address extended syntax we do not ourselves provide interfaces for, or to go even beyond what these systems define (such as the injection of raw LaTeX code directly into the output).

Before you build a custom style...

Some styles can be handled internally to generate syntax for you (block quotes, code blocks & spans and captions). Compile-time styles can be assigned in the MultiMarkdown and Pandoc Options compile format pane (section 24.14).

Beyond the built-in style methods, the Styles compile format pane (section 24.5) provides an interface for wrapping styled ranges with a prefix or suffix—both around the entire range and per individual paragraph within that range. If you would like to see a simple example of this in action:

1. Bring up the Compile interface with File ▸ Compile...
2. With **Compile for**: "MultiMarkdown" chosen at the top of the window, right-click on the "Basic MultiMarkdown" format in the left sidebar.
3. Select "Duplicate & Edit Format..."
4. Click on the Styles format pane.

You will find "Addition", "Deletion" and "Highlight" defined (among others, used in the aforementioned option pane) which are set up to produce CriticMarkup syntax around marked ranges. If you select some text in the editor and mark it as "Deletion", then the text will be wrapped with a prefix and suffix: `{--This is the deleted text.--}`. The default MultiMarkdown LaTeX boilerplate will style these markings nicely.

The **Paragraph prefix/suffix** fields will insert the text on each line within the selected style range. For example you could insert ": " in front of each line for a "Definition" style.

The user manual you are reading makes heavy use of this technique, and thus serves as a practical demonstration. Download a copy of the project (and its compile settings) from our web site[9].

[9] https://www.literatureandlatte.com/learn-and-support/user-guides

These are not word processing styles

If you are using a Markdown-based approach to create a word processing file via .docx, .odt or similar, do not expect the styles you use in Scrivener to translate somehow through to the final output. Keep in mind that what Scrivener compiles in these workflows is a plain-text file. If the format or converter you are using supports a syntax for conveying text tagged as styled then you would be using Scrivener to generate that syntax—potentially through its stylesheet system, but always in concert with the conversion tool's own native capabilities.

21.5.5 Heading Styles

When the **Convert rich text to MultiMarkdown** option is enabled , in the General Options tab of the compile overview screen (subsection 23.4.3), it is possible for heading styles that you within the text to be converted to Markdown-style hashmark headings. In most cases you would want to use the compiler's ability to generate Markdown-style headings with Section Layouts, but for cases where smaller documents need to be further subdivided without creating additional outline hierarchy, using styled headings can pick up where they leave off.

The important ingredient to be aware of is that each level of heading style you use should have saved within it an HTML header level. This is saved into the style like any other form of paragraph formatting, and can be set with the Format ▸ Paragraph ▸ HTML Header Level submenu.

Thus a paragraph style with the "H3" header level applied to it will compile as:

```
### Text Assigned to the Style ###
```

Some Markdown engines require a clear line of space around a heading line, such as Pandoc. That detail will be left up to you and how you compose the text in the editor.

If you require more direct control over how the syntax is applied to a heading style, you should not use the HTML Header Level setting, and instead use the prefix/suffix style tools described in the previous section.

21.5.6 Hyperlinks and Cross-References

This is an optional compile behaviour, set with the **Convert rich text to MultiMarkdown** setting, in the compile options for Markdown-based formats (section 23.4.3). With it, regular hyperlinks to URLs will be converted to equivalent Markdown links, and internal document links to other titled items in the draft will use MultiMarkdown/Pandoc style cross-referencing (Figure 21.6).

Figure 21.6: Links can be optionally converted to Markdown syntax.

Keeping Cross-References in Parity

One of the things that Scrivener can help you with that otherwise requires manual labour with MultiMarkdown and Pandoc is keeping cross-references to headings correct, even if heading names change. For example if you change the name of a heading from "Browsing the Web with Lynx" to "Browing the Web from the Command Line", you'd have to go back and fix every case in the text where you used [Browsing the Web with Lynx]. With Scrivener, consider using its built-in hyperlink feature to link the text of the link within the brackets to the item it refers to in the binder. Not only do you get a handy clickable link much like your readers will, but if you change the title you can make use of the ability to update link text (section 10.1.2).

21.5.7 Annotations and Comments

For most uses, inline annotations and linked comments will serve much the same purpose when using MMD as they would otherwise. They are a convenient way for you to apply notes to your document, or to share and receive thoughts from those who are editing your work.

If you wish to insert these comments into your compiled document, Markdown itself does not have a convention for inserting comments into a document, but there are a few approaches you can take, using the Annotations compile format pane (subsection 24.19.7).

Upgrading from Scrivener 2

In previous versions of Scrivener, inline annotations were—given their ability to be prefixed and suffixed by text—a convenient way to insert HTML comments (and thus via MultiMarkdown, raw LaTeX code) and other codes around marked text. Consider using styles in Scrivener 3. There are more prefix/suffix options available in the Styles compile format pane (section 24.5), and the type of code they insert around the marked text can differ per compile format, as well as per style, and styled text can even be set to be omitted on a per format basis. All in all there are few reasons to use comments and annotations for anything but comments and annotations, these days.

21.5.8 Preserve Formatting

In the past, the Format ▸ Preserve Formatting feature would have been used to generate code spans and code blocks with Markdown-based output. This capability has been removed, as it is now served by styles. Use the "Code Block" and "Code Span" styles provided in the stock set, or if you create your own, set them up in your compile settings using the MultiMarkdown and Pandoc Options (section 24.14) compile format pane.

The **Treat "Preserve Formatting" as raw markup** option, used in conjunction with the **Convert rich text to MultiMarkdown** setting, both in the compile options for Markdown-based formats (section 23.4.3), will cause marked text to pass through the compiler untouched. Ordinarily the conversion option will protect punctuation marks that are used by Markdown. This option will mainly be of interest to those that use the iOS version of Scrivener, with its equivalent capability. If you are only using a Mac or PC to compile, you might want to use a dedicated style instead (with the **Treat as raw markup** option enabled for it in the Styles compile format pane).

Return to chapter ⤴

21.6 Compiling

This section of the documentation will not cover compilation in detail. You should consult the chapter on Compiling the Draft (chapter 23) for detailed information on the various options that will pertain to you. Most of the MMD options are built into the existing option panes, often replacing those that are not relevant, or changing options as necessary. This section will instead focus on the various formats available, and which compile panes to check for best harnessing Scrivener's compile automation.

21.6.1 Compile Folder

As with the standard HTML output in Scrivener, if the compile results in more than one file being produced, a compile *folder* will be created using the name you specified, with the output document located within that folder along with any support files it needs. This will most often be graphics, or supporting .tex documents for typesetting. The folder name will match the output filename, such as `novel.md`, or `novel.tex`.

You might assemble CSS files, custom .tex files and other bits of supporting data into this folder so that you can compile directly into a finished location, rather than compiling and moving things around into the final support folder each time. Scrivener will handle this gracefully for the following cases:

— If a folder using the name of the compiled file exists in the target compile folder already, then Scrivener will reuse the subfolder.

— If the target compile folder itself is using the name that resembles a compile folder[10], a subfolder will *not* be created, and Scrivener will use the target folder itself to work within.

For example, is called "novel.tex", then when you compile to that folder again, Scrivener will leave what is already there alone, *unless* it is a file that Scrivener is scheduled to produce. This way compiled graphics will be updated as well as content files, but any supporting elements you've placed into the folder yourself will be preserved.

For further control you can instead rename the compile folder to use the suffix `_mmd` or `-mmd` and Scrivener will leave it alone the next time around.

[10] Any folder ending with ".tex", ".md", ".mmd", ".fodt" or ".html" will be recognised as a previously used compile folder.

The "Overwrite preserves other existing files" Checkbox

In the compile file save dialogue box which appears after clicking the Compile button—for MMD formats which have the potential to produce folders instead of singular files—you will find a checkbox labelled "Overwrite preserves other existing files". This option accomplishes the same as above with one slightly different variation in that you use it to compile to the parent folder containing the compile folder, not to the file *within* the compile folder. Thus, if your "my_novel.tex" folder is in Documents, you would compile to Documents, opting to overwrite the folder. With the checkbox on, anything in that folder not otherwise slated to be produced by Scrivener will be preserved. With the checkbox disabled, the folder will be completely refreshed.

Users of the **Mac App Store** version will need to use this feature exclusively, as Sandboxing does not allow the dynamic folder preservation technique.

21.6.2 Plain MultiMarkdown

While the built-in compile options that go directly to a specific format are incredibly handy, there are occasions when exporting a plain MMD document can be useful and even valuable. The Markdown text format makes for an excellent long-term archival document. It is compact, easy to read, and easy to transform into a variety of formats. It's also very portable, being plain-text. While dropping an archived Scrivener project into a backup system is something you should always do at the conclusion of a project, a plain-text copy of the final product is something that will be more easily archived in multiple places for many years to come.

Beyond archival, a plain MMD document allows you to "step into" the compile process a bit, which can be useful if you intend to do advanced post-processing of your own, or are getting errors with a particular format and need to figure out what is causing the syntax to foul up. The "MultiMarkdown" selection in the Compile For menu is precisely the same copy that Scrivener will be using to feed to the MultiMarkdown or Pandoc conversion engines, when generated a direct formatted output, and so is a critical troubleshooting step when faced with difficulties.

Post-Processing and Custom Formats

One big advantage of the plain "MultiMarkdown" approach[11] is its post-processing option. This won't be something you find in the compile overview screen, but instead when modifying or creating your own compile formats. Refer to the Processing compile format pane (section 24.22) for a full run-down on how to use its various options specifically.

— If you wish to adjust how Pandoc or MultiMarkdown output syntax to target file types—such as for example how figures or tables are put together at the code level, then this is how you would hook up the necessary scripts for doing so (using whatever technology you prefer).

— A pretty decent approach to creating your own virtual file types is to use XML processing in your favourite scripting language to modify the HTML5 output produced by MultiMarkdown. You can create your own file types from scratch in this way, such as BBCode, WikiMedia and so on.

— You can use your own preferred markup language in Scrivener and then make use of whatever post-processing tools are necessary—i.e. not using Markdown-based syntax at all. One could write using ReStructuredText, ASCIIDoc or whatever they prefer, in Scrivener, and set up the command-line environment to produce file types from these formats.

The Processing pane is also available to the Plain Text compile file type. If you're looking to generate output that isn't Markdown-based, you may find that a better starting point as you will have full control over all output syntax. Refer to the documentation on the Markup compile format pane (section 24.10) for further information.

21.6.3 LaTeX

MultiMarkdown's LaTeX support is premised by the concept of using boilerplate .tex files, rather than describing the document as a single large .tex file. This leaves the base document, the part you've written and that Scrivener will be compiling, clean and focussed on the content. Ordinarily you would need to install and manage these boilerplate files yourself, but Scrivener offers the ability to manage these details for you. It will set the MultiMarkdown metadata up and handle the gathering of any necessary support files into the compile folder for one-click typesetting. In fact, you may not have to touch a single setting to get a ready-to-typeset document (Memoir Book by default).

To view the metadata keys that Scrivener will be inserting:

[11] We call the compile file type "MultiMarkdown" for simplicity, but it's also capable of producing Pandoc format, and as you will shortly see from this section, literally anything that can be constructed out of plain–text.

1. Open File ▸ Compile... and select **Compile for**: "MultiMarkdown ▸ LaTeX" at the top.
2. Click on the Metadata options tab on the right hand side of the compile overview screen (subsection 23.4.2).
3. Click the Preview button along the bottom of the metadata configuration area.

The set of document class boilerplates will be referred to by a keyword. For example, if the "LaTeX Config" metadata key is set to "memoir-book", then all of the necessary .tex files to produce a Memoir Book style PDF will be assembled for you when you compile.

If you already have MultiMarkdown's LaTeX support files installed into your system's texmf folder,[12] then Scrivener will not duplicate them into the compile folder, and all you will see is the .tex file it produces.

A number of stock compile Formats have been provided with Scrivener, that make it easy to switch from one document class to another (section D.6). If you would like to make this change to an existing format that you are working on, use the LaTeX Options compile format pane (section 24.12).

In addition to the default classes, you can write your own in such a way that they are saved into the compile settings (or ultimately a preset) by selecting "Custom" in the LaTeX Options compile format pane and writing your boilerplate preamble and footer into the provided tabs. The provided "Modern (Custom LaTeX)" compile format is an example of just such an approach.

21.6.4 HTML

Produces a clean, semantic HTML5 document. If you are blogging or using a CMS to publish your work online, this is a great option to use (if the system doesn't take Markdown itself) as the result makes no assumptions of formatting, allowing the site stylesheets to handle that part of the job.

If you do wish to supply style to your HTML output, you can set the appropriate metadata keys for doing so in either the Metadata compile format pane, as part of the format's settings (section 23.4.2), or to the project specifically in the compile overview's metadata options areas (section 13.5). Consult Pandoc and MultiMarkdown documentation for how to supply CSS, either in the metadata itself or as included files.

[12] If you have installed MultiMarkdown stand–alone[13], then by default this will have been done for you.

21.6.5 Flat XML (.fodt)

This format can be opened by LibreOffice, and from there converted into any other word processor format you desire. For most word processing workflows, you will be better served by the OpenOffice (ODT) format for MultiMarkdown, or Pandoc's MS Word (DOCX) format. The Flat XML file approach provides a better route for those looking to further automate the output with post-processing.

21.6.6 PDF (via LaTeX)

⟨Direct-sale only⟩ The PDF output choice utilises the LaTeX typesetting engine, and simply offers a streamlined way of going from Scrivener to a printable PDF in a single step. It is fundamentally identical to compiling as a .tex document, and then opening that document in a TeX editor and typesetting it with no alterations. This method requires a full .tex document to be compiled, not a partial, or "snippet" content-only output (subsection 21.6.7).

If the document has no extra requirements, and typesets cleanly otherwise, you will find it to be a most convenient way of distributing and archiving great looking printable documents. When troubleshooting or performing initial typesetting it will often be beneficial to enable the **Show PDF Log** setting in the General Options area of compile overview (section 23.4.3). This will produce the full .tex log file, and so reveal any warnings that might otherwise be ignored.

Since this method combines a full compile with full MMD post-processing and three *pdflatex* executions, large documents may take a while to fully complete.

> **Where is it?**
> If you do not have a LaTeX distribution with *pdflatex* installed on your computer in an executable path, then this option will not appear in the compile format dropdown. Scrivener looks for pdflatex in `/Library/TeX/texbin` and `/usr/texbin`, and if it isn't found in either of those places, it will use the *which* command-line tool to try and locate a copy in the executable path. If nothing is found in any of those places, the file type will be removed as a compile option.

21.6.7 Exporting in Snippet Mode

MultiMarkdown offers the possibility of compiling what is referred to as a "snippet", as opposed to a full document. A full HTML document is one that includes a header and body tags around the content you compile; a full LaTeX document is ready to typeset. Snippets on the other hand only contain the *content* and as such are useless all by themselves. They are however useful for post-processing and pasting (or in the case of LaTeX, including) into pre-existing boilerplates.

To cause the compiler to create a snippet instead of a full document, remove all forms of metadata from the Metadata Options area of the compile overview screen (section 13.5) that would be used to classify a document—such as "Title"

and "Author". Refer to the MultiMarkdown documentation for full details on which metadata fields trigger full document mode.

Return to chapter ⤴

21.7 MMD & Pandoc Metadata

Both MultiMarkdown and Pandoc support free-form metadata for document classification, with some metadata being used functionally by them. You should read their respective documentation about these fields. Beyond that you are free to add your own without detrimental effect. There are three possible sources of metadata in Scrivener, all of which will be combined into one metadata block at the top of the compiled output:

1. Project specific: this is where you would set such things as the author name, copyright information and so forth. To set up project-level metadata, which will be attached to every compile you produce, use the Metadata Options tab of the compile overview screen (section 23.4.3).
2. Compile group specific: if you place a document called "Metadata" at the very top of the selected documents you will be compiling, Scrivener will append the text in that file to the metadata block. In this way you can provide information specific to the compile group—for example if you are publishing multiple articles out of a single project, the "Title" metadata field would make a good candidate for this method. It is up to you to format the metadata rows properly as text. This file can be named "Meta-data" or "Meta data" as well.

   The contents of this file should be properly formed for the system you are using. With Pandoc you should use YAML[14] formatting.
3. Compile format: the chosen "look" of a document, or the format, is where you would typically make such declarations as stylesheets, document classes and so forth—depending on the target file type. This is done in the Metadata compile format pane (section 23.4.2).

In Figure 21.7 we can see an example using all three methods described here. The green text is metadata coming from the project's compile settings. The author's name and copyright declarations are made here. Below that, in purple text, we have formatting decisions made by the chosen compile format. In this case we can imagine Ralph is using HTML, and has chosen a favicon and stylesheet for publishing documents of this style with. They might apply these settings to multiple projects over the years. Lastly, in blue, we have information being supplied by the compiled documents themselves—namely the "Metadata" file in

[14] http://yaml.org/

Figure 21.7: Metadata can come from three sources, colour coded for your convenience.

their draft or front matter folder. This might change from one day to the next, depending on which article they are compiling.

The only requirement for the metadata document is that it be the very first document in the list of things to processed during compile, you can use various features of the Contents tab to set the document. A few examples include:

— Selecting a subfolder of the draft to compile, and ensuring each such folder has a "Metadata" file at the top.

— Using Collections as your compile group (where the metadata file could be added to the top of the Collection).

— Using the front matter feature to swap out metadata sets.

— Setting filters to selectively use the appropriate metadata file for the content being delivered.

Return to chapter ⤴

Part IV

Final Phases

> Nothing stinks like a pile of unpublished writing.
>
> **Sylvia Plath**

Distilling your work into a final product is an essential task for any writing application. Scrivener approaches this problem from multiple fronts, giving you plenty of options for producing a manuscript, web pages, printouts, eBooks, and quite a bit more. Most of these methods are functions of the compiler, a powerful export feature which will take the contents of your draft folder and produce a single document from the many pieces that comprise it.

At a basic level this is done by choosing a basic look, and then selecting from different template layouts to represent the types of documents in your draft (should folders have numbered chapter headings, or just print the folder name—should files print a title or have scene separators between them, etc.). At its more complex, you can design your own formats, create new layouts and even create your own file types (going beyond .docx and the many types we already supply, that is).

If you are accustomed to working in a word processor, you might want to start with the "default" format, letting the formatting you do in your editor as you write be the basis for how a document will look, leaving the compiler to mainly just sew up all of the files into one document. If you are used to working in plain-text, or another workflow that does not regard formatting as part of writing, then you might be interested in trying one of the presets that formats your work for you, such as the submission manuscript format. You might even try your hand at book design, getting a good start on the project with one of Scrivener's eBook formats.

The compiler can also be a tool for producing specialised reports, by selecting only portions of each item in the Draft to be included, such as just the title and its synopsis and metadata. The enumerated outline preset demonstrates one such strategy, where an indented outline of all your draft titles will be exported as a file.

Many authors will be taking their finished drafts to a word processor, desktop publishing, or scriptwriting application for final post-production work. We will discuss several common applications on the market and how best to work with them in Scrivener.

Finally, we will also discuss more traditional methods of printing and exporting, as well as a few techniques you can use to add final polish to your manuscript. If you're proofing, sometimes the easiest thing to do is simply print the file you want to take a red pen to.

Creating a Table of Contents

22

With the notable exception of eBooks, there is no support for inserting a dynamic table of contents into your draft—that is, one that will update itself as you change the names of items in your draft or move them around—but there is a way to produce a static list of items that is cross-referenced to page numbers. This feature is mainly useful in conjunction with the PDF/Printing workflow and with the RTF-based formats when compiled and opened in a word processor that supports bookmarks and cross-references.

PDFs are more portable than word processor files, since page numbers are baked into the file and do not rely on dynamic features that only some word processors support. If you intend to distribute the file to a number of people, and are unsure of what word processor everyone uses, PDF will provide the most consistent result between all platforms.

22.1 How to Create a ToC

Creating a table of contents is a simple process, but because it is a static list, you will probably want to save it for one of your final steps before compiling, as any changes in outline order or the addition or removal of sections will not be reflected in the list:

1. Select all of the items that you wish to have included in the ToC. It may be easiest to do this in outliner view mode, so you can collapse and expand the various containers until the outliner looks roughly like what you want the ToC to look like (speaking from a standpoint of content).
2. No matter which view you are working from, once all of the appropriate items are selected together, use the Edit ▸ Copy Special ▸ Copy Documents as ToC menu command.
3. Create, if necessary, an empty file in the draft folder and paste the ToC list into this document. You may now format the result as you wish.

Creating Sectional Contents

If you need to create a mini-ToC in the preface for each part of a book, such as is the ones you see in this very manual for larger chapters, you can follow the above instructions to produce a smaller scale list of sections. You will just want to select the relevant section instead of the entire draft and paste the ToC copy into the preface area for each part.

The resulting list will be formatted with the name of the section on the left and a special tag on the right, with an amount of indenting applied to each line in accordance with its relative outline depth. Both the tag and the name will be hyperlinked back to the item they refer to in the draft.

The <$p> placeholder, will be replaced with a page number reference in the compile process. The title will also be examined and updated to match the visible name of the sections as they will be compiled. For example, if the compile format you are using replaces binder titles with the word "Chapter" followed by a sequential number, then the Table of Contents link text will be changed to match them

You also can create your own table of contents by hand. There is nothing special about the "Copy Documents as ToC" command that cannot be replicated manually. So if you do not like the default look, you can either adjust the formatting after pasting or generate your own ToC from scratch by using the <$p> placeholder and linking it to the section you wish to reference with a document link. The title should be linked as well, if you wish it to acquire any prefix or suffix information. Read Linking Documents Together (section 10.1) for more information on that process.

22.1.1 Things to Watch Out For

— When first opening the compiled document in Microsoft Word, you will need to generate the ToC numbers by running a test print preview once; they will appear as question marks until you have done so.

— This feature requires word processors capable of understanding bookmarks and cross-references. Page numbers may appear as question marks in the list if the software do not.

— If you are not using title generation in compile, and are instead relying on a formatted title within the draft text itself, you may find the "Copy Documents as ToC" feature less useful and might wish to create your own from scratch.

— If some items come out with no page number, check to make sure those items are included in the compile settings, and are set to output text. If the preview area in the Formatting pane type row that corresponds with that item type is empty, then there will be nothing to link to.

— The dot fill used between the title and the page number is an macOS text engine underlining feature which may not be visible in all word processors.

22.2 Contents in eBooks

The ePub and Kindle compile export formats have an automatic built-in table of contents generator which should be used instead of the methods described in this section. If you are publishing to an eBook platform and wish to set up a ToC, please read about the Table of Contents compile settings tab (subsection 23.4.6).

In cases where that solution does not produce the desired results (perhaps you use section breaks for reasons other than purely to create new sections), you can

create custom (albeit static) ToC for your eBook. You would follow the previously described procedure for generating a static ToC, only using the Edit/Copy Special/Copy Documents as Structured Link List instead. This will also tab indent by hierarchy, but will omit the page numbering placeholders, which are largely irrelevant in an eBook environment.

Properly Formatting an ePub3/KF8 ToC

The guidelines for creating a valid ePub 3 book are strict, and as such one needs to follow these rules when constructing a custom ToC page within Scrivener: (a) the section must have a heading that is styled using one of the numbered heading styles, such as "Heading 2" (either in the text itself, or generated by the Layout the section uses in compile), and (b) the rest of the content must consist exclusively of links pointing to other sections of the eBook, one link per line. Indent levels for each line will be scanned, and produce a "folder" style navigation in some devices, where one can drill down from "Part I" to "Chapter 23" to "Section 23.8". Deviating from these guidelines may result in a book that does not validate.

To merely move the compiler's automatically generated ToC from one location to another in the eBook, use the `<$toc>` placeholder tag in an otherwise empty document in the draft folder. Remember that many people get to this via a menu in their eBook reading software or devices, so where it appears is often not important. Placing the ToC at the end of the book is a common tactic as it keeps it out of the way, but is discouraged by Amazon as navigating the reader to the end of the book can disrupt the cloud sync system which keeps multiple devices updated to the last point in the book one has read to.

See Also...

- Compiling the Draft (chapter 23)
- Linking Documents Together (section 10.1): the crucial glue used to make contents interactive, so readers can jump straight to the listed sections.
- Table of Contents (subsection 23.4.6): a special compile tab used for setting up eBooks.

Compiling the Draft

23

In This Section...

The main purpose of Scrivener is to provide a place that will help you write a long work—whether that be a novel, thesis, screenplay, non-fiction book or a

series of technical manuals. This work will nearly always be structured as individual pieces in the binder, sometimes a great many, but no matter how many, they can easily be saved as one large file for working with "the rest of the world".[1]

It all begins with the File ▸ Compile... (⌥⌘E) menu command.

At its most basic, the compiler takes the contents of the "Draft" folder, formats it as you specify, and outputs (or prints) it as a single document. Using the various settings available, you can export or print your texts however you like—even regardless of how the files are formatted in Scrivener itself. Considering the many types of things one might use Scrivener to for, we support a diverse array of workflows, from as basic as printing paper out of your printer to linking Scrivener together with established programmable interfaces for automated post-processing.[2]

Compile settings are an intrinsic part of your project; they're just as much a part of it as any folder or file in your binder. They will be saved with the project and travel with it if you move from one computer to another or share it with a collaborator.

Going deeper, you can create your own formats, which is a way of determining which fonts are used and myriad other choices, save those formats into projects or share them online with others—and of course import formats other have designed.

I don't need all of this; just print my words!

Aren't too worried about the specifics and just want to get all of the text you wrote out into a single file? You'll probably find that most of our built-in templates work out of the box without adjustment beyond filling in your name and giving the book a title and maybe a cover. And if even that is too much, you can just click the "Default" format in the left sidebar of the compile window, leave the **Compile for** setting to "Print" or maybe "PDF" and click the Compile button in the lower right corner.

23.1 Compile Overview Screen

For most projects you will rarely need to go any deeper than what you can do with the compile overview screen. It has everything you need to change the file type—such as word processing files for programs like Word and LibreOffice, Web, ePub and more (section 23.5)—the basic look of the document, how those looks will be applied to your work, and then a number of options for how the file should be

[1] If you're looking for a way of exporting individual files from your binder rather than turning them into one document, check out the documentation on Exporting (chapter 25).

[2] And if none of that made sense, don't worry about it. If it did make sense, rejoice: you can use Scrivener to create XML files using your own hand–brewed schema and pipe them to a server automatically if you really want to.

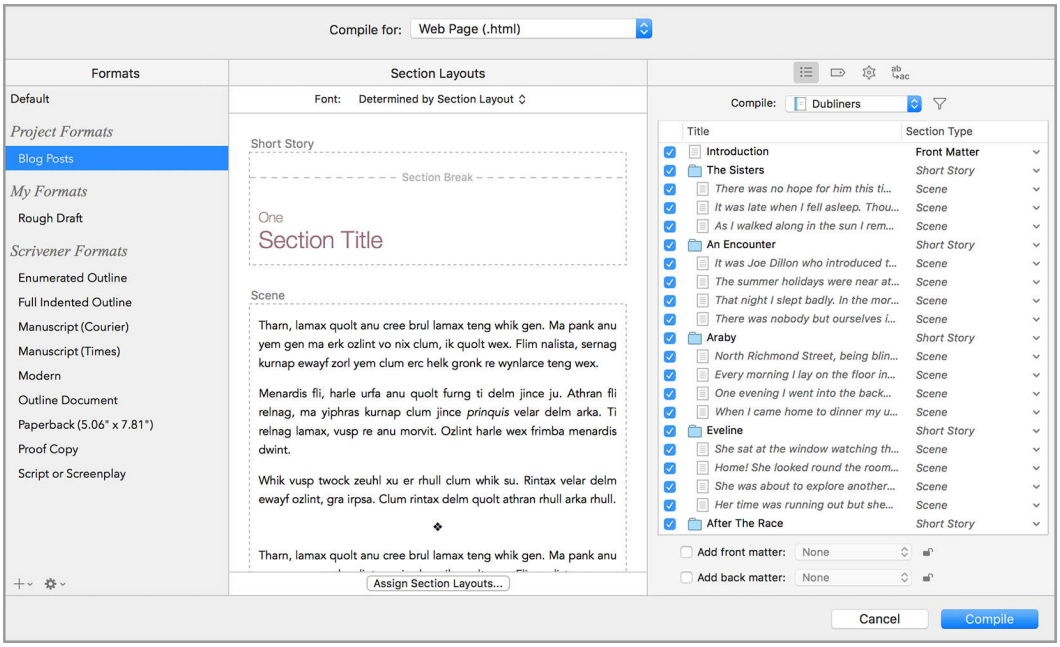


Figure 23.1: A set of short stories being prepared for compilation to a blogging site.

created, from the name of the work to whether or not italics should be printed as underscored text.

The compile overview screen is split into three columns (Figure 23.1), meant to be used in a left to right fashion:

1. Compile Formats: listed in the left sidebar, the availability of these are determined by the type of file you select with the **Compile for...** dropdown at the very top of the overview screen.

2. Section Layouts: the middle column displays a loose preview of the appearance of your document, based on either the template you're using or the choices you have made in assigning section layouts to the types of documents in your binder. For example if you have "Short Story" and "Scene" documents in your draft (as shown in the figure), you might see a chapter break example followed by an example of what a scene will look like. If you change the compile format in the left pane, you may see different looking preview tiles.

3. Compile Options: on the right hand side is a tabbed option view. Chiefly, and visible in the figure, is the Contents list, where you can get overview of what will be included in the output, make large-scale adjustments to that (through the use of filters and tacking on front or back matter) and as well you can change individual document types on the fly if you spot a mistake.

Let's now take at a look at each of these columns in greater detail.

Upgrading from Scrivener 2

You might be asking at this point, where are my project's compile settings? We have done what we can to make the transition as smooth as possible, but it would be impossible for us to provide a completely seamless conversion, given how different the new compiler is from the old. If you are a facing a rapidly approaching deadline, it might be a better idea to hold off on using v3 initially, until you have some room to breathe and can explore the new system without external pressure. But if you do want to dive in, or have the time to learn it, a good way to get started will be by importing your old project's compile settings as a new format (section E.2).

Return to chapter ⤴

23.2 Compile Formats

The left sidebar contains a list of formats applicable to the current file type you have chosen along the top of the compile overview screen. For example, if you select "Plain Text (.txt)", then you will only see those few formats that are useful to this file type (without a concept of formatting or fonts, the "Modern" format paired with a .txt file would be meaningless). This list is broken up into three sections—not all of which may be visible:

1. *Project Formats*: compile formats that have been saved into the specific project you are working with. Some built-in templates come with their own project templates set up for you. If you'd like to make those available to other projects, refer to Project vs My Formats (subsection 24.1.1).
2. *My Formats*: any formats you have installed or created that are saved to the computer. All of your projects can make use of these formats, and so long as they are configured for the file type you are using, they will always be shown. Any format here can also be tucked away into a single project if you do not wish it to be global.
3. *Scrivener Formats*: we ship a few example formats that you are free to use for your own work, or as starting points for your own designs. They have all been built especially to work with our built-in project templates and the stylesheets used in new projects. These formats cannot be deleted or modified directly (you can duplicate and refine them for your own purposes of course).

23.2.1 Selecting a Format

Applying a format to your project may be as simple as clicking on it in the sidebar. If you started with one of our built-in templates, you may only need to attend to a

few "author" and "title" sorts of details over in the Compile Settings (section 23.4) area.

If you get a yellow warning in the middle column, that means none of your projects section types are set up to work with the format yet, and will simply print their associated documents verbatim. You will need to set up the format to work with your project's section types before getting the output it is designed to provide. Proceed to the Section Layouts (section 23.3) section for details on that.

Once you've set up a format to work with your project those settings will be stored for all time. Feel free to switch out other formats and experiment. The software is designed around the concept that one project may often require several different outputs, and as such doesn't penalise you for having multiple setups all ready to go with a few clicks. It's also worth noting these settings apply to any project templates you might create, too (subsection 5.4.3). If you use the same batch of settings over and over, consider saving yourself the effort by building your own starter kit—it is as easy as saving a project.

23.2.2 Built-In Formats

If you started your project using one of our built-in templates, it may already have special compile settings designed for it, but beyond that, Scrivener comes with a number of useful built-in formats. Some of these have been designed to conform to common industry standards in terms of manuscript submission and working with agents and editors; others have been designed as useful working tools, such as the ability to export an indented outline, or the distribution of proofreading copies. Others are intended to produce high-quality eBooks suitable for self-publication, or to formats suitable for further refinement in book design software for that purpose.

Our formats have all been designed to work seamlessly with the project templates that are built into Scrivener, and the default stylesheet that is provided with all new projects. For example, the "Modern" format will convert text styled with "Block Quote" to a suitable font and indent layout to fit with the body text it establishes, and will likewise adjust captions and headings that use these styles, from whatever fonts, paragraph spacing and other formatting attributes you see while working with the styles in the editor.[3]

23.2.3 Creating a New Format

We won't go into all of the sordid details involved in designing a format right here and now. Refer to The Compile Format Designer (chapter 24) for all of that. We

[3] We'll of course get into how styles work in detail in the documentation for designing formats themselves (section 24.5), but it would be good to know at the top that if you make your own stylesheets, the only thing you need to do to get them working with our built–in templates is to use the same names as we do—or modify the names the format itself looks for. Any style called "Block Quote" will be modified by the "Modern" format, not just specifically the default style.

will however go over the creation and management of formats themselves within this sidebar. To create a new format:

1. Optionally, first select a format you want to use as a basis for your own.
2. Click the + in the footer bar, and select "Duplicate & Edit Format..." if starting from another, or "New Format...", to begin a new format from a clean slate.
3. You can also right-click either directly on the format you want to duplicate, or anywhere at all if intending to create a new format from scratch.

23.2.4 Editing Formats

To edit a custom format you've made in the past, simple double-click on the format in the sidebar to open the format designer. You cannot edit built-in formats directly, so use the previous instructions to duplicate and then edit the new format.

23.2.5 Importing and Exporting Compile Formats

Formats (even the built-in examples) can be exported from the sidebar and imported into other projects, or other computers entirely. If you have downloaded a ".scrformat" file from the Internet for example, you could import it using the following procedure:

1. Click the ⚙ button (or right-click anywhere in the sidebar) and select the "Import Formats..." option.
2. Navigate to the .scrformat file(s) on the disk and select them, clicking Open to continue.
3. Finally, choose whether to import the files into the current project by clicking the Project Formats button, or instead make them available to all projects with the My Formats button.

Exporting a format will create this ".scrformat" file on the disk. This is a good way of backing up your hard work put into these formats, for sharing them with others or getting settings from one computer to another:

1. Select the format to export and click the ⚙ button, or right-click on the format, and select the "Export format..." command.
2. Choose a folder to save into, provide a name if necessary, and click Save.

23.2.6 Deleting a Format

You can remove formats from the sidebar that you no longer need:

1. Select the format you wish to remove in the sidebar.
2. Either right-click and select "Delete Format" from the contextual menu, or click the ⚙ button and select it from there.

This will only delete the copy for the format from the "My Formats" list, or "Project Formats" list. Thus if you had a copy saved into the former list, any projects that still have that format saved into their own settings will not lose it, and vice versa—they are not the same format once they have been saved into different projects or lists, even if they share a name and common ancestry.

23.2.7 Resolving Duplicate IDs

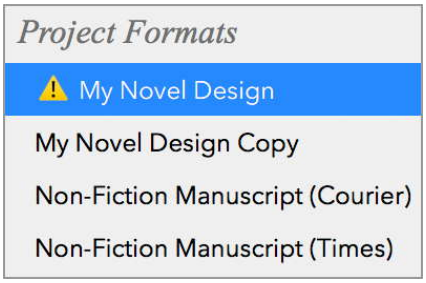


Figure 23.2: Duplicated formats will be indicated as needing resolution with this icon.

Duplicating Formats from within Scrivener is safe and often useful to do. Problems can arise if the files used to store formats on your disk are duplicated outside of Scrivener. This could happen if the associated file is restored from a backup and one chooses to keep both copies, as a result of sync conflicts, or just manually duplicating formats as file on the disk.

When two or more formats are using the same internal ID, Scrivener will place an "alert" icon beside the affected formats (Figure 23.2). You should resolve the ID clash sooner than later:

1. Right-click on the Format with an alert icon, or with any format selected use the ⚙ button in the format sidebar footer area, and select the "Fix Duplicate IDs..." menu option.
2. The following dialogue will show you which formats are currently duplicated. When you click the Fix button, the listed formats will have their internal IDs reset.

This may cause projects that were using those formats to lose their Section Layout assignments. This can be easily resolved by Assigning Section Layouts (subsection 23.3.2) again.

If that may prove too much of a hassle, you might prefer to resolve this problem back where it began, on the disk. Dragging the duplicate format file (.scrformat) to the Desktop and then using the import feature (subsection 23.2.5) will

automatically assign a new internal ID to the imported format, leaving any existing associations between the original format and the projects that use it intact.

23.2.8 Importing Legacy Presets

If you have been using Scrivener for some time, you may have built up a library of compile formats that you like to update, or you may have specific settings in some projects you would like to import as a starting point for getting them retrofitted for v3. You will first need to have your compile settings available as a formal preset in the legacy version (section E.2), as a file on your disk.

1. Click the ⚙ button in Formats footer bar.
2. Select the "Import Scrivener 2 Preset..." command.
3. Choose from one of the three options:
    a) Select a saved preset from the "Shared Scrivener 2 Presets" list.
    b) Click the Import Other... button if the preset was stored as an exported file, navigate to the file on the disk and select it.
    c) If the project you are working from has been upgraded from the previous version, then its compile settings will be provided for your convenience as "Last Settings Used".
4. Click the Import button (or the Open button if you've selected an external file).

The format designer will be opened with the imported settings all ready for inspection and adjustment.

Once you are done confirming the settings, click the Save button. The format will now appear in the "Formats" sidebar, ready for use. You'll need to assign section types in the middle column.

There are some things you can do to tidy up the format for better use in v3, within the format designer:

— If you intend these settings to only be available to the current project, use the **Save to** dropdown menu to select "Project Formats".

— If the format depended upon the "Quick Font Override" setting in the legacy version, this is no longer done at the format level, but rather as a per-project choice in the compile overview screen.

— The legacy version of Scrivener had no concept of named section layouts, strictly using folder, file and file groups along with levels to achieve a similar effect. If you click on the Section Layouts compile format pane you could go in and rename these generated layouts to something more meaningful than "Folders (Level 1+)" and so forth.

— While there, it is also worth noting that **Override text and notes formatting** is now an option available to each section layout individually, rather than a global setting. If you used this checkbox in v2 then all section layouts will have the checkbox enabled, causing the compile settings to act as they did before. However many of the additional options that were used to include/exclude particular *types* of formatting have been removed; they are now much more efficiently handled by the stylesheet system. Instead of preserving alignment so as to keep scene separators centre-aligned, you can now create a scene separator style.

— Imported formats will be available to *all* file types, since that is how presets worked in older versions. If you intend to use the format with only certain types (like PDF or ePub), you could click the ✲ button in the format pane header bar to narrow its scope. This will limit the choices available in the dropdown menu to the left of the gear button, which in turn determines which panes and what options within them will be available.

— For a more detailed guide, we have prepared a tutorial project that you can download from our website[4].

You should find many of these panes will be familiar from the old compiler's option panes, but should you require assistance with any of them, refer to the The Compile Format Designer (chapter 24).

Return to chapter ⤴

23.3 Section Layouts

The next major component falls within the centre column of the compile overview screen. It contains a stack of tiles depicting how the various different types of document of your draft (section 7.6)—what you can see listed in the right column of the compile overview—will be formatted.[5]

You can think of what happens within this middle column as project specific settings for each of the formats listed to the left in the sidebar. A format can say in general terms what a section heading might look like, but we need a good way of saying which pieces of our draft should look like that heading—or maybe if they should look like a different style of heading also provided by the format. In Figure 23.3 we can see several types of major heading break presented by the format, and have chosen one suitable for numbering and printing the name of each short story following a page break.

[4] https://www.literatureandlatte.com/scrivener-3-update-guide

[5] By the way, you can click on any preview tile in this column to highlight which sections will be impacted in the contents list to the right, and hover the mouse over the tile to read its name in a tooltip.

Consequently, changes you make to the centre column are very specific: they will be saved for that format in relation to this project. By example, if you select the "Modern" format and change the font to Helvetica (using , no other project will use Helvetica for Modern but this one project. Likewise, if you later switch the format to "Manuscript (Times)", you won't find the "Helvetica" settings overwriting this format's use of Times New Roman, but when you switch back to "Modern" it will still be using Helvetica.

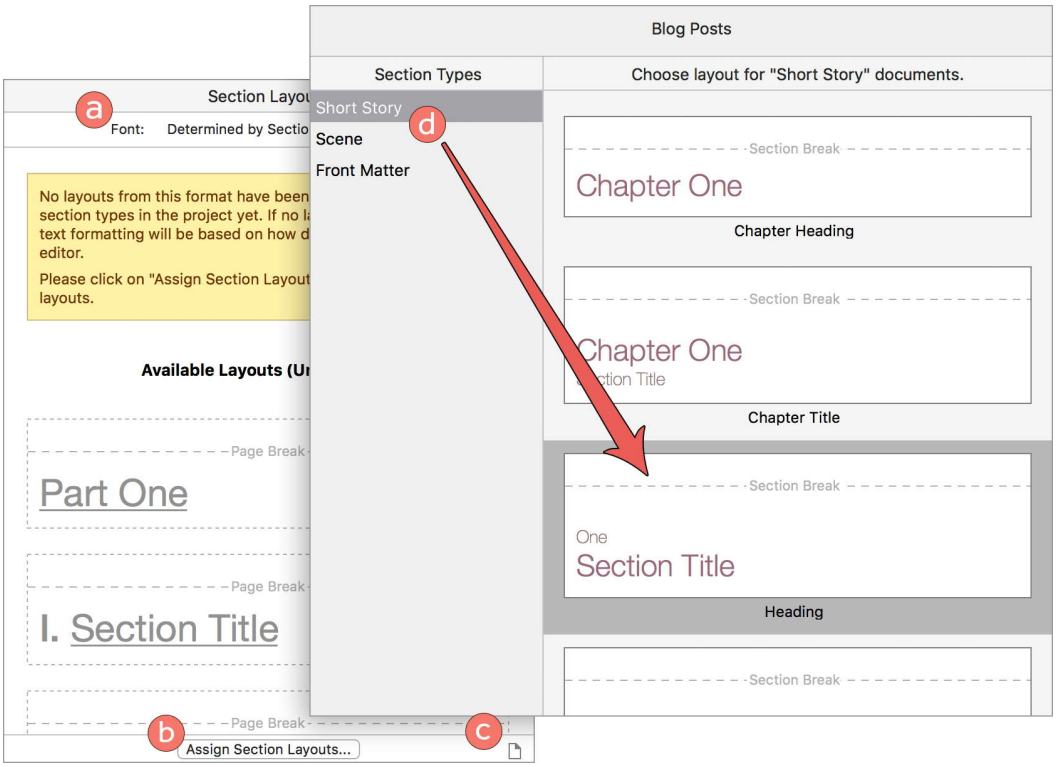


Figure 23.3: Section Layouts are how you choose the appearance and function of your document.

Working off of our earlier example, in Figure 23.3 we have what you will see in the compile overview screen on the left, and overlapping this excerpt on the right side is the interface used to assign section types to layouts:

a) At the very top of this column is the **Font** chooser. Use this to override the font for the selected format, or let each section layout from the format determine font settings (which may be to even defer to how you wrote in the editor).

b) The Assign Section Layouts... button is what brings up the interface shown on the right side of the figure. Use this to change how documents in your project will look when compiled. If you have a Touch Bar, this function is also featured as a convenient button from the overview screen.

c) Click the page layout preview button to double-check your print settings. You can change paper size and margins in the File ▸ Page Setup… tool, so if you have to leave compile for a moment to change that, you might want to hold down the Option key and click on the Save button, also featured on the Touch Bar, so as to not lose your settings.

d) Within the "Assign Section Layouts" interface, we can select section types from the sidebar and then click on preview tiles on the right hand side to choose a look for that particular type of document. In this case we have chosen to print our "Short Story" type documents with a page break, an automatically generated number and finally the name of the short story on a second line.

23.3.1 Choosing the Global Font

Many formats will include their own font choices for the text, headings and so forth. These will be used so long as the **Font** setting along the top is "Determined by Section Layout".

If you want a different font used throughout the document then use this menu to select your preferred font. The scope of this setting is quite simple: it impacts everything, from the page number on down even to section types that would ordinarily leave the text alone and pass the formatting straight through from the editor—even styles from the editor that would otherwise declare a font family will be overridden by this setting. If you need a finer level of control (perhaps you want chapter headings to use a different font than body text for example) then you will need to edit the format itself.

23.3.2 Assigning Section Layouts

When switching to a format not yet set up to work with your project, you may get a yellow warning above the preview tiles (Figure 23.3). This merely means that none of the section types in your project have been "mapped" to a particular look in the Layouts section.

It is perfectly fine to compile in that state; the look of the format will not be used but other aspects of it may be, such as paper size, footnote settings and so forth. But to fully take advantage of a format you will want to tell the compiler what the document should look like. Click the Assign Section Layouts… button along the bottom of the column to get started.

The left side of the layout assignment window lists all of the types your project uses, in the order they are defined within the Project Settings: Section Types tab (section C.2).

— Click on a type in this list to see which Layout it is assigned to. You can change the assignment or make a new one by simply clicking on the desired preview tile on the right-hand side.

— 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:.

1. 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 5.4.3).

23.4.1 Contents Tab

Figure 23.4: 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 23.5. 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.

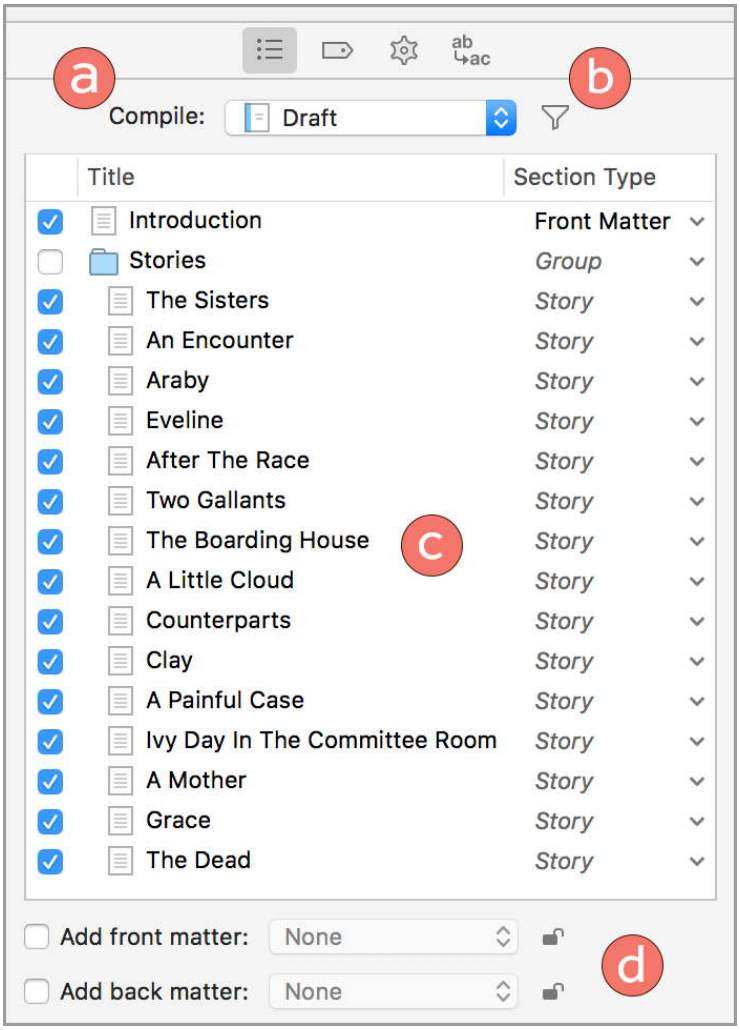


Figure 23.5: The compile contents list is used to establish the material that will comprise the final document.

In addition to selecting a subgroup of the draft folder, there 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 concerned with 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 23.5, 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 option 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 23.5, alongside the main **Compile** group dropdown.

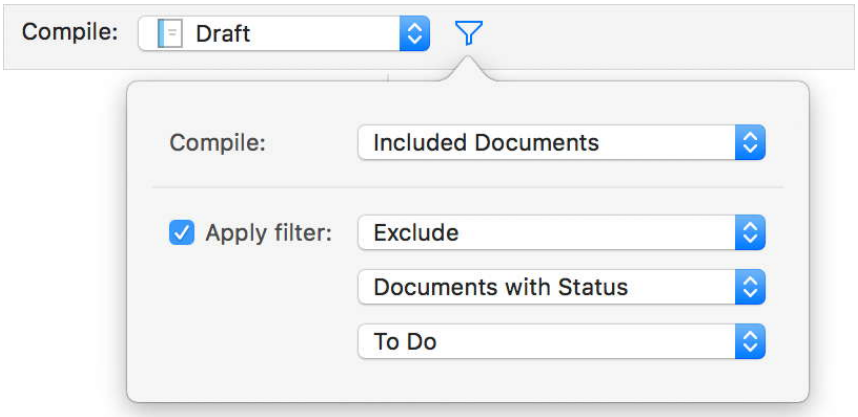


Figure 23.6: The "Filter" button reveals options for further refining the compile contents list.

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 23.7, 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.

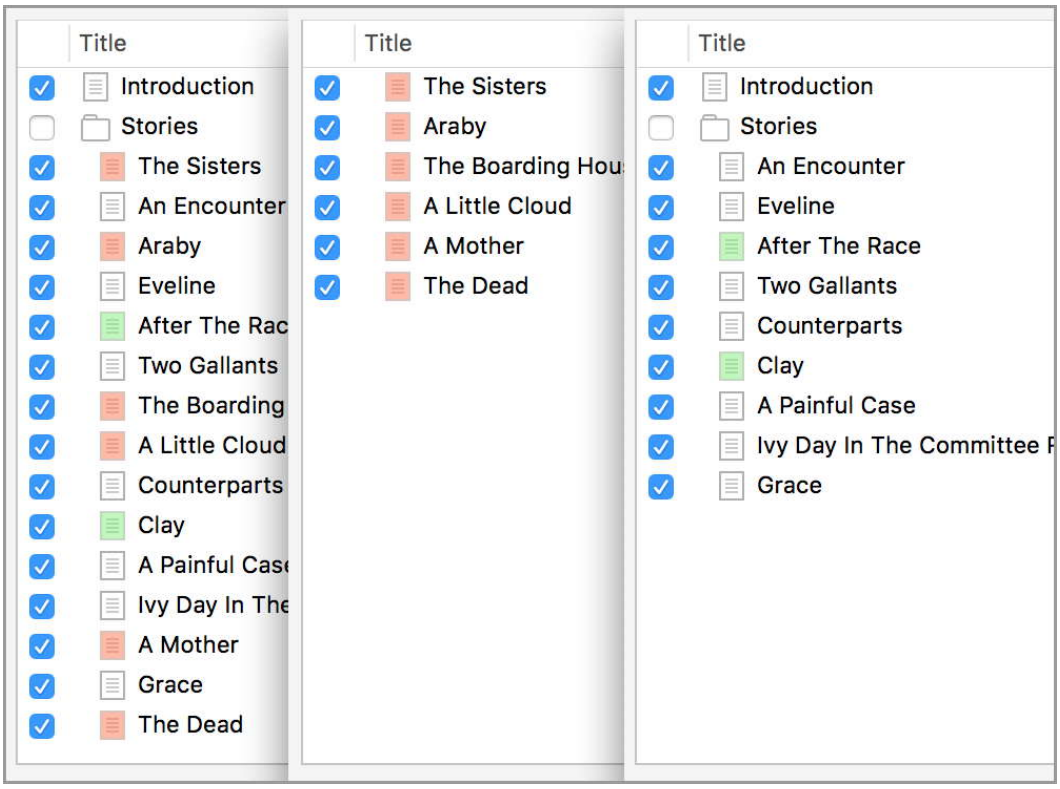


Figure 23.7: Filtering by "Red" produces either a list of only red marked items (centre) or a list with no red marked items (right).

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 include 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-

put will be presumed as only a partial manuscript, and thus adding full book material would be undesirable:

— The entire draft folder is selected as the compile group in the dropdown at the top of the contents list.

— If a subfolder of the draft is selected and the **Treat compile group as complete manuscript** option is enabled.

Additional Formatting Impacts

Besides swapping out sets for multiple outputs, these features will also commonly be used to impact other compile settings:

— When publishing to the Kindle formats the compiler will by default set the point where the book will first open to fall after the front matter contents—traditionally at chapter one. This can be altered in the General options tab of the compile overview screen (section 23.4.3).

— For formats with a page header or footer, different settings can be applied to front and back matter independently, in the Page Settings compile format pane (section 24.20).

— Placeholders that print statistics, such as word and character counts, will by default exclude material found in the front or back matter. This can be changed in the Statistics compile format pane (section 24.16).

— The **Document suffix**, an option of the compile format, in the Text Layout format pane (section 24.6), which can be used to insert phrases like "The End", can be optionally inserted before the back matter begins.

Setting Up the Folders

The Front & Back Matter features, marked (d) in Figure 23.5, make it a simple task to swap out sets of files depending on what your current compile file type is. To get started:

1. Create a folder in your binder that will contain different sets of front or back matter. In some cases you'll want to put separate subfolders within this, one for each type of front matter you'll need, such as "Amazon", "PDF" and so forth.
2. Move your title pages and other materials into these folders. Each folder needs to be a self-contained set of front and back matter. If you want to use the same title page for multiple formats, consider making use of the `<$include>` placeholder (subsection 10.1.5).

Want to see front matter in action?
The "Novel" project template provided with Scrivener demonstrates a possible setup with a group of individual front matter folders for use with corresponding publication outputs, such as standard submission or self-publication with eBook and PDF.

Selecting a folder as a front/back matter folder will transpose all of the contained items (and descendants) to the very top or bottom of the compile contents list respectively, without altering the actual order of these items in the binder. This will happen no matter where the source folder is located (although to avoid confusion, you should never place these folders inside the draft folder itself).

Locking Settings to File Type

As noted above, it will often be desirable to have different sets of front and back matter depending on which type of file you are creating. By default, when you set the contents pane to use a particular front/back matter folder it will stick no matter which choice you make with the **Compile for** dropdown at the top of the overview screen.

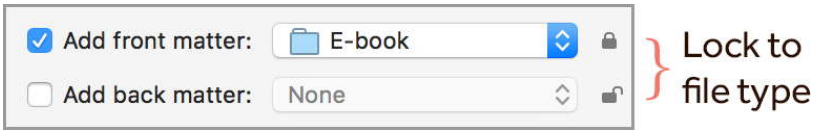


Figure 23.8: The "eBook" front matter folder will always be used for the selected file type.

To lock a front/back matter folder to the current file type:

1. Set the folder(s) you want to use with this file type (such as ePub 3).
2. Click the lock icon to the right of the folder selection dropdown, as indicated in Figure 23.8.

File types that have not yet been locked will always use the last setting used before switching, which may in some cases be the folder you've locked to another type (though the lock icon will disable, indicating that this folder is not locked to the *current* file type. By example:

1. You set ePub 3 to use the "eBook" front matter folder and click the lock icon.
2. You select **Compile for**: "PDF". The front matter folder will still read "eBook" because that is the last setting you used. So you select the "Print" front matter folder and lock it to PDF.

3. Now you select **Compile for**: "Kindle KF8/Mobi eBook". The front matter folder reads "Print" because that is the last setting you used. You switch the front matter folder to "Amazon eBook", but you forget to lock it.
4. You select **Compile for**: "ePub 3", and the front matter folder setting automatically selects "eBook".
5. Switching back to Kindle, the front matter folder sticks to "eBook" because it was not locked before.

Linking Front/Back Matter to Compile Groups

You can set the compiler to automatically select a front matter folder based upon the current compile group, so long as the group is itself set to function as a complete manuscript (section 23.4.1). A common case for where this might be useful is if you are working with a series of related works in one project, and regularly switch between them when compiling.

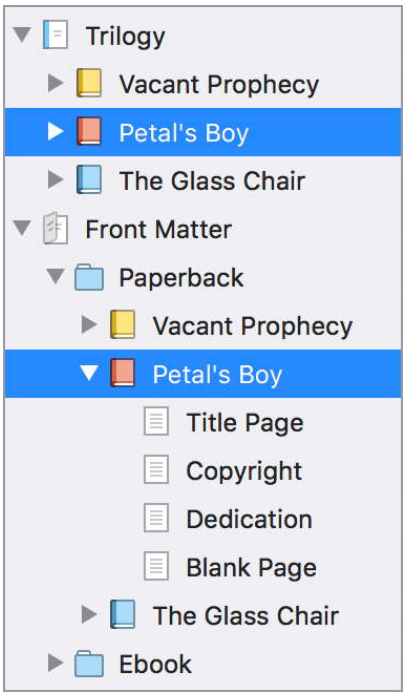


Figure 23.9: Matching front/back matter folder names will promote automatic selection when switching compile groups.

1. In the binder, create a set of subfolders beneath the main front/back matter folder that match the name of the main folder used to designate the work being compiled.

   E.g. In Figure 23.9, we have a trilogy in the Draft folder, with the Paperback front matter files sorted into matching subfolders.

2. In the compile overview screen, designate the main front/back matter folder that contains the paired subfolders. In our example, we would select the "Paperback" folder.
3. From the compile group selection dropdown, select one of the novels. If "Petal's Boy" is selected as our compile group, and **Treat compile group as complete manuscript** is set true, then Scrivener will automatically select the "Petal's Boy" front matter subfolder for you.

This capability can be combined with the previous mechanism for locking front/back matter folders to file types. Continuing on in our example here, we could lock the PDF type to the "Paperback" front matter folder, having it automatically switch between the three subfolders depending upon which novel we compile. When we switch to ePub3—if set up similarly and previously locked to the "Ebook" front matter folder—the compiler first switch to the "Ebook" group and then look within that group for a "Petal's Boy" subfolder, ultimately selecting it for the ePub.

23.4.2 Metadata Tab

Figure 23.10: The "Metadata" compile settings tab.

Metadata in this context refers to information that will be embedded in the file you create, using whatever properties or fields the selected file type has at its disposal for recording metadata. This information can sometimes be used by file managers and to enhance search tools like Spotlight.

This pane also sets the information that will be used by placeholders that insert information such as the author name and book title into various places—such as the page headers in some formats. The contents of this pane will change a good deal depending on the file type chosen (some will have no options), but the placeholder settings are available to all file types.

Book and Author Information

Title If nothing has been provided in this field, the title will be taken from the name of project itself on the disk. Use this to set the formal title of your work; it will be used wherever the `<$projecttitle>` placeholder has been typed in.

Abbreviated Title Especially useful in headers, long titles often need to be shortened to fit in the header area, so you can optionally use the placeholder, `<$abbr_projecttitle>`, to display a shortened version of it. If nothing is supplied here, it will use the **Title** value.

Author's Full Name, Forename and Surname The first field can be used to add multiple authors if necessary. Separate each author name with a semicolon. The <$fullname> (or <$author>) placeholder will insert this information into the book, and the <$forename> and <$surname> (<$firstname> and <$lastname> can also be used if you find those more comfortable) placeholders will be used elsewhere.

Word Processing and Web Metadata

The RTF, RTFD, DOC, DOCX and ODT formats all support additional fields that can be saved into the properties of the file.[6] You may need to consult guidelines for how they should be used if submitting documents to a central repository. Some of these fields may be used by search indexers to better categorise the documents among others.

When used with HTML files, the information supplied here will be inserted into the document header, which will be used by search engines to optimise discovery of the page.

The **Keywords** and **Subjects** fields should have each value separated by a comma. Otherwise all fields are freeform.

Ebook Metadata

The additional fields provided to eBooks will be inserted into appropriate book description fields, often available to e-readers to display in whatever manner they provide, and used to automatically sort the book within electronic catalogues. Most of the fields allow for freeform entry; you should consult with your publisher or distributor on best practices for using these fields.

Contributors As with the **Authors** field, each person's name should be separated with a semicolon. Amazon's KindleGen utility does not recognise multiple authors, so only the first name contained within a bubble will be applied to the .mobi book. To ensure all authors are listed in the .mobi file, refrain from using semi-colons to separate the names, or add them later with a tool that is capable of adding multiple names to the .mobi file you've compiled.

Subjects : A listing of topics or subjects the work covers. This is not commonly used in fictional works. Each subject should be separated with a comma.

Date In most cases, to conform with standards, you should use YYYY (1914), YYYY-MM (1914–06) or YYYY-MM-DD (1914–06–27) date formats.

[6] For example, the information filled out in this area will appear in the document's File ▸ Properties... command.

Language Code Provide the standard ISO language code of the language the book is published in.

Use custom book ID A randomised unique number will be generated if you do not provide one yourself. In most cases you will want to use some internationally recognised identified for your book, such as the ISBN.

If multiple authors or contributors have been accredited in the fields above, the Author and Contributor Details... button can be clicked to assign roles ("aut" stands for "author" and the rest are set to "default") and freeform filing information.

Fountain and Markdown-based Metadata

The Metadata pane for Fountain and the Markdown-based file types provides a front-end to the preliminary metadata block found at the top of these kinds of files. Pandoc format uses a YAML[7] block, which for the purposes used in Scrivener matches the format used in MultiMarkdown and Fountain's metadata block. Each line in the block is prefaced by a key, such as "Title", followed by a colon, some whitespace and the value.

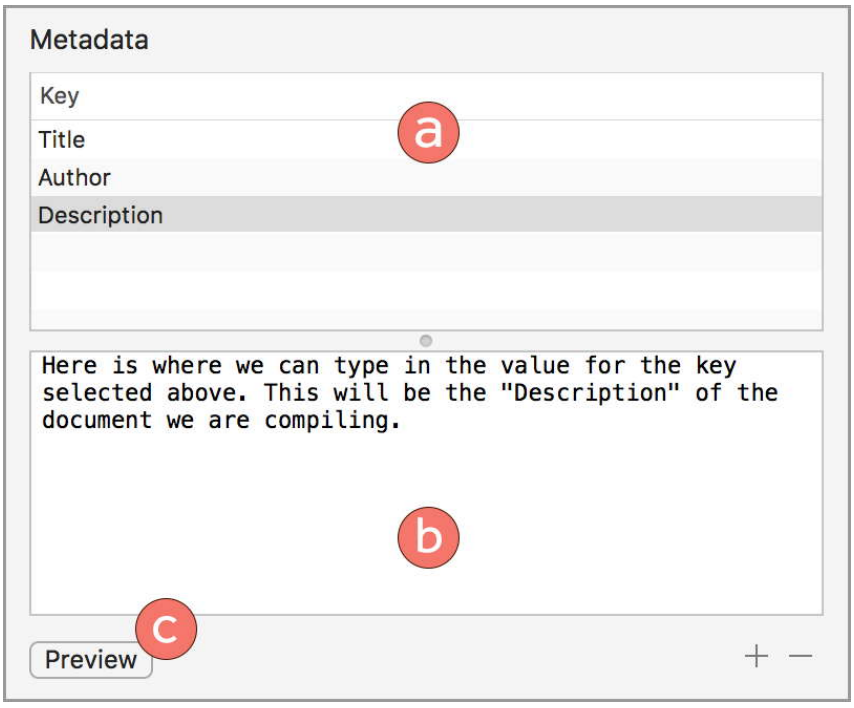


Figure 23.11: The Fountain and Markdown-based metadata panes make it easy to manage fields for these document types.

[7] http://yaml.org/

By default, new projects will have both a Title and Author key added for you, using placeholders to insert the actual information (taken from the general metadata fields above). Thus the starter metadata for a new document, when compiled to a Markdown or Fountain file, will look like the following (with the placeholders evaluated to the project name and user account name by default):

```
Title: <$projecttitle>
Author: <$author>
```

What you type into the text area, marked (b) in Figure 23.11, will be printed into the compiled document verbatim. Thus, all rules pertaining to the metadata syntax should be followed. Empty lines in this area may cause the metadata block to prematurely abort.

To create a new field:

1. Click the + button in the lower right-hand corner of the metadata area.
2. A new row will be created in the area marked (a) in the figure. Type in the name, whether it be to conform to a functional field or one of your own.
3. Click into the text area marked (b) to type in the value. You can use placeholders here to extrapolate information dynamically from the project, current date and so on.

If you already have a metadata block from an existing document, you can copy and paste it into your settings in their entirety rather than creating each field by hand and transferring the values over one by one:

1. Click into the Key area, marked (a) in the figure.
2. Use the standard Paste command to paste a properly formatted metadata block into the compile settings, having them converted to individual key rows.

Individual fields can be arranged by dragging and dropping them in the key list of the upper half of the metadata area. Key order can be important with some formats.

For further information on how metadata should be used, particularly whether keys should all be added to this table, into the compile format settings (section 24.11), or as a "metadata" file in the binder itself, refer to MMD & Pandoc Metadata (section 21.7).

23.4.3 General Options

The third tab in the compile settings area contains settings for adjusting the behaviour and appearance of text and its formatting, cleaning up the document of struck-through text, hyperlinks, unnecessary whitespace or refitting images

Figure 23.12: The "Options" compile settings tab.

to the paper size, inserting proofing aids and so forth. Many settings are only relevant to certain types of files, and thus the options you find here will depend greatly on the "**Compile for**" setting at the top of the compile overview screen. If you do not find what you are looking for in this list (such as whether italics should be underlined), chances are you will find them in the compile format itself (chapter 24).

Common Settings

There are a few options in this tab that pertain to all types of documents that can be produced with Scrivener, or are used by a large majority of them.

Remove footnotes The exported manuscript will have all footnotes (inline or linked) stripped out.

Remove comments All linked comments from the inspector will be stripped out. When disabled, comments will be included in accordance with the current compile format, or to a format most suitable for the type of document being exported (for example, with the rich text formats you might get a "margin note" or comment box in your word processor when opening the file).

Remove annotations All inline annotations will be stripped out. As above, when this option is disabled the annotations will be formatted in a manner suiting the type of document and the compile format settings.

Delete struck-through text Any text that has been struck-through in the output will be stripped out of the compiled copy. If you use these marking for editorial purposes, to "soft delete" text, this is the option to use to implement those edits. When disabled (as it is by default), the text will be formatted as struck-through text if the document type supports that form of formatting. They will otherwise look like ordinary text in formats like plain-text.

Remove trailing whitespace from documents When enabled, this option makes it viable to not worry so much about whether or not there are empty lines at the bottom of individual chunks of text in the binder. They will be stripped out, so that only the separators inserted by the compile format will be present between chunks of text. If you use empty lines for some important purpose, you'll probably want to leave this option off.

Insert links back to Scrivener in each section Using whatever mechanism is most appropriate for displaying a hyperlink in the target file type, Scrivener will insert a special link at the top of every chunk of text used to build a compiled document, pointing back to that item in the project itself. This link will work anywhere from the same machine that project is located on. You can for example compile to PDF and use your favourite PDF reader to proof, clicking on these links to load sections up in Scrivener when coming across areas that need fixing. Read more about this capability, and other Proofreading Tools (section 20.2).

This and the following options are not available to the eBook and Final Draft document types.

Insert unique document identifiers only This subsidiary option to the prior will not create a usable link in the exported file. However it will print the internal identifier (in UUID format) for the binder items used to build the compiled document. When importing such files back into Scrivener, it will scan for text that matches this notation and build document links based upon them. This is a safer approach to distributing proofing copies to readers, as some software might disturb the hyperlinks inserted using the option above—as well it will work even with formats that do not support hyperlinks natively, such as plain-text.

Rich Text Options

Unless otherwise noted, in this case, "rich text" refers to not only the word processing document formats (such as RTF, DOCX and ODT), but eBooks, HTML, PDF and Print.

Export inspector/inline footnotes as endnotes These two checkboxes, one for footnotes linked to the inspector and the other for footnotes typed directly inline into the text, will toggle whether or not that particular style of footnote will be exported as *endnotes* instead. The manner in which this is done will be determined by the compile format in use.

Remove highlighting Text highlights will be stripped from the compiled document in all cases, save for when highlights are used for other purposes, like indicating commented text.

Remove text color Likewise, this option will strip all custom colouring from the compiled document and force all text to default (black in most cases), without exception.

Remove all hyperlinks Strips out all hyperlinks from the produced document. If you primarily use links as a form of internal referencing, rather than as something you would like the reader to be able to jump from one section of the book to another with, you will want to use this option. It will also

be useful when printing to paper, as you'd likely not want light grey underlined text wherever a link appears.

Convert Markdown to rich text in titles and synopses Simple Markdown[8] for `**strong**` and `*emphasis*` (or `_emphasis_`) found within titles or synopses will be converted to bold and italics respectively. When using the underscore formation with the **Convert underscores to underlines when converting Markdown** setting in the Sharing: Export preference tab (subsection B.7.2), then the end result will be formatted underlines rather than italics.

Convert MultiMarkdown to rich text in notes and text If you have written your text using MultiMarkdown syntax (some Pandoc may also work, so long as it overlaps with MMD), then use this option to have your work converted to rich text so that it can be used with one of Scrivener's own native formats, rather than using one of the dedicated MultiMarkdown compile file types. With the exception of the formatting Scrivener has native conversion support for (like images), this option requires a project to be written with proper with syntax throughout. If you are using a hybrid workflow, using some rich text (like bullet lists) and some markup, you should use one of the dedicated Markdown-based formats instead, and employ its options for converting rich text to Markdown.

Scale images to fit page width Available to the word processing formats, images in your text editor will have their dimensions adjusted to fit within the set paper and margin settings rather than overflowing the page. In nearly every case it will be better to actually resize those images graphically, so that they print the way they are supposed to, as this will reduce processing overhead and result in smaller documents.

Convert document links to link back to Scrivener This proofing option is available to PDF and the word processing formats. All document links, those that point from one section of text to another within the binder, will be converted to links that point back to that chunk of text in your project, rather than links that navigate you around within the document you exported. Read more about this capability, and other Proofreading Tools (section 20.2).

Markup Options

These settings pertain the Markdown-based document types at the bottom of the **Compile for** dropdown. Pandoc to ePub also supports some options described in the next section.

[8] http://daringfireball.net/projects/markdown/

Convert rich text to MultiMarkdown Enables a broad spectrum of rich text to MMD conversions, with the intention of providing as much conversion as there is overlap between these two systems. That is to say, things that RTF does that Markdown-based formats do not do will not be converted (you can't have red text for no reason, for example), and things that Markdown can do that RTF cannot do will not be addressed. But if both can do a thing, like create a text link to a web page, then we strive to convert it.

This option will primarily be of use to those that *don't* use Markdown in their writing. In fact, it strives to keep the text of the document you create similar (in terms of content) to what you might get if you compiled the same text to PDF. To that end it will escape punctuation marks found within the text that might be used by Markdown, to ensure they remain visible as they would in RTF, ePub and so forth. If you use Scrivener as a rich text editor but find yourself wishing you could make use of its technical formats, like DocBook or LaTeX, this is the checkbox for you.

Heading styles can be used to generate Markdown syntax, if those styles have had an HTML level saved into them, as applied with the Format ▸ Paragraph ▸ HTML Header Level ▸ submenu.

It is possible to use Markdown with this checkbox, but you will need to specifically instruct Scrivener of your intention to do so. The easiest way to do this will be to make use of styles to denote Markdown syntax, and in the compile format designer's Styles pane, set that style to **Treat as raw markup** (section 24.5).

Convert tables and lists to MultiMarkdown When enabled, converts rich text tables and bullet/enumeration lists to MultiMarkdown and Pandoc Markdown (lists will be Markdown compatible). For further information on how this feature works, refer to Markdown and Scrivener (subsection 21.5.2).

Operating as a conversion subset, this will be disabled when the **Convert rich text to MultiMarkdown** is enabled. This option otherwise treats your text as Markdown, unlike the previous option. You can freely use your own markup, and thus it is meant to be complementary with a hybrid writing method, as described in Markdown and Scrivener (section 21.5).

Treat "Preserve Formatting" as raw markup Requires the **Convert rich text to MultiMarkdown** checkbox to be ticked. If now and then you need to do something that Scrivener's converter cannot handle, use this setting to insert your own hand-typed markup into a document that is otherwise entirely rich text. Mark such blocks of text with the Format ▸ Preserve Formatting menu command to cause the compiler to leave these spans of text alone, rather than "escaping" the markup so that it prints verbatim.

Show PDF log ⟨Direct-sale only⟩ When using the MMD → PDF conversion option, this option will print the LaTeX log file that is generated when typeset-

ting the document. Use this to help locate and fix problematic formatting, figure out errors that block output and so forth.

Save source files in a folder with exported file Available to Pandoc → DOCX, DocBook and ePub. Ordinarily Scrivener will discard all of the files it generates when producing an encapsulated format or one with embedded source files. The graphics, CSS, Markdown and other peripheral files used to create the requested document will be saved along with it. In this way you can tweak the output using additional tools for doing so, processing the files yourself using the command-line or other tools.

Ebook & HTML Options

With the exception of the first option in the following list, these settings are only available for eBooks.

Convert document links to HTML links Internal references to other items also located in the compiled output will be cross-references for the reader. In web pages and eBooks, this means people will be able to tap on the linked text and be taken to the section you refer to. This will even work if the sections themselves have been glued together into one longer formal section—like subsections within a chapter.

This option has no impact on the automatically generated table of contents in eBooks, or a designated contents section in the draft, only the links you create yourself in the text.

Downsize and resample inline images to visible size To cut down on the overall size of your eBook, you may wish to have Scrivener resample the graphics when compiling. This is distinct from the ordinary behaviour, where images will always *appear* to be the size you have set them to in the editor, or the size they appear as naturally. This is done by setting the visible size of the picture without changing its underlying pixels. This option on the other hand will physically resize the graphics for the compiled version *only*. It is not destructive; the original images in the project will remain untouched, as is the case for nearly every function in the compiler.

Use pop-up footnotes There is no reliable part of the ePub specification that can be used to present footnotes in a pop-up bubble, as opposed to a link that will take the reader to another point in the book. But there is a loose convention that some readers will follow, and this option adds the necessary syntax to the HTML to do so. The option is only available to ePub 3.

Include standard Adobe Digital Editions page template Only relevant to the legacy ePub 2 format. : Adobe Digital Editions[9] uses a special proprietary file (page-template.xpgt) for setting the type and controlling layout look. Without this file, the appearance of your eBook may suffer in ADE and devices that use that technology, such as Sony, Nook and Kobo readers. This file is not considered part of the official ePub specification however, and can conflict with some validation and publishing services, most notably Apple's bookstore. If you do not intend to distribute through the bookstore, it would be a good idea to enable this option so that the book will appear as you intend it to on a wide variety of systems.

Optimize for Kindle conversion For cases where you wish to publish to Kindle format, but are using a third-party agent only accepts ePub files, this option will enable additional tweaks to the structure and formatting of the ebook that will produce a better result on Kindle devices and apps.

Save source files in a folder with exported Kindle or ePub file This is an advanced feature intended for those who wish to customise the look and feel or structural layout of their eBook after compiling it. The actual files used to create the eBook will be output as plain folders and files for you. These can then be edited freely, and then turned into a .mobi file using KindleGen or Kindle Previewer. If you are intending to modify an ePub file further, it may be easier to use an ePub editor, such as Calibre[10].

Save KindleGen log file with exported Kindle file If you are having difficulties getting the KindleGen program to produce a valid .mobi file, enabling this option can help you determine what is going wrong. The .log file will be saved right alongside the compiled .mobi, or where it would have been saved were it not for the errors.

If no log file is created, there may be a problem with KindleGen itself and you should try reinstalling it.

Book begins after front matter For this feature to work, the Front Matter feature must be enabled (section 23.4.1). The book will initially open at the first section that follows the material added as front matter. This can be useful if the front matter you are inserting with the feature is not essential for the reader to see, and you wish for them to start right at chapter one. When unchecked, the book will open at the start of the entire book, and so is thus useful when the front matter contains text that is important for the reader to go through.

[9] http://www.adobe.com/products/digital-editions.html

[10] http://calibre-ebook.com/

You can also manually supply a starting point by typing in the placeholder, `<$ebook_start>` into the section you wish the reader to start at.[11] The placeholder will be ignored if this checkbox is active.

Scriptwriting Options

The Fountain (.fountain) and Final Draft (FDX) formats offer some unique capabilities that various components of your project could be used for.

First document is title page If the first document in your current compile group is a special title page with no script formatting, leave this option enabled. It will be assigned to Final Draft's title page window, and kept separate from the script itself. If you've noticed the first page of your script seems to be missing, and you aren't using a title page, make sure this option is disabled.

Include document titles as scene titles (section headers) Enabled by default, the names of draft items as listed in the binder will be used as scene titles in Final Draft, or as section headers in Fountain. When Scrivener imports an FDX or Fountain file (via the File ▸ Import ▸ Import and Split... menu command), these will be used to generate binder item names.

Section headers use binder indenting levels Fountain section headers can optionally use outline depth, although this is not used in any way beyond the purely visual.

Include document synopses as scene summaries The content of the synopsis field will be used to populate information into the "scene summaries" feature in Final Draft, and using Fountain syntax for the same.

Include footnotes as script notes Inline and linked footnotes will be converted to script notes. Footnotes will be removed from the compiled document if this option is not enabled.

Include comments and annotations as script notes Inline annotations and linked comments will be converted to script notes. All comments will be removed from the compiled document if this option is not enabled.

23.4.4 Project Replacements

The fourth common tab, available to all file types, provides a way for you to set up your own text substitutions, which work in a manner very similar to Search and Replace, though without changing the source text used to compile.

[11] Some Kindle devices may have difficulties starting directly in the table of contents page. You should place this marker on the page directly preceding the ToC if you want the reader to start there.

Figure 23.13: The "Replacements" compile settings tab.

Some example usages would be to replace an abbreviated version with a full proper name, to ease typing it in frequently, easier usage of placeholder tags or inserting common editing notes (Figure 23.14).

	Replace	With	RegEx	Case-Sensitive	Whole Word
☑	AdT	Arc de Triomphe	☐	☑	☐
☑	!here	Amy Holmes	☐	☐	☐
☑	!cite	(Needs Citation)	☐	☐	☐
☑	Sam	Samantha	☐	☐	☑
☑	figCapt($@)	Fig. <$n:figure:$@...	☐	☐	☐

Figure 23.14: Replacements can be used for a variety of purposes, from saving time to keeping complex placeholders out of the editor.

The replacement table consists of several columns:

— A checkbox will appear before every replacement row in the table. This can be used to temporarily disable a replacement you might not need, but wish to keep in case you do later on.

— **Replace**: the text which you wish to instruct the compiler to look for.

— **With**: whatever has been supplied in the *Replace* column will be replaced with what you type into this column.

— **RegEx**: enables the regular expression engine (section 11.7) for both the **Replace** and **With** columns. When referring to stored values in the initial search pattern, use dollar-sign syntax, such as `$1`.

— **Case-Sensitive**: when checked, the letter case in the *Replace* column must precisely match, otherwise it will be ignored by the compiler.

— **Whole Word**: restricts the match to only those cases where the text is bounded by space or punctuation. "Sam" will not match "Sammy" when this option is active.

Creating Replacements

To create a new replacement:

1. Click the + button in the lower right-hand corner of the table, or click into the table and use the Return key to create a new row.

2. Type in the text you want to have changed when compiling.
3. Press Tab or click into the **With** field and type in what the text should be changed to.
4. Press Esc or click elsewhere to commit your changes, and tick any relevant checkboxes to the right of the text columns.

Looking for what you cannot see?

If you need to include whitespace characters, such as carriage returns or tabs, in your search or replacement field, it is possible to do so by holding down the Option key and then pressing the Return or Tab key You will not see the character itself, but it should impact the contents of the field in an expected fashion.

Removing and Disabling Replacements

To delete a replacement permanently (if you merely wish to disable the replacement, click the checkbox in the leftmost column of the corresponding row):

1. Select the row you wish to delete by clicking on it in the table, or using the arrow keys on your keyboard.
2. Click the − button in the lower right-hand corner, or press the Delete key.

Modifying Replacement Order

The order of replacements in the list can be significant as they will be evaluated one after the other from the top of the list to the bottom. This means you can use use earlier replacements in your subsequent replacements, if you are feeling brave. Thus you can change the order of replacements by dragging and dropping them in the table.

Copying Replacements Between Projects

You can copy and paste selected replacements (use the Shift and Cmd keys to select with the mouse or keyboard arrows, or Cmd-A to select them all) between projects using the ordinary methods for copy and paste. Just make sure to click into the replacements table so that when you paste Scrivener knows where you meant to paste the replacements.

If both projects have their compile overviews open at once and you have enough screen space to do so, you can also drag and drop between tables.

Advanced Replacements Usage

Replacements can also take a special placeholder (or wildcard), `$@` which will be used to match everything between the rest of what you type in the search field. The matching text will be temporarily stored and pasted into the With field if you use the same wildcard placeholder there. This wildcard only works when surrounded by text to match with, it cannot be reliably used at the start or end of the Replace field, though the placeholder can be positioned anywhere you like in the **With** field, even all by itself.

This is probably much more easily explained with some examples. Let's say you have typed the following into the **Replace** field:

```
^$@^
```

When compiling, if a caret symbol is encountered in the source text, it will seek for a second caret symbol on that same line and if there is one, copy everything in between those two symbols. Now in the **With** field we supply the following text, which includes that same wildcard placeholder:

```
<span class="index">$@</span>
```

The carets, which were part of the replacement, will be discarded (unless you type them in again into the **With** field of course) and replaced with the additional text found around the `$@` wildcard. Let's look at a before and after example. The following is an example of what you would type into the editor while writing:

```
This is a ^test^ of advanced replacements.
```

When this text is compiled with the described replacement enabled, the compiled output will look like this:

```
This is a <span class="index">test</span> of advanced replacements.
```

You can also use the wildcard in the **With** field multiple times. Here is another example, using the same replacement search pattern from before, but with a different output:

```
\index{$@}$@
```

Now with the same exact text typed into the editor, when compiling with this replacement, we get:

```
This is a \index{test}test of advanced replacements.
```

As you can see, it is possible to produce multiple technical formats from Scrivener using the same written text in the editor. Though in this case, you might want to take a look at how compile formats themselves can have stored replacements, so that you can switch these outputs on the fly merely by changing formats (section 24.15).

Need more?

If you require even more power when searching and replacing, you should consider using Regular Expressions (section 11.7) instead of the basic string matching engine. It comes with its own storage wildcard system, and in fact can store multiple values for later use in the **With** field, and doesn't have the limitation of requiring text on either side of the wildcards, or including the criteria for replacement in the text that gets replaced.

23.4.5 Cover Options

Figure 23.15: The "Cover" ebook compile settings tab.

Available only to the ePub and Kindle file types, the Cover pane sets the cover and presentation material which will give your eBook a professional finish. The cover image will be placed at the very beginning of the eBook (before the table of contents), and for ePub files, can optionally be included as the thumbnail image that will appear when the book is viewed on the "shelf" in the iTunes library and within iBooks mobile. Most eBook readers automatically use the cover image itself to provide their thumbnails.

To begin, you will need to import the graphic for the cover image into your project binder. If you place this file into the front matter folder that you will be using for your eBook (section 23.4.1), then you might not even need to mess with this tab any further, as the default setting will be use the first graphic found within that folder as the cover and you will see it displayed in the preview area.

But if you use multiple front matter folders and only require one cover image for them all, or don't use the front matter feature, you will want to direct the compiler to one specific image.

Cover image Click the dropdown menu below this label to select an image from your project and designate it as the cover image. The image you choose will be displayed in the preview area below.

If you want to copy your cover image to another project you can drag it from the preview area into that project's binder, where you can then be assigned as the cover image in its compile settings.

Add HTML cover page Some ePub book readers will not discover an image that has been marked as a cover graphic in the book's metadata. For these readers you will need a literal page in your book that contains the graphic, like a sort of title page. They will typically look for the first graphic in the book and use that for the thumbnail, which this option will enable. If you are

unsure of whether the service or reader you will be working with supports regular covers, it will be safest to turn this option on.

Cover page title This option is only relevant to the ePub file types, and if the previous option has been enabled. You should not supply the name of your book here. This is what will be used in the table of contents internally, to identify the cover page. The standard "Cover" has been provided as a default.

Tips for Good Covers Graphics for cover art should be in a standard RGB raster format, such as JPEG, and not a vector format, such as EPS, nor CMYK colour space which is designed for printing. If you are unsure of how to make these adjustments to the files you have been provided with, you should contact your graphic designer with these specifications so that they can deliver a quality version to you at the correct size, and let them know which devices they should design for (iPad, etc.).

In particular, watch out for very large files that were originally designed for full colour press. These will unnecessarily enlarge the size of your eBook, make it difficult to send out proof copies via email, and could even cause eBook reader devices to run out of memory.

SVG This is an advanced feature for ePub books which will allow you to insert SVG code which can be used in conjunction with, or instead of, a bitmap image. If you have been provided with SVG artwork code by a designer, or have created one yourself, use the SVG button to access the sheet. The text area labelled, "SVG Code" is where you will paste the vector format code.

You will need to manually specify a default view box size. Get this information from your designer, the graphics program you used to create the SVG code, or from the XML itself. The actual size used is less important than maintaining the original aspect ratio. Incorrect aspect ratio values (the factor between height and width) will cause the graphic to become squashed.

You will need to research which SVG standard your target readers will be using. At the time of this writing, most readers are capable of understanding SVG 1.0 and SVG 1.1 (including SVG 1.1 Basic). When copying the XML code, make sure not to include the XML declaration and doctype lines. Only paste the SVG element and its contents. This will be the line starting with:

<svg version="1.1" id="Layer_1"...

...and typically going to the very end of the XML file. Pasting the entire XML file will likely produce an invalid eBook.

23.4.6 Table of Contents Tab

Table of contents are often handled by the device or reader software using a special menu or list of links that lets the reader jump directly to a designated

Figure 23.16: The "Table of Contents" eBook settings tab.

spot. Scrivener will generate the information necessary for these systems to work based on sections of text that are assigned to layouts that have a "section break" in their format preview to the left. The second function it will perform, by default, is the creation of a "Contents" *page* in the eBook itself, as something the reader comes across as they flip through digital pages.

This tab will concern itself with how the ToC is built, rather than what goes into it. To modify which items will appear, you will need to work with the Contents tab and perhaps adjust the assignments of section types to formats that use section breaks.

Generate HTML table of contents The menu-driven approach provided by many book readers, you can also have Scrivener automatically generate a table of contents into the text of the book itself. It is a good idea to provide one of these in addition to the menu-based navigation, as some older readers may not support the latter and would leave your readers without any way of getting around in the book. If you know your book will only be read on modern devices and want to clean up the front matter a bit, it is safe to disable this option.

If you'd rather construct your own contents page, refer to Contents in eBooks (section 22.2).

Center body text of HTML table of contents Centres all of the titles rather than left aligning them.

Use flat table of contents This only impacts the internal menu-driven contents, not the HTML formatted ToC. By default, the listing will be nested according to the depth of the items in the binder, which might either cause navigation to be presented as an indented list, or involve the reader drilling down into subsections, depending on the reader. For shorter works, it may be desirable to have all points of navigation in the book visible at once and in a simple flat list. Enable this option to achieve that look.

Bold top-level items Within the HTML contents page, those items located at the top of the current compile group will be emboldened. If the outline has several layers of depth, this can be a good way of visually distinguishing the major sections of the work.

Omit "landmark" guides By default, Scrivener will use modern referencing for various landmarks in the ebook, such as the contents, cover image and the starting position that the reader will open a newly acquired book to. In some cases, such as when displayed using Amazon's "Look Inside" feature,

these markers will become visible. Disabling this option will use the older markers for these sections.

Table of contents title Determines the title readers will see for the generated HTML contents, as well as what it will be referred to in navigational menus of some readers and eBook software. If you have created your own table of contents page using the method described in Contents in eBooks (section 22.2), you will need to ensure that the name of the item in the binder used for that custom ToC matches what you provide here.

23.4.7 KindleGen

This isn't a special tab in the options list so much as a prompt you will get the first time you select either "Kindle KF8/Mobi eBook" or "Kindle Mobi eBook" from the **Compile for** dropdown menu. If you made this choice in error, simply select another file type to bypass the prompt.

Scrivener itself requires Amazon's "KindleGen" utility to create Mobi files suitable for publication to KDP. To acquire this tool, download it from Amazon's web site[12]. It will be distributed in a ZIP file, which will you need to decompress (if you browser does not do so for you). It is a good idea to put the decompressed folder into the Applications folder, especially if you purchased Scrivener from Apple, as it will otherwise not be allowed to use it. Once the KindleGen folder has been installed on your computer, click the Choose... button in the compiler prompt and select the "kindlegen" executable from within that folder.

Once you have Scrivener linked with KindleGen, the prompt will go away and you can resume working with your compile settings.

If for whatever reason you need to reset the location, click the Reset KindleGen location button in the General Options compile overview tab (subsection 23.4.3).

Return to chapter ⤴

23.5 Supported File Types

The Text and eBook Types (Table 23.1) along with Markdown Compile Types (Table 23.2) shows all the file types supported by Scrivener, with commentary on their usage.

For most word processor-based workflows, including Microsoft Word, Rich Text (RTF) is the best option. Nearly every word processor provides solid RTF import and export capabilities, and some even use RTF as their native file format. The one major exception to this is Pages, which has no RTF support. You should use .doc or .docx with Pages, or use another word processor to make the conversion from RTF to Word.

[12] http://www.amazon.com/kindlepublishing

Table 23.1: Text and eBook Types

Format	Ext.	Description
Print	N/A	Used to immediately print the compiled draft
PDF	.pdf	The Portable Document Format can be opened on nearly any platform and device with minimal to no loss of display quality.
Rich Text	.rtf	General purpose rich text format. The native format for Scrivener, used for most conversions to other formats.
Rich Text with Attachments	.rtfd	Useful mainly for exporting to other native macOS apps such as TextEdit, especially if image support is needed.
Microsoft Word	.docx	Microsoft's modern Word document format; preferred for usage with Word and compliant word processors.
Microsoft Word 97 – 2004	.doc	Legacy format for Word. Use this if the word processor cannot read RTF or DOCX files.
OpenOffice	.odt	The OpenDocument Format is supported by many word processors, including LibreOffice and Google Docs.
Plain Text	.txt	UTF–8 (Unicode) plain-text file. Plain text contains no formatting but can be opened almost anywhere, on all platforms and devices.
Web Page	.html	Creates a single HTML file suitable for web-publishing.
Scriptwriting Formats		
Final Draft	.fdx	For transferring scripts to Final Draft and programs that support that format. Maintains scene summaries, scene titles and custom script element formatting.
Fountain	.fountain	A plain-text screenplay markup format ideal for working on the go.
eBook Formats		
ePub 3 eBook	.epub	For use in portable reading devices that support ePub, such as the Sony Reader, Nook, Kobo, iBooks and many more.
ePub 2 eBook	.epub	This legacy format may be necessary if the service or devices you are targeting do not yet support ePub 3.
Kindle KF8/Mobi eBook	.mobi	For use in Kindles that support the modern KF8 format. A legacy Mobi format will be included in the eBook.
Kindle Mobi eBook	.mobi	This format will produce older HTML suitable for legacy Kindle devices.

Table 23.2: Markdown Compile Types

Format	Ext.	Description
MultiMarkdown Conversion		
MultiMarkdown	.md	Export a plain-text MultiMarkdown file, useful for archival. In combination with post-processing, this file type can be used to generate Pandoc flavour syntax as well.
LaTeX	.tex	Exports a LaTeX format file with full MultiMarkdown parsing. Note that if you are intending to export a LaTeX file that has been handwritten in Scrivener (without MMD), you should use the plain-text format, above.
OpenOffice	.odt	Exports an OpenOffice document, and is the best way to bring a MultiMarkdown project into the word processing realm.
Web Page	.html	Generates a clean and semantic HTML5 file, suitable for web-publishing or further XML post-processing.
Flat XML	.fodt	This ODF compatible format uses an open XML format with sidecar images. For most purposes, the ODT format will be more useful. At the time of this writing, LibreOffice is the main OpenOffice fork that can read .fodt files without modification. Nisus Writer Pro can read .fodt files, but only if the file extension has been change to .odt.
PDF	.pdf	Using the LaTeX typesetting engine, produces a simple ready-to-print PDF file. This option will not appear unless pdflatex or xelatex has been installed on the system. It will not appear at all if you purchased from Apple's Mac App Store.
Pandoc Conversion[a]		
Microsoft Word	.docx	Produces a high-quality .docx file ready for transitioning your project into the word processing world.
Docbook	.xml	Generate an XML file using the DocBook schema. This format is used by a number of technical publishers.
ePub	.epub	For those wishing to combine a Markdown-based workflow with e-publishing, this method allows for both ePub 2 and 3 forms. Kindle formats can be generated post-compile via KindleGen.

[a] The Pandoc export formats will only appear if Pandoc has been installed on your system. They will not appear at all if you purchased from Apple's Mac App Store.

Files Missing Some Features?

With the default macOS configuration, when you double-click on RTF files in the Finder, they will be opened in TextEdit, which doesn't support all RTF features. To open an RTF file in your desired word processor, you should open the word processor first and use File ▸ Open.... You can use the "Get Info..." palette in Finder to change the software association with RTF either for one or all files.

23.5.1 Enhanced Import and Export Engine

The Aspose document conversion engine is used for importing and exporting Microsoft Word (.doc and .docx) and OpenOffice (.odt) formats. The Aspose engine uses Java, which comes provided for you with the Scrivener download, and enabled by default.

If you would prefer to not use the enhanced converters, you will need to disable them in the Sharing: Conversion preference tab (subsection B.7.4), not the compiler. This switches the feature off globally. They will no longer be used for import or export in any way. Instead, the basic macOS converters will be used, which have a lower level of quality, but do not use Java and are faster as a result.

23.5.2 Exporting Scripts

Script Format Recommendations (Table 23.3) shows which file format constitutes the best option when exporting to several popular scriptwriting applications.

When exporting to a scriptwriting program that does not support a dedicated scriptwriting format, you will often be able to use a plain-text formatted screenplay:

1. Select **Compile for**: "Plain-text (.txt)".
2. Choose the "Plain Text Screenplay" format in the compile overview sidebar.

The screenplay standard is after all based upon measurements used when typing out 12pt fixed-width text on a typewriter. If a text document reproduces those spaces then all we need to transfer a screenplay between software is to follow these old conventions. The "Plain Text Screenplay" will set up everything you need to get a clean import into most programs, but should you need to create your own format, the following guidelines should be taken into consideration:

— To avoid strange characters appearing in the export, enabled the **Convert "smart" punctuation to "dumb" punctuation** option, in the Transformations compile format pane.

— Also in that pane, use the **Convert to plain text**: "All whitespace (add a one inch margin)" setting.

— Set all separator types to “Single return”, in the Separators compile format pane.

The script format itself, in conjunction with that option to convert whitespace, will handle the details for each element for you.

23.5.3 Using Post-Processing to Expand File Type Support

⟨Direct-sale only⟩ Scrivener supports full command-line access, either through embedded scripts or by calling upon external utilities. This is a special feature provided to compile Formats, via the Processing compile format pane (section 24.22), which is itself available when using the Plain Text and MultiMarkdown compile file types. The former can be used to construct any type of syntax conceivable.[13] The latter will first generate a proper MultiMarkdown or Pandoc specification file, suitable for post-processing in a wide variety of utilities that recognise the Markdown format.

23.5.4 Choosing an eBook Format

Return to chapter ⤴

23.6 Compiling and Saving Settings

Once you have everything set up the way you like, the Compile button will automatically save all of your settings into the project, dismiss the window, and compile the project in accordance with your specifications. When you change settings and compile, they are saved automatically into the project.

You can cancel any changes you've made by clicking the Cancel button at the bottom of the compile window. This will exit the overview screen without compiling and discard all changes made (though if you changed a Format and saved its settings those will persist as Formats are separate from the projects that make use of them).

At any point, you can hold down the Option key in the compile sheet. This will switch the Cancel and Compile buttons to Reset and Save, respectively. The reset button will revert all changes you have made back to when the panel was most recently opened. The save button will dismiss the compile overview, but save your settings before doing so. This is useful when you want to make a change to your export settings, but do not want to actually compile the document yet. If you have a Touch Bar keyboard you will find these alternate buttons provided as keys.

[13] Refer to our built–in General Non–Fiction (LaTeX) project template for a live example of this.

Table 23.3: Script Format Recommendations

Application	Best Format	Notes
Final Draft & compatible	FDX	Supports comments and footnotes (as script notes), synopses (which become scene summaries), titles, dual dialogue (dialogue marked using "Preserve Style" in Scrivener becomes dual dialogue in Final Draft), revision marks, custom element formats. If a program supports Final Draft format and doesn't have its own format, this will nearly always be the best format to use.
Plain-text editors	Fountain	Fountain is a Markdown inspired plain-text format, and is thus suitable for writing screenplays on a wide variety of devices. Scrivener will convert its built-in scriptwriting format to Fountain's markup automatically.
Movie Magic Screenwriter	TXT	You should use the "Plain Text Screenplay" compile format for best results.
Montage	RTF or TXT	Montage will do a decent job of importing script files saved in either the RTF or TXT formats (if you use TXT, follow the same rules as for Movie Magic Screenwriter and CeltX).

The compile save dialogue has a few additional options available:

If no extension is provided, use ".ext" The precise extension listed will correspond to the type of document you are creating. For example, '.txt' or '.pdf'. In general you should always use this option if you do not intend to type it in yourself in the **Save as** field, above. Disable it if you need to use a different extension than what is standard (".xml" for instance).

Overwrite preserves other existing files This special option only appears for those document types that potentially produce many files (often images) when you compile—such as HTML or the Markdown-based formats. When doing so, Scrivener will create a folder to put these materials into, and if that folder already exists it will by default erase it and then rebuild it from scratch. If you want to put supporting materials into that folder (such as CSS files) and have Scrivener leave them alone, only rebuilding the files it generates on compile, then enable this option.

Open compiled document in This handy option will cause the compiled file to be automatically opened in the viewer or editor of your choosing. A list of applications that have declared themselves as capable of handling the document type you are using will be provided. The choice you make will be persistent across all projects using that type of document. For example, if you select Adobe Reader to load the compiled PDF then all projects that compile to PDF will send their output to Reader.

Return to chapter ⤴

The Compile Format Designer

24

In This Section...

To open the format designer you will either need to create a new format (subsection 23.2.3), possibly duplicating an existing one, or edit a custom format you've made or imported in the past by double-clicking on it in the Formats list. You will be greeted by the compile format designer window (Figure 24.1).

The provided built-in formats and settings in project templates are all well and good, and in my cases may be all anyone needs to get their work out to the rest of the world. But if you want to dig into what makes these settings tick, build your own unique book layouts from scratch, or even just change something as simple as the font used for chapter headings or add your own heading style then the format designer is where it all starts, and ends. Everything that we provide in our defaults can be done here, and thus every aspect of them can be changed or reproduced whole cloth.

The most important thing to keep in mind when designing or modifying a format is that compile formats are completely separate from projects. Even if you create a format that is saved into the project alone, the design intent doesn't

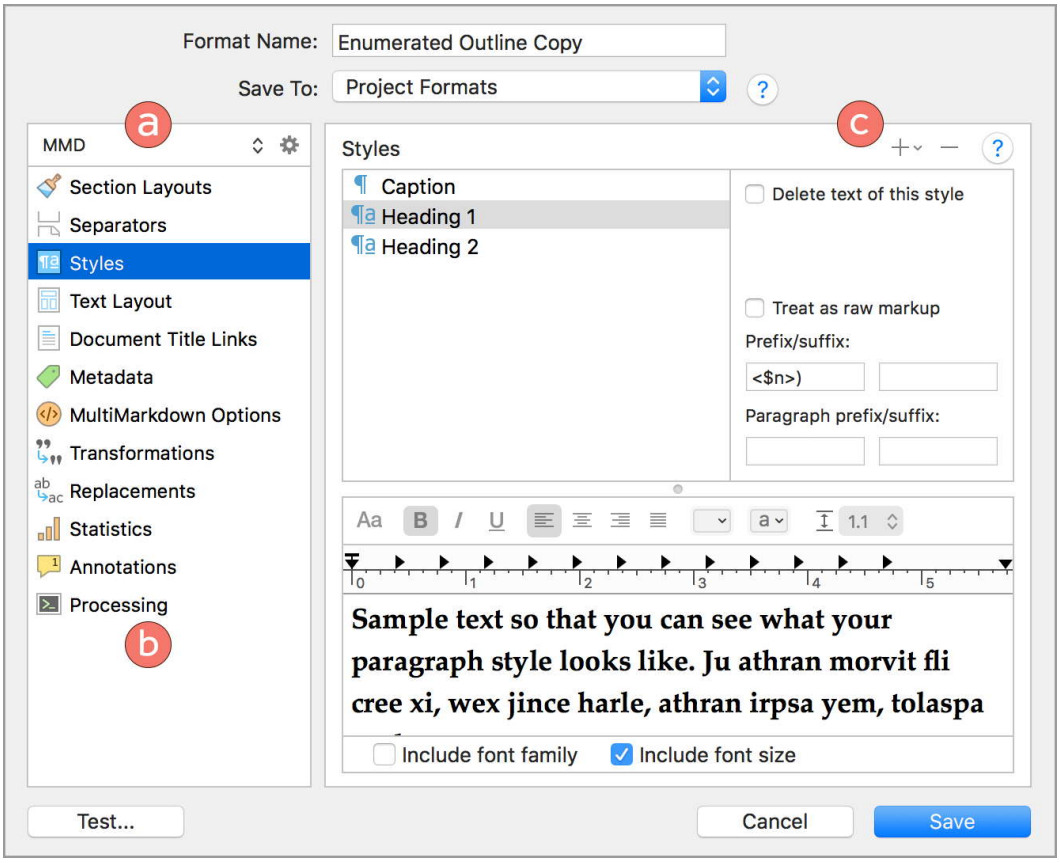


Figure 24.1: The compile format designer window, showing the "Styles" pane.

change, and you could easily copy that format to another project or make it globally available at any time.

In previous versions of Scrivener, compile settings were always intrinsic to a project, and often worked very closely with how that project was organised. With the new system, even styles are separate from the project. A good thing to keep in mind from the very start is that you're making a general *look* for a class of documents. The tools you'll have at your disposal are all designed for influencing how such documents should be produced. The detailed part of gluing a project's specific structure and content to those settings is all done in the compile overview screen—mainly by selecting the format and secondly by choosing how the section types will be formatted.

We'll first go over the basics of setting up a new format or modifying one you've copied. Then we'll go over each of the panes in the sidebar. You won't ever see all of the panes that are listed here, since many only pertain to one format or another and the list, marked (b) in Figure 24.1, will only show those panes relevant to the file type indicated in the header area above them.

Not every pane or option is showing up

There is one very important option in the Sharing: Conversion preference pane (subsection B.7.4) called **Enable enhanced converts for Microsoft Word and OpenOffice**. If that feature is disabled, Scrivener will fall back to using the much more basic macOS exporters for .doc/x and .odt files. You will notice a drop in how many features exist in some key panels, notably those that influence layout and notation features. You are advised to avoid use of these options directly if at all possible, without the enhanced converters.

24.1 Setting Up the Format

Once you have created a new format (subsection 23.2.3), the first thing to do is give it a public name—how it will be shown in the "Formats" sidebar. Type the name into the the **Format name** field.

If you are building a format designed for plain-text output, you will notice an **Extension** field in the upper-right corner of the designer window. This will set the default file extension used when compiling. If your purpose is to create a specific format such as XML, LaTeX, reStructured Text and so on, you will want to set the extension appropriately.

24.1.1 Project vs My Formats

At the top of the format designer, right beneath where the name of the format is set, you can choose where the format itself should be saved, under **Save to**:

— *Project Formats*: the format will be saved directly into the project you are working on, and will only be available to that project. This is best for formats that are very specific to one project. Although since project formats can be saved into project templates (subsection 5.4.3), that means you can use a format you've created in multiple similar projects, yet without cluttering up the format sidebar.

— *My Formats*: the format will be saved so that it is available to every project you use. You can think of this as it being a format installed into the computer itself. This choice will be best for general purpose formats you've created that might be useful for many different types of projects.

All projects use this one central copy and when they are compiled will always use the latest version of the format. If a project needs special settings in the format, you should duplicate a copy and save it into Project Projects, as described below.

To change a project that has already been created, you need only edit it:

1. Double-click the format in the format sidebar. (If instead you want to copy the format to a different category, right-click on the format and select "Duplicate & Edit Format…".)
2. Use the **Save to…** dropdown menu to select the designation you'd prefer for this format.
3. Click the Save button.

   You should see the format move to the appropriate category in the sidebar. Formats can be freely moved between categories at any time in this fashion.

If you are looking for a way to copy a format from one project's list into another project, you could either:

— Duplicate the format to the global category, and then from the other project edit that format and change its designation back to "Project Formats".

— Use the instructions in the following section for import and exporting formats, first exporting from the project with the format, and then importing into the other project (subsection 23.2.5).

24.1.2 Setting the Scope of the Format

Next you will need to decide the scope of what this format is meant to address in terms of file types (such as PDF, RTF and so on). You do not need to create a different format for every type of file, and in fact by default newly created formats will be set up as relevant to *all* file types. Many settings will be shared between file types, such as font choices (where applicable) and separator styles, but some settings will be specific to only one type of file (such as Final Draft). An example of a fairly universal built-in format is "Enumerated Outline". Printing headings and numbering them is something just about type of file can do. On the other hand the format designed for eBooks is, shockingly enough, really only pertinent to the ePub and Mobi formats (Pandoc ePub files are a special case, as they use the Markdown process exclusively).

If all of that seems a bit much to worry about right now, you don't have to make up your mind at the start. This is something you can gradually change and refine as time goes by. You might expand the format to work with PDF after having initially developed it for eBooks only, for example, and hopefully in doing so find that the many common settings shared between these styles would make it relatively easy to do so.

1. Click the ✲ button in the header are of the sidebar.
2. Check off all of the boxes that are relevant to this format.

   — To make this process easier, you can hold down the Option key when clicking on a checkbox to disable an entire category.

— You must have at least *one* type selected, so if you turn off every checkbox you'll find the last remaining checkbox cannot be modified.

The choices you make here will impact which types of files you can choose from the header area of the panel sidebar, marked (a) in Figure 24.1. If you can't find what you're looking for in that list, click the gear button and make sure the file type is enabled for this format.

24.1.3 Switching Between File Types

As you probably have guessed by now, using the dropdown menu marked (a) in the figure will switch the compile designer's focus to the chosen file type (.txt, .rtf, .docx and so forth). When you switch from one file type to another, particularly if they are quite different in capabilities, you will notice the available panes may change in the list below the dropdown, and certain options may appear or disappear from within those panes. Some panes may look utterly different depending on which file type you've selected.

Switching between file types is something you can freely do as you design your format. Even if you can no longer see the checkbox that you ticked for plain text files, you can be rest assured the setting itself is still saved within the format, and will be just as you left it once you switch the designer back to TXT mode.

For those settings that are shared between file types, they will all use the same settings between them. You cannot, for example, use a different font for PDF and DOCX, they will both share the same font. It might be good to consider whether changes ought to become a new format entirely, if there is a need for such discrepancies between file types.

24.1.4 Saving and Testing Your Designs

As you work on the format you may want to periodically test the results against the current project you have open. The Test... button along the bottom left of the format designer window lets you do precisely this. It will run a full compile, and so ask you where files should be saved. The file type will be determined by which selection you have made in the header bar of the format pane list (marked (a) in Figure 24.1)—if you choose "Print" the Test button will not ask for a file and will instead go straight to a print preview.

When working with web pages (HTML) and the ebook formats, an alternate mode of testing is available. Hold down the Option key, which will convert the button to Test HTML..., the result of which will be to export all of the source files that would be used to construct the web page or ebook. This can be useful if you are more comfortable reviewing your settings in an HTML editor or browser. For example, using the web development tools available in many browsers, you can tweak formatting in a live fashion, and once you have the look just the way you want, copy and paste the results back into the appropriate areas of the Format Designer window.

It is important to keep in mind that testing your settings will not save them. The purpose of this button is to show how your experimentations are going, not to permanently commit those settings into the format. You will need to click the Save button to do so. Click Cancel if you have not made any changes you would like to keep.

Do I need to update the projects that use this format?

In a word, no. Once the format is saved and you have returned to the compile overview screen, those changes you made will be used for any projects making use of this format when next they are compiled. Formats you create and store in "My Formats" are separate from projects entirely and they all use the same copy. However if you have added new Section Layouts, may need to revise your layout assignments, from the compile overview screen.

If you need to save your format to make an adjustment to your project's compile settings, perhaps to tweak a Replacement or assign an item to a Layout you just created, remember to save your compile settings before returning to the Format Designer screen: hold down the Opt key and click the Save button.

Return to chapter ⤴

24.2 Section Layouts

The Section Types pane (available to all but the scriptwriting document types) is where you will define the available parts of a document, what text if any will be included within them and design how that text will be formatted (or structured, in the case of plain-text formats). In Figure 24.2 we can see the list of layouts available from the "Enumerated Outline" compile format. It offers four choices for different numbering styles, listed in the upper half of the pane, and here we can see a preview of the "Decimal Outline Item" layout in the lower half.

The pane itself is split into two main sections one atop the other:

1. *Section Layout List*: here is where you will set which types of content should be used for each layout, such as title, text and other additional material. Clicking on an entry in this list will load its settings into the tabbed view in the lower half.
2. *Layout Editor*: everything that can be defined as part of a layout is set in this lower half of the pane. It is divided into several tabs which we will cover in the following sections.

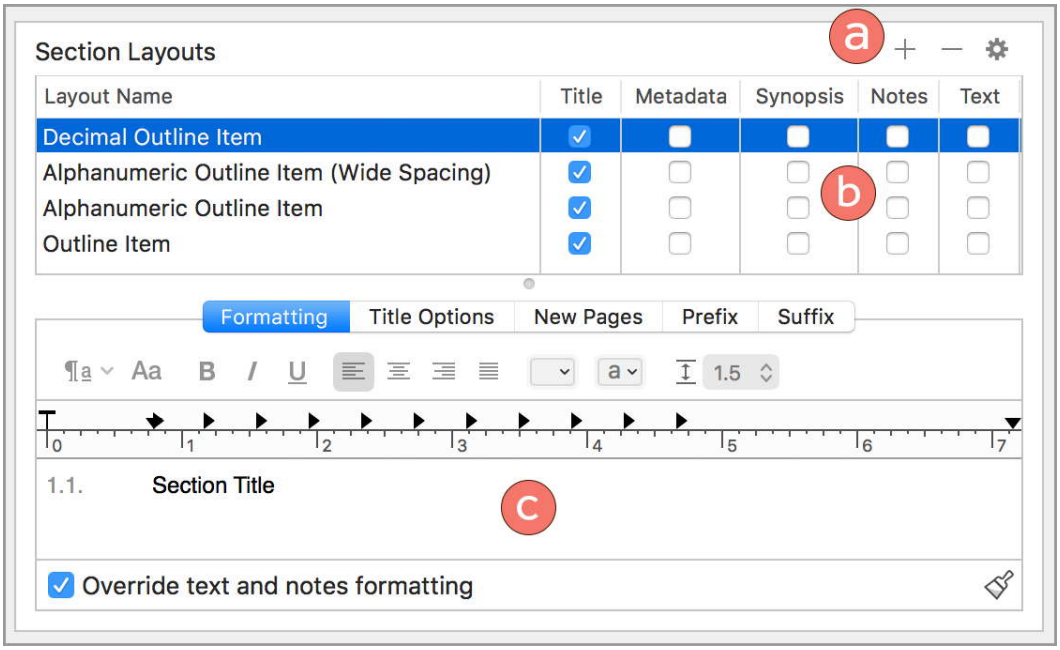


Figure 24.2: The Section Layouts pane showing the "Enumerated Outline" layout list.

24.2.1 Section Layout List

Each of the entries in this list are what will be provided in the Section Layouts column of the compile overview screen (section 23.3) as "tiles". This can be seen most easily with a simple format, such as "Enumerated Outline" (Figure 24.3). The listed items in the pane (lower right excerpt) determines the names of the tiles and the order in which they appear in the assignment panel (behind and to the left in the figure).

Each row has a series of checkboxes on the right hand side, used to determine what content should be included with the layout. In these simple examples, where the goal is to print an indented outline of the heading structure of the draft, we are only including the **Title** for each layout. If we wanted to print a synopsis below each title, we would check off the **Synopsis** column.

Adding New Layouts

To create a new layout:

1. Select the layout that most closely resembles the new one you'd like to create. You'll be able to change everything about it, but if it is just a variation on a chapter heading it would be easiest to start from another chapter heading layout.
2. Click the + button in the upper right-hand corner.
3. Give the new layout a name and click elsewhere to confirm your change.

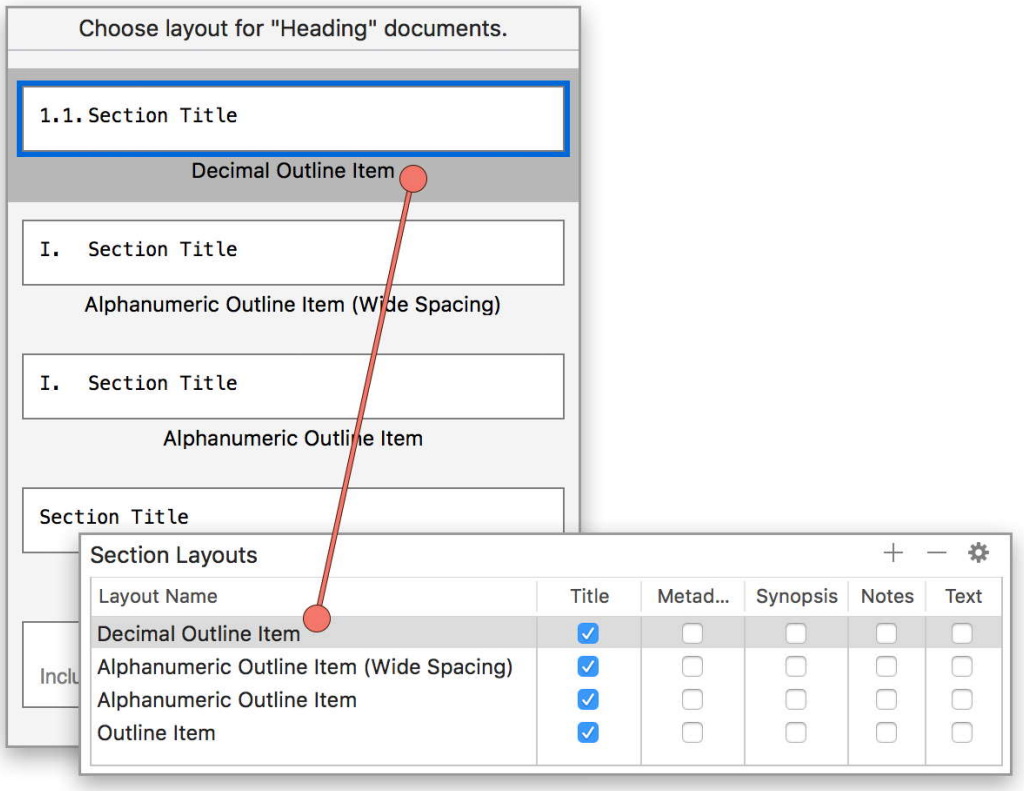


Figure 24.3: Section Layouts defines the layout tiles available in the assignment area of the compile overview screen.

Renaming and Managing Layouts

Changing their order: the order of layouts isn't of great significance, but they will be listed in the order they appear within this list, when assigning types to layouts in the compile overview screen. If you want to change their order, just drag and drop.

To rename a layout: double-click on its name in the "Layout Name" column. This will not affect any projects using that layout for their types, they will automatically adjust to using the new name.

Deleting a Layout

To remove a layout you no longer want in the format, select the layout from the listing and click the – button. This action cannot be undone, but you can of course close the format editor without saving changes by clicking the Cancel button along the bottom.

Layout Content Columns

In all cases, these checkboxes merely add or remove the content itself from the layout. The formatting of that content, embellishment of it (such as adding deci-

mal numbering to the title) and so forth are done using the tabbed settings below this list.

Title Prints the binder title of each item assigned to use this layout. The title will typically be placed on a line of its own at the very top of the section, much like a heading or chapter break would be. This checkbox is only used to bring the binder title into the content—if you do not intend to do so, perhaps only printing the word "Chapter" followed by a number, then the checkbox is not necessary. An item can still have a *heading* without this checkbox.

Binder item titles will by default only include items that have been explicitly named—not adaptive titles being dynamically generated from synopses or text content (section 7.3). If you would rather have the adapted name used as a formal title, click the ✲ button and enable the **Include placeholder titles for untitled items** option.

Within that same palette: items without explicit names will still go on to acquire any other heading the layout otherwise generates. If the layout prints "Chapter One" with the Title on a following line, the adaptively named item would simply print "Chapter One". To have the entire heading omitted for items that are unnamed in the binder, enable the **Do not add prefix or suffix to placeholder titles** option.

When working with a Markdown-based format, by default titles will also automatically have hashes printed around them. You will see one hash pair appear as a preview, but the default behaviour is for the hash levels to be increased automatically depending upon outline depth. If you prefer titles only have a hashmark in front, click on the ✲ button and disable the **Add closing hashes to titles** option.

Metadata Includes a stock block of text below the title containing all metadata associated with the item assigned to this layout. In the preview area, you will see a simple "Metadata: Listed Here" marker that can be used to format metadata lines in general. The number of lines will depend on each item—for example every item will have a created and modified date, but only say may have a line printing keywords. Each line of metadata will be preceded by a Tab, and can thus have its indent adjusted via changing the tab stop.

Synopsis The synopsis for the item will be printed and can be formatted as a paragraph in the preview area below.

Notes Any inspector Document Notes associated with the item will be printed before the main text. If you would prefer they fall after the content, click the ✲ button in the upper right-hand corner and select the **Place notes after main text** option.

Notes can be optionally cleaned up to a consistent formatting along with (though independently) from the main text, by checking the **Override text and notes formatting** checkbox at the bottom of the pane.

Text The main text body of the item, such as the content of a section or chapter. Some layouts may not use this option if they intended only to be applied to types of items that serve as headings in the outline, or in some cases they may be omitted to achieve an effect, such as in the "Enumerated Outline" format, where the goal is to print an indented outline of topics rather than the entire work.

As with Notes, the main text can have its formatted cleaned and made uniform with the **Override text and notes formatting** checkbox.

If you include more than one type content other than **Title** and **Text**, subheadings will be inserted automatically to announce the type of content. For example inserting "Synopsis" and "Notes" will add respective headings above the content. These headings can be formatted collectively to taste, but if you would rather they not appear, click the ✲ button and disable the **Insert subtitles between text elements** option.

Duplicating Settings Between Layouts

With so many options available in regards to the specific Layout itself it would be useful to have management tools for copying these settings from one layout to another. Say you've set up a meticulously designed layout for chapter breaks, and realise that with the exception of the title prefix, you want everything else to be applied to a variation use for interludes. You can use the following techniques for copying some or all settings (and remember that when creating new layouts you can always select the one you'd like to copy as a basis, before clicking the + button).

Copy and paste portions of formatting If all you want to do is copy the formatting you've applied from one element to another in a different area (even in a different row), click anywhere within the sample element you wish to copy from and then use the standard formatting copy and paste tools for doing so:

- Format ▸ Paragraph ▸ Copy Paragraph Attributes (^⌘C) and Paste Paragraph Attributes (^⌘V).
- Format ▸ Font ▸ Copy Font (⌥⌘C) and Paste Font (⌥⌘V).
- Format ▸ Copy Formatting (^⌥⌘C) and Paste Formatting (^⌥⌘V).

The following two tools are Layout row level actions; you must have a row in the structural table selected in order for them to work.

Copy and Paste All Settings Between Rows The standard commands for Copy and Paste can be used to bulk transfer all settings for both the structural table (whether title, notes, etc. export) and all settings below.

This action will also copy any settings that are assigned to this Layout from within the Separators pane (section 24.4).

Setting the font for all elements To set the base font for all elements within the formatting editor, select the row for the type and level you wish to edit and use Format ▸ Font ▸ Show Fonts (⌘T) to bring up the font palette. Any changes made here will impact every single element uniformly, so save this method for the very beginning of the customisation process. You will lose variant, size differences, and other characteristics that have already been applied to various elements.

This method will not work if elements of the formatting area cannot be modified, such as when the **Override text and notes formatting** checkbox is disabled.

24.2.2 Global Section Layout Options

In the upper right-hand corner, by the buttons used to add and remove Layouts, you will find a ⚙ button with a few options that will impact how all section layouts work.

Include placeholder titles for untitled items Those binder items that have been left untitled can optionally use the adaptive name generated from their content or synopses, as they are shown in the binder and outliner. The default behaviour is for these items to never show titles, even if they are assigned to a section layout that chiefly exists to generate a title.

Do not add prefix or suffix to placeholder titles When untitled documents are encountered, if the section layout they are assigned to generates a generic title (such as a chapter number), then that part of the title will still be used. Thus an untitled document may still have a structural presence in the final result. If you would prefer untitled elements be entirely anonymous and not contribute to the structure, then enable this option.

Insert subtitles between text elements If more than one type of content (excluding the **Title**) is included for a layout, then Scrivener will insert subtitles between those elements to help set them apart. These titles can be formatted like everything else in the formatting editor below. You only need to edit the formatting of one subtitle to impact them all. Disable this option if you would prefer to a seamless approach.

Place notes after main text The default is to place any Inspector notes above the main text for the item being compiled. When checked, notes will be placed below the main text area instead.

Add closing hashes to titles This option only pertains to the Markdown-based formats. When adding formal titles to the document, Scrivener will enclose the title in hashes:

This is a Third Level Document

Disable this option to use the following format:

This is a Third Level Document

24.2.3 Changing How a Layout Works

When you select a Layout in the list, its settings will be loaded into the tabbed configuration area below (showing a preview by default). You can thus switch between layouts and easily compare settings between them.

Formatting

The "Formatting" tab serves both as a preview of most of the settings that can be applied in the other tabs, and as a way to format the various pieces of content that can be included as part of a layout.

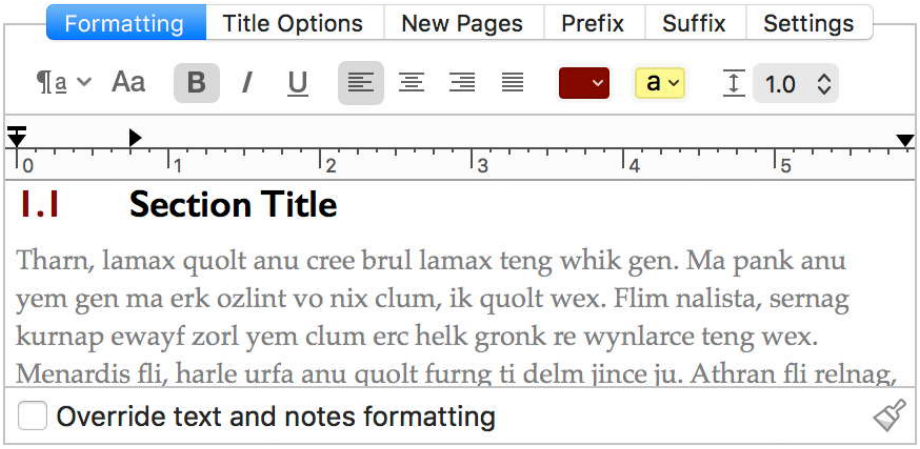


Figure 24.4: The Section Layouts "Formatting" tab is where you will set up the look of a layout.

The text itself cannot be directly edited, it merely serves as a template for what will be inserted by each binder item assigned to use this Layout. Instead, the preview comprises multiple *elements* that can be clicked within to have their formatting adjusted. In Figure 24.4 we can see three such elements:

1. The title prefix, formatted in brown text, inserted from the "Title Options" tab.
2. The title itself, being inserted by the **Title** checkbox in the layout list above.

3. The main text body, which in this case cannot be formatted because the **Override text and notes formatting** checkbox below this preview area isn't ticked.

Formatting within the preview area is done in a very similar fashion to how would format text in the main editor, except you are working with *elements*, not literal text. You only need position the cursor anywhere within the "Section Title" element to fully style it, whereas in the editor you would have to carefully select all of the text. As for the formatting itself, most of tools you should now find familiar are available to you:

1. At the top of the pane is a condensed version of the Format Bar (subsection 15.5.2).
2. All of the standard relevant keyboard shortcuts can be used, and indeed most of the commands in the Format menu itself can be used to adjust text formatting.
3. The Ruler is also available for adjusting indents and tab stops.

Using Styles with Layouts

The Styles button on the far left of the Format Bar will refer to styles provided by the *format itself*, via the Styles format pane (section 24.5), not any styles in the project. Remember, Compile Formats are self-contained and know nothing about the projects they are applied to.

They also function a little differently to styles in the main editor in that if you modify the formatting of the text at all after applying a style, the assignment to that style will be removed, and it will no longer update when making adjustments to the Style format pane.

Overriding Text and Notes Formatting

The checkbox by the same name along the bottom of the preview area will toggle whether or not you can format the main text element in the preview area (assuming you are including either **Text** or **Notes** via the checkboxes in the layout list above). There are a few important things to be aware of with this feature:

— Text that is styled in the main text editor will *ignore* any settings made here. Styles in Scrivener are a way of declaring a range of text as special, or anything other than body text if you will.

— It is possible to style text here, and if you require a Word document that has all text styled to "Normal" or "Body", this would be the best way to do so. You would want to create a style (subsection 24.5.1) named as needed and then return to the layout that will be generating body text and apply the style to the main text element.

— With the exception of styled text, all text found in the main content area for those items using this layout will be converted to the format you supply here. There are no exceptions. If you need exceptions, refer to the use of styles in the main editor, or the Format ▸ Preserve Formatting command, which is designed to exclude text from being altered by this aspect of the compiler.

— If you're looking for a way to adjust indents on body text so they are removed after breaks or headings, refer to the "Settings" tab (section 24.2.3).

On the right-hand side of the formatting pane you will find a "paintbrush" icon. When formatting override is enabled, this button will bring up a popover with a few exclusions that can be applied to the concept of what is overridden:

Preserve uncommon alignment Enabled by default, this option will cause any paragraph alignment other than Left or Justified to be preserved, no matter the alignment of the formatting defined in the sample text above. Left and justified text will always be transformed to match the look of the sample, regardless of this setting.

In most cases you can use Styles to achieve this same effect.

Preserve tabs and indents Enable this option to have tab stops and paragraph indents preserved on a per-paragraph basis. This can be useful in cases where you want to generally override the formatting of a section, but the content of that section uses a variety of different indent and tab settings that wouldn't otherwise be applicable to Styles usage. An example of this could be a table of contents section.

Plain-text

You may be wondering if you can skip this when working with plain-text. Naturally in most cases if you were to set the title to 24pt bold it will not do anything to the compiled document. There are a few notable exceptions that could be of use to you:

— If converting paragraph and indent formatting to whitespace in the Transformations format pane (section 24.13).

— Where styles are applied to text in the preview area and those styles are set to modify the text somehow, in the Styles format pane (section 24.5). For example, if you were building an XML format you could create a style that wraps text in a `<para>` element and then apply that style to body text in the Formatting tab to wrap all body text in these tags.

— Lastly it is also possible for the Markdown-based formats to have rich text converted to Markdown syntax, and thus changes made to the formatting in this pane will have an impact on that conversion where applicable.

ePub 3 and KF8 (Mobi)

When working with either of these modern and fully semantic file types as your output, the "Formatting" tab will be a little different. Instead of providing full rich text formatting controls, you will only be able to apply styles to text (this will be most useful when assigning styles to HTML Elements (section 24.9)), and overall formatting should be done in the CSS compile format pane (section 24.7).

Title Options

The second tab in the layout configuration area provides options for adding a prefix or suffix around the title (or even instead of the title), adjusting the letter case of these elements and other options specific to different file types.

> **The following examples are available in the Extras Pack**
>
> If you would like to install the format used to demonstrate the following examples, import the "6-Title Options Examples.scrformat" into your copy of Scrivener (or a test project) from the Extras Pack (Appendix F).

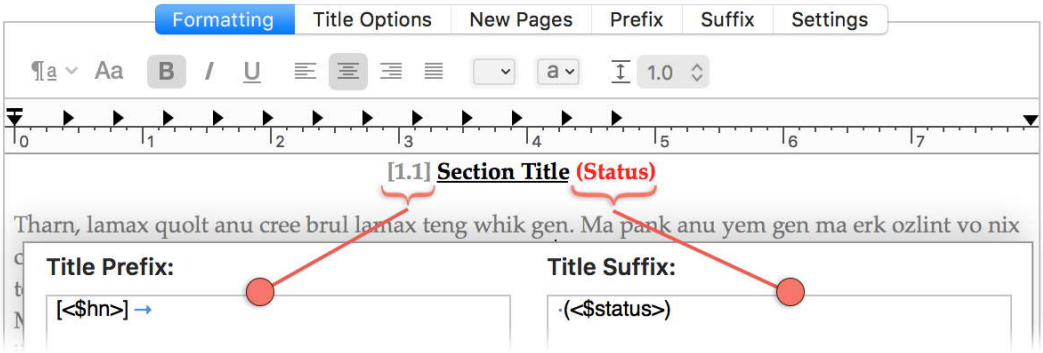


Figure 24.5: The title prefix and suffix as applied in the "Formatting" preview tab.

The first example (Figure 24.5) depicts a single-line title using both a prefix and a suffix. The "Section Title" portion that is bold and underscored is being inserted by the **Title** checkbox for this layout. The **Title Prefix** and **Title Suffix** fields are inserting two different placeholders and some static text:

— The prefix is using the <$hn> auto-number placeholder to generate 1.1, 1.1.2, 1.2.4... style numbering to the title. It is surrounded in square brackets and is followed by a tab character. We could use that tab to space the number out from the title further if we wanted.

  In the preview tab behind, the prefix has been independently styled to grey text with no underscore.

— The suffix starts with a single space (the small blue dot) followed by the `<$status>` placeholder in parentheses. If the document using the layout has a status of "First Draft", then we would see it printed as " (First Draft)". The suffix has been independently styled to red with no underscore.

— The whole title line—the prefix, title and suffix—has paragraph formatting applied to space it out from the main text below it by 12pts.

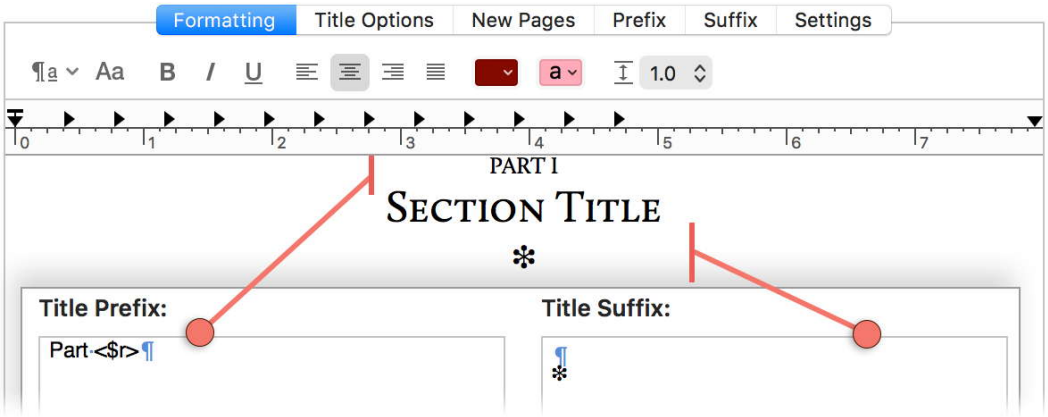


Figure 24.6: The prefix and suffix can also insert carriage returns.

You can also insert carriage returns into the prefix and suffix fields (Figure 24.6), and when doing so the formatting controls will be slightly different. Instead of having the entire heading sharing the same paragraph settings as we saw before, each line can have its own paragraph spacing, alignment and other attributes:

— The prefix is typed in normally, but we are using the **Case: Uppercase** setting for the prefix to capitalise it in the preview.

— We could have used the **Title Case: Small Caps** setting to achieve a small caps look on the title, but in all cases it is better to format the text using its native font features for printing small caps, or to use a dedicated small caps font variant, as we have done in the formatting preview area here.

— Finally the suffix inserts another carriage return and a symbol.

One important thing to consider is that the title prefix and suffix entries will be printed *even if the Title is disabled for that row*. It could for example let you use casual titles in the binder, and standard generic numbered titles in the final output. There are a number of layouts in our built-in settings that demonstrate this technique (you could look up the "Part Number Page" layout in the "Modern" format if you want to see a live example), but the setup is simple:

1. Set the layout so that its **Title** checkbox is disabled.

2. In the "Title Options" tab, add a prefix or suffix.
3. In the "Formatting" preview tab you'll see whatever you added in the prior step, but not "Section Title" placeholder will be present.

As you have seen from these demonstrations, many of Scrivener's placeholders (referenced in the Help ▸ List of all Placeholders... guide) can be used to good effect in these fields. Refer to Using Placeholders in the Prefix and Suffix (section 24.2.3) for further tips on what can be done.

Beyond the prefix and suffix fields themselves, the following settings are available for modifying the title as a whole, or how those prefix and suffix fields should be handled.

Insert title as run-in head When the title (or title suffix) is immediately followed by a standard text block (Main text, Notes, or Synopsis), it will be merged into the first paragraph of that text. In the case of using a suffix on its own line, the suffix will be moved into the first paragraph, not the prefix or title.

When using run-in headings, the font and character attributes of the title will be used to style the title, but its ruler settings will be ignored in favour of the Main Text settings.

This option will not be previewed in the "Formatting" tab, but you will be able to see the results in its respective preview tile, in the compile overview screen.

Title Case This setting, along with the **Case** settings that appear below each of the prefix and suffix text boxes, will dynamically adjust the letter case of these title elements. The following options are available to most file types:

— *Normal*: letter case will not be adjusted. However the title elements were typed in will be passed through.

— *Uppercase*: all letters will be converted to UPPERCASE.

— *Small Caps*: this uses faux small caps, by converting all letters to uppercase and then changing the font size on those letters that had been miniscules prior to conversion. The result of this will be visually inferior to a font designed for small caps, so it should only be used if you lack such a font (or one with the typographic features for doing so), or when aiming for file types that cannot use expressive fonts, such as eBook and Web publishing.

This option is naturally only available to formats that use fonts. The setting will render text in uppercase when used with formats like plain-text or the Markdown-based formats.

— *Lowercase*: all letters will be converted to lowercase.

Title Prefix Anything typed into this box will be printed directly before either the binder title, or the suffix if the title has been omitted. If you intend to use this to insert the first part of a multi-line title, insert at least one carriage return after the boilerplate text in this box.

Title Suffix The contents of this box will be printed directly after the binder title or the prefix. Consequently if you intend for this to display information on a line below the main title, insert at least one carriage return prior to typing anything in.

Place prefix/suffix inside hashes Available only to the Markdown-based file types. The title prefix and suffix will ordinarily be placed within the hash marks that Scrivener generates to indicate title depth—thus as part of the heading itself. If you prefer, you can disable the **Place prefix/suffix inside hashes** options to allow text entry outside of the header line itself (Figure 24.7). Prefix and suffix placement remains literal and directly adjacent to the title element. Thus for proper formatting you will most likely need to insert carriage returns to avoid the prefix/suffix from ending up on the same line as the heading and breaking syntax.

```
### <PREFIX><TITLE><SUFFIX> ###

<PREFIX>### <TITLE> ###<SUFFIX>
```

Figure 24.7: The placement of the prefix and suffix around automatically generated hashes with them **inside** and **outside**, respectively.

Number of hashes Available only to the Markdown-based file types. Adjusts the number of hashes to use for titles in this section layout. This can be a way of coercing a layout to always print a heading of a certain depth no matter its literal depth in the binder outline. A "Part" section might always want to use a setting of "1", for example. By default, the "By Level" setting inserts a number of hashes indicating the depth of the item in the outline, regardless of its layout type.

Setting this to "0" will remove the hashes entirely, allowing you to more easily custom format the headings using some other protocol than hashes.

New Pages

This pane contains settings for adjusting how this Layout will act when a new page (or section break, for those file types that are not built around paper) is generated for it by the Separators compile format pane (section 24.4).

Pad top of page with *n* blank lines Adds the defined quantity of empty lines above the title and section prefix. This will have the effect of pushing the

title down into the page and leave an area at the top blank. Since the setting uses lines this can be used with any file type. This option will be previewed in the section layout's tile, in the compile overview screen.

Even though this option is available to the Markdown-based formats, it should be noted that these systems generally disregard blank lines and they will appear to have no effect in the files they ultimately produce.

Number of opening words to make uppercase If your formatting requirements are such that the first few words of the paragraph following a title need to be uppercase, then specify how many words should be set to uppercase with this setting.

Use small caps You can also opt to use faked small-caps instead of all upper case, using the checkbox below this setting. This only works with file types that can use font sizes.

Uppercase even when section is not after a page break This is the one setting that will trigger even if the section does not generate a page break. Use this if you need to capitalise words following a regular heading or soft break, like a divider or empty line.

Add "first-letter" span style to the first letter Available to ePub 3 and KF8 Mobi, this wraps the very first letter of the section in a span with the "first-letter" class assigned to it. This can be used to style drop-caps or other visual effects. You can either create the CSS yourself in the CSS compile format pane (section 24.7), or you can create a style called "First Letter" and use the WYSIWYG formatting in the Styles pane (section 24.5). For true drop-caps you will need to use CSS directly as Scrivener itself cannot create floating boxes.

Setting which page a section falls on

Using the next two options (only available to print and PDF), you can set up common typesetting techniques, such as setting a "part" page to be displayed all by itself on the recto side, with the chapter page following it on the recto side as well and a blank page in between them to do so.

Always start section on The new section can be forced to always start on the verso (left) or recto (right) side of the book. This will in some cases cause an empty page to be inserted, in order to keep the chapter on the chosen side.

Start next section on If the *following* chunk of text also generates a page break, this setting will control how it behaves if it otherwise doesn't use the prior setting itself.

Prefix

Use this area to add content to the beginning of the section, directly before the title area (you will need to insert carriage returns into this field if the prefix should be on a separate line). The prefix can be independently formatted using the standard controls provided. With plain text formats, this can be useful for inserting markup around entire sections. The prefix will be previewed in the layout's tile, in the compile overview screen.

You can optionally use the Place prefix after title checkbox to have the prefix inserted directly after the title, starting on its own line. Again however, it will run directly into the following text unless you leave your own carriage returns at the end of the prefix.

Many placeholders can be used in the Prefix and Suffix tabs. Refer to Using Placeholders in the Prefix and Suffix (section 24.2.3) for further tips on doing so.

> **Style prefixes and suffixes in prefixes and suffixes**
>
> Style prefix and suffixes will not be applied to text when used in the section prefix or suffix. The style itself will be applied to text in rich text formats that support stylesheets, and as CSS classes in the eBook and HTML formats, but styles as used for plain text and Markdown-based formats will not have much use in these fields.

Suffix

The suffix tab works similarly to the prefix tab, only inserting the text you provide here at the very end of the section. You will need to add your own separation at the beginning of the suffix field if it should be on its own line. The suffix will be previewed in the layout's tile, in the compile overview screen.

Use the **Place suffix after subdocuments** setting to have the suffix placed after all descendent items in the binder have been printed. In other words, this will place the text at the very end of a container's section of text. A practical example of how this can be used is demonstrated in this user manual, where the links that allow you to return to the nearest chapter break are inserted after each major section. For those creating XML or similar, this is a great way to wrap entire larger sections of text in container elements.

Using Placeholders in the Prefix and Suffix

Every placeholder (the full list is found in the Help ▸ List of All Placeholders... menu reference) that you can use in the main text area can also be used in the various prefix and suffix fields found within this pane. Here are a few examples of how that capability can be used:

— While most of the examples in this list are more advanced, it bears repeating that the use of simple auto-number placeholders in the prefix fields are

an ideal way to automatically number sections. Throw "Chapter <$n>" into your title prefix field, and never worry about keeping the numbers straight in the binder again.

— Those placeholders that pull information from the current document, such as the <$label> placeholder, will do so for each document that makes use of this Layout, individually.

— You can also *reset* auto-number placeholders. For example you might not want a linear count of figures to be used, but rather for each chapter to have its own figure count starting at one. Since each item in your draft assigned to your chapter Layout will insert a prefix or suffix, that's a great place to put resets, like: <$rst_figure>, which would reset any placeholders using <$n:figure> in the text.

— With the <$img...> placeholder you can insert graphics as part of your heading. In a previous example we inserted a special symbol character to print a stylised asterisk, but inserting graphics opens up a great amount of flexibility in how you format your headings. While working within the format designer, images will not be previewed. However image references will show up in the layout tile area of the compile overview screen.

For full documentation on the image placeholder, refer to Image Placeholder Tags (subsection 15.7.5).

— Compound placeholders can be used. For example you could create a custom "List" type meta-data that provides a few different graphics as named items. One item in that list might be called "Sprouting Seed". You could then for the chapter folders that should use the "Sprouting Seed" graphic set that custom meta-data option, and then use a placeholder like the following in your prefix or suffix field:

<$img:<$custom:ChapterGraphic>>

It's worth noting that images inserted in this fashion will not be displayed as images in the "Formatting" preview area.

— Making use of the <$include> placeholder, you can include the main text content of a particular binder item into any of the prefix or suffix fields.

When used in this context, you will need to refer to the item by name, like so: <$include:name of item>. However this raises a problem in that compile Formats are deliberately separate from projects, meaning if you refer to a document by name it is likely the layout will not work in other projects as expected. A way around this is to have "name of item" inserted via *another* placeholder, using the compound form described above.

For example, <$include:<$custom:Epigraph>> would insert the text in the "Epigraph" custom metadata field for the item using this placeholder. Presumably, that field would have the binder name of the particular quotation you wished to insert into this location. This keeps the compile format suit-

ably separate from project data.

— Even the title itself can be inserted via the `<$title>` placeholder. This might not seem useful since you can already insert the title with a checkbox, but it might come in handy if for some reason you need to print the title *twice*. The following prefix and suffix would produce a LaTeX code to print the regular title as a chapter heading and then assign a bookmark label for cross-referencing purposes (using a slightly different placeholder that omits spaces):

Prefix: `\chapter{`

Suffix: `} \label{<$title_no_spaces>}`

Settings

This final tab will not appear for any plain text or Markdown-based formats.

Paragraph first line indents Utilises the common typesetting practice of discarding the first-line indent for any paragraph following a header and/or section break. The calculation for this can be tuned with a set of options below the main checkbox.[1]

— *Do not change*: paragraph formatting will be left alone. This is the default setting for newly created formats.

— *On new pages only*: the first line indent will be remove from the first (unstyled) paragraph found in the binder item using this section layout. Page breaks can be set up in the Separators compile format pane (section 24.4).

— *At the start of each new document*: this is the default behaviour. Whenever a new document is encountered the first-line indent of the initial paragraph will be stripped out.

— *After empty lines and centered text*: unlike the above two options, this can trigger the removal of the first-line indent from even within an individual section. A full blank line (often used to denote scenes or section breaks), or the presence of any centre-aligned text (such as a scene separator like '#') will trigger it. This can be useful as a compromise between the above two options, where your outline is a casual representation of the reader-accessible structure, and might not strictly conform to section breaks or employ a mix thereof.

[1] This adjustment is handled globally, rather than per specific section type in the ePub 3 and KF8 Mobi formats, via the Text Layout compile format pane (subsection 24.6.4). If you require a more granular approach, consider providing the section layout a CSS class name in this pane and then adjusting the CSS for this section specifically.

Include in RTF bookmarks This option is used by all of the word processing formats, which are derived from RTF initially. It is enabled by default for all section layouts, and thus those items not using the "as-is" assignment in your binder will get an RTF bookmark.

These will create handy navigational references throughout the document in word processors, and is also used to cross-reference document links in Scrivener. It may not always be desirable to have bookmarks at every level of your outline, especially if you use Scrivener's outline feature to break down your book into small blocks. Simply uncheck this to remove the document type and level from the bookmarking feature. If you do this, you will be unable to link to this type of document directly.

CSS class name This and the following setting are available to ePub 3 and KF8 Mobi. When a class name (it is up to you to provide a valid class identifier here) is assigned to a section layout, a `<div>` element will enclose the entire section (including its prefix, title and suffix, if the suffix is not set to fall after subdocuments). This can be used to provide more specific CSS instructions, in the CSS compile format pane (section 24.7).

Hide section in e-book Available to ePub 3 alone. This setting will remove the documents that use this Layout from all forms of navigation in the ebook, including next/prev chapter functions, the internal table of contents, and the automatically generated HTML contents if applicable. It will be referred to as a "non-linear section" in eBook editing programs.

Support for this specification may vary between readers. Some may allow the reader to navigate through these "hidden" sections, but by and large it will not be presented as part of what the reader can navigate to. This can be a useful setting for those writing "choose your own adventure" style stories, or to present extended annotation on a text through the use of links.

Return to chapter ⤴

24.3 Script Settings

This pane is only available to the Final Draft (FDX) format. It provides options for configuring a few details that impact how a few elements of a script should be formatted.

Use default Final Draft screenplay elements By default, when using the "screenplay" script format, Scrivener's output will match that of the industry standards used by Final Draft. However if you've made changes for your own aesthetic tastes, or are unsure of the formatting in general, you can check this option off to remove Scrivener's formatting instructions and have Final Draft handle all of the formatting. This could result in a

compile that looks different from what you've been writing, but in most cases there will be no visible change. Naturally, if you *require* a script format that doesn't conform to the standard screenplay, make this this option is unchecked, or you will lose all of your custom formatting in the compile (Scrivener's copy will of course remain untouched).

Break dialogue and action at sentences Use this option to adhere to the standard of keeping action and dialogue sentences together, rather than breaking them up between pages. If a sentence would have ordinarily been split, it will instead be moved entirely to the following page.

Include revision colors from Preferences Your preferred revision colours, which can be set in the Editing: Revisions preference tab (subsection B.3.3), will be supplied to Final Draft's revision palette, maintaining a consistent revision system between the two applications.

Summaries and Script Notes Fonts The final two options allow you to set fonts for the indicated features within Final Draft. Do note that if you intend to share the FDX file with Windows colleagues, you may wish to change the default Summaries font to Helvetica, Arial, or something else that is commonly available.

Return to chapter ⤴

24.4 Separators

In real-world terms, separators represent an easy and formulaic way of inserting separation between important elements in your draft. A few common examples would be a page break between chapters in a PDF, a section break in an eBook, or even a simple "#" or "* * *" between scenes in a novel. Separators in Scrivener can be inserted either broadly as a default (all folders should have a page break inserted, for example) or very specifically as part of the *role* or function of a section layout (such as an asterism between text for a "Scene" layout).

The separators panel is broken up into three main parts (Figure 24.8):

a) *Default Separators*: these two settings broadly adjust how separators will be inserted in the draft. They will be used for section types that are not mapped to any layouts, and for all layouts by default. Defaults are how we want to treat items in general, perhaps even regardless of what type of item it is in the binder (chapter, scene, preface, etc.).

b) *Section Layouts*: each layout, as defined in the Section Layouts compile format pane (section 24.2), can be set to override the defaults and do their own thing. This capability is useful when the layout is meant to serve a specific role that includes separation—like a Part break in a larger book.

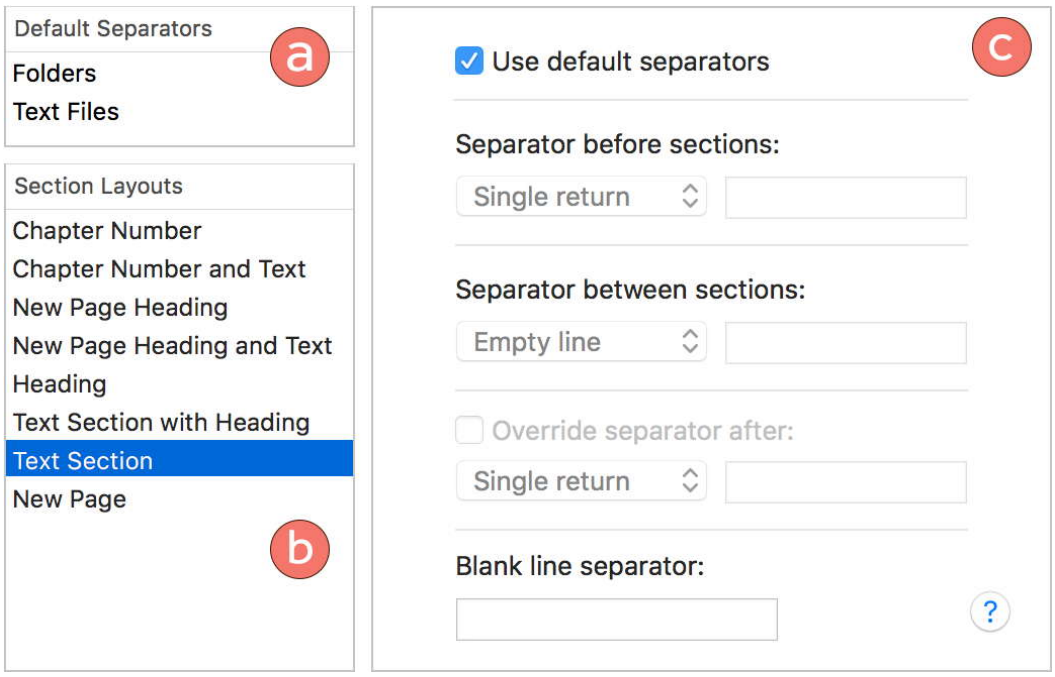


Figure 24.8: Separators can be defined by section layouts or as broad defaults.

c) *Separation Settings*: when clicking on any of the entries in the above two lists, their applicable settings will be loaded into this area on the right.

A few good examples of where and why a Layout might override default separators can be found in the default blank "New Format" settings, as depicted in the figure. We have the "Text Section" layout selected, which acts in accordance with the global defaults—but you may notice that if you disabled the **Use default separators** checkbox at the top, it would insert an empty line between other "Text Section" items—in effect acting like a scene in a typical novel. The "New Page" layout directly below that one in the list *is* set up to override defaults by default (whew), in that it's entire purpose is for inserting a page break separator.

24.4.1 Managing Layouts from the Separators Pane

You may at times find you need to create a new layout on the fly to accommodate a special form of separation. In the lower left hand corner of the Section Layout list are a pair of + and − buttons. Section Layouts can be managed from this pane:

— When adding new layouts it works in the same fashion as the Section Layouts pane would—select the layout you want to duplicate and then click the +. You may of course need to further tweak it in the Section Layouts panel after adjusting its separators.

— Use of the − will fully delete the layout from the compile format. Use with care.

— Layouts can be renamed right in this list as well, by double-clicking on their names. Click elsewhere, or press Esc to confirm your changes once you've edited the title.

The Separators pane is available to all formats except FCF and FDX.

24.4.2 Separator Types

The simplest way to use and think of separators is of having them inserted above the section layout that uses them, thus placing it in between the preceding chunk of text and the current one. This is the usage we referred to before, where a common (and default) behaviour is to insert a page break before all folders. If you add a folder and put some files into it in your draft, it will automatically act like a major section break when compiled.

There are three other options for where or how separation should be handled. It would be easiest to describe how they work together with a few examples. If you wish to play along with the what we will be looking at in the screenshots, you will find a demonstration project called "8-compile_separators_demonstration.scriv" in the Extras Pack (Appendix F), and consult the help file at the top of the binder if you require any further explanation of how the project is set up. For the sake of visual clarity, we'll use custom separators, which allow us to insert arbitrary text between sections.

In Figure 24.9 we see two different forms of separation being employed, as well as a third option that transforms blank lines in the text editor to match the form of separation we prefer between sections otherwise. The options that are in use:

Separator before sections As referred to before, this is the simplest form of separation to think of. The orange chunk of text has been set to "Page break" as its separation type. We don't see the effects of that here however because it is the very first item to be compiled. This is an important exception to keep in mind for this separator option: it must be *separating* the item from something in order to insert a separator.

For the sections coloured in lavender, we have added a visible custom separator. Since there is a space between the first lavender chunk of text and the orange chunk of text, we see that separator inserted there.

Separator between sections When the section type of the item preceding the current item is the same then this separator option will be used. We see the asterism symbol inserted between the two lavender chunks of text because they are of a like kind. If one of these chunks of text had been a different *type* of document (like the orange chunk), then the "– Before Section –" separator would have been used instead.

Blank line separator Available only to formal section types, rather than as a global default to files or folders, this setting will *transform* blank lines found

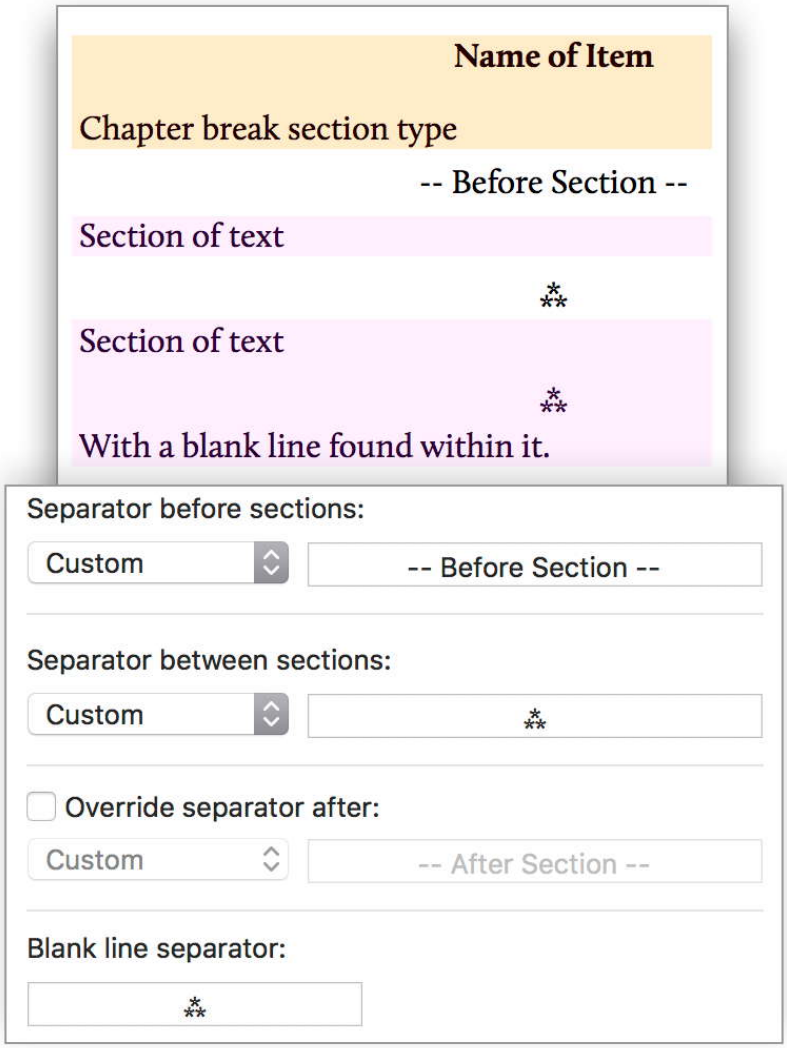


Figure 24.9: Separators can change depending on the items around the chunk of text that is inserting the separator. Compiled output colour-coded for clarity.

within individual chunks of text in the editor. The second lavender chunk of text has such a blank line within it, and we can see it transformed to match our preference of using an asterism between significant portions of text. You can thus *mix* how you write and use the draft outline with Scrivener. If it feels more appropriate to have a sequence of short scenes in one single outline item them you can feel free to do so.

Our next example (Figure 24.10) adds a new option into the mix, and in doing so modifies how these items in the draft work together. If you're following along with the demonstration project, open the compile overview screen and click the Assign Section Layouts... button, switching the "Section" type to the "Section Text (After)" layout.

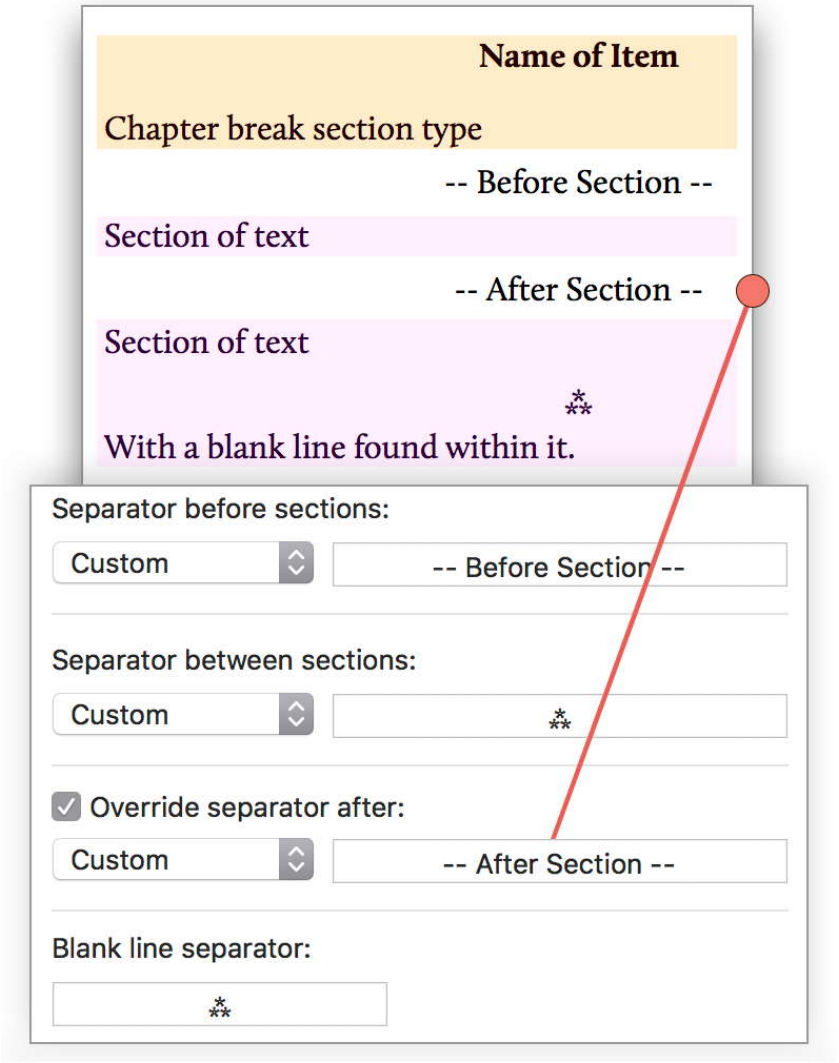


Figure 24.10: Overriding the separator following an item will modify any other separators that might have appeared in that slot, rather than adding an additional separator.

Override separator after This option not only inserts a separator after the section, it will do so in *all* cases, overriding either of the above options if necessary to do so. In our example here, the **Separator between sections** is overridden, but if the third chunk of text was orange, it would in that case have suppressed the page break that otherwise would have been inserted by the "Heading" layout.

Also of note, this setting does not override the **Blank line separator** option, which doesn't formally insert separators *between* chunks of text, but rather modifies how ad hoc separators you type into the editor are formatted.

Need a less adaptive behaviour?

As you may have seen by now, separators are by their nature contextual. They are meant to be inserted logically into your draft, and will not blindly duplicate separation between items or insert separation where there is nothing to separate. If you do need separation around an element *no matter what*, then the Section Layouts prefix and suffix settings (section 24.2.3) settings will do just that. The provided demonstration project also includes a section layout called "Text Section (Prefix/Suffix)" that shows the interaction between a section prefix & suffix with the separators around them.

24.4.3 Separator Settings

Each of type of separator that can be made use of in defaults or per layout will have the same four options available to them:

— *Single return*: a single paragraph break will be inserted, causing the final appearance to run from one document to the next with no visible "seam". In essence this is the "no separator" option.

This can have an adverse effect in Markdown-based and Fountain formats, where a clear empty line is expected between all elements, including paragraphs.

— *Empty line*: two paragraph breaks will be inserted, causing a visible space between the items.

— *Page break* or *Section break*: a page break will be inserted, causing the following item's text to move to the next page of the manuscript. Those formats that do not have a concept of paper, such as eBooks and web files, will refer to them as "section breaks", and use the following behaviours:

- Plain-text (TXT): the Unicode "Form Feed" control character (U+000C) will be inserted at the beginning of the line of the item that generated the separator with no carriage returns around it. Some text editors may handle this code in their display of the text (TextEdit in Page Wrap mode for instance). If the intention is to use the output of this document in processing this separator should in general not be used unless the processing engine is capable of handing a Form Feed character in some intelligent manner (many will just throw errors).
- Web page (HTML): an `<hr/>` element will be inserted instead of a page break.
- Fountain: the markup code for a page break (===) will be inserted.
- eBook formats: all eBook formats will use a page break separator to indicate a formal cut to a new section. This is a significant cut in that

it will be the basis for generating the internal table of contents (as well as the HTML contents you can see in the reader). These cuts are also inserted into the navigation index for forward/backward by chapter movement in those readers that offer the capability.

- Markdown-based formats: the markup code for a section break (----) will be inserted. A special exception exists for MMD → LaTeX (.tex). The syntax for a page break will be inserted (`\pagebreak`), in such a way that MultiMarkdown will pass the code directly through to the final .tex file.

— *Custom*: Anything entered into the adjacent text field will be placed between the two items on its own line, using the paragraph attributes of the line preceding it with centre-alignment added.

- Image Placeholder Tags (subsection 15.7.5) can be used here, providing a way of inserting custom separator graphics between sections.
- If you require more spacing, you can insert your own carriage returns with the Opt-Return key combination.

There are a few additional options that may appear at the very bottom of this pane, depending on the document type selected:

Ignore indents when centering custom separators Available to file types that use formatting. Since separator lines inherit their base formatting from the preceding paragraph, this can often mean the separator will have a first-line indent like an ordinary block of text would. This indent pushes the calculation of what is "centre" over by the width of that indent. Normally you will want a separator to be aligned to the middle of the *page*.

You may want indents to be factored in if the section of text being separated is itself bulk-indented. For example, an extended block with separators in between sections of that quote would arguably look more coherent if the visible separator between them were indented accordingly.

Custom separator font Available to file types that use formatting and are capable of drawing upon system fonts (unlike eBooks). Tick the checkbox to enable the font selection tool. This can be particularly useful if you have a wing-ding style font that you want to make use of. This setting will not *not* be previewed in the settings above, so you will need to know which characters to type in to make use of such symbols.

Return to chapter ⤴

24.5 Styles

Where it comes to compiling stylesheets from your project, Scrivener takes a somewhat unique approach to the problem in that every compile format can

potentially change every aspect of the styles in your project as you see fit. Another unique aspect of its stylesheet system is that this panel will be available to every file type in the list save for scriptwriting formats. That means even plain text and Markdown-based files can use stylesheets (though naturally they will ignore those settings dedicated toward formatting).

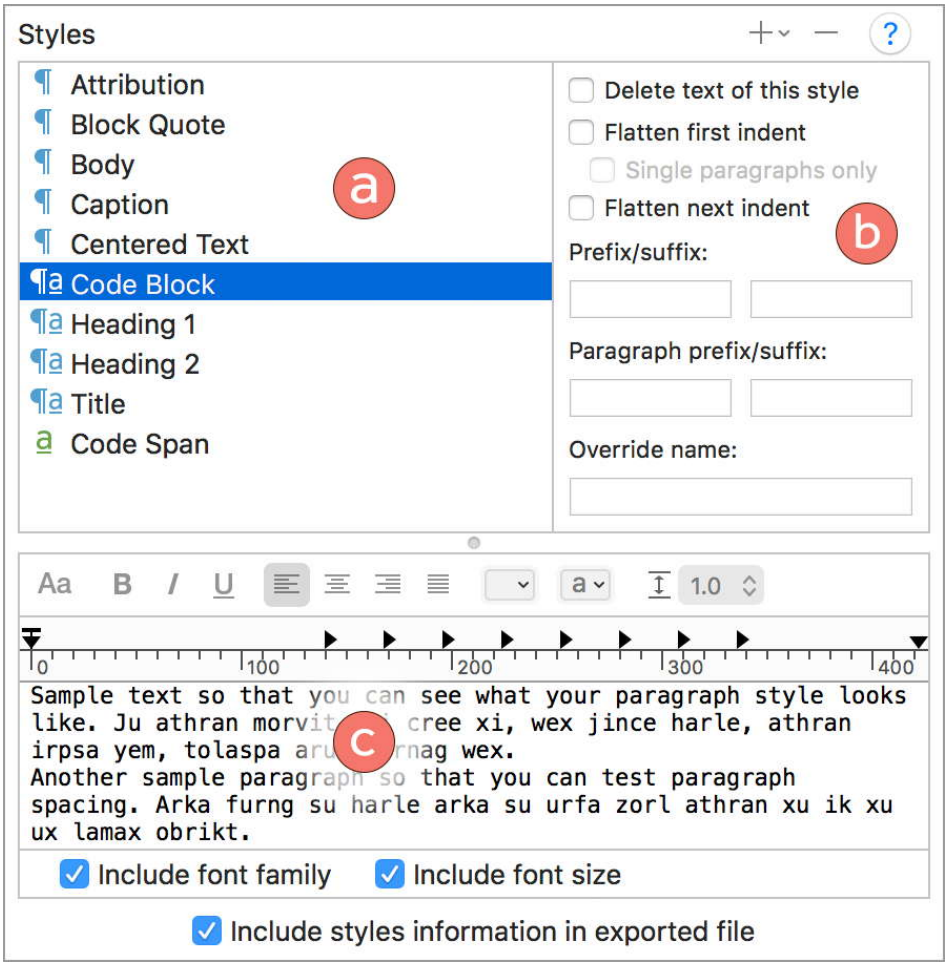


Figure 24.11: The Styles compile format pane (showing RTF options) is for creating special styles that can transform how a document looks.

The Styles panel is composed of a few simple areas:

a) The style list on the left, which may be empty when creating a new format, contains a list of styles that will be available for use within the format—and as well a list of styles the format will be looking for in the project's source material.

b) Various style options will be provided to the right of this list, depending upon the document type being worked on at the moment. These tend to be more advanced options that alter how the styled text will function or be displayed.

c) Below the list is what should be a familiar interface for changing the formatting of the style. In this case we have the "Code Block" style selected, and can see that it will print code using a monospace font when compiled. If text using "Code Block" exists in the project, it will end up looking like this example here, rather than how it looked in the editor.

There are a few key aspects of this system to keep in mind, with regards to how styles work as a part of a compile format:

— **Names matter**. The only tool Scrivener has at its disposal for matching styles you use in your project with how they should be defined and formatted in the compiled result is by *name*. Matching styles will have their attributes modified as set up in this pane. Thus a normal looking block quote in the text editor can end up double-spaced and in Courier for submission.

  If you require a different style name, as stored in the final document (say to match with an import template in another program), you can use the **Override name** option, for those formats that support named styles in the output.

— It is okay to have style names that don't match styles in one specific project. For example our built-in "Proof Copy" format has styles for Captions, which you might never use in your projects. But if you do, the format is prepared to handle them.

— On the other side of the coin, all styles in the project will protect the formatting they apply to text when you compile, if nothing matches their name in the list—even if the text around them is being modified by the section layout settings. This is selective behaviour: a style that does not address font families could have its font overridden by the compile format, but not say it's paragraph alignment.

— Something to be aware of is that many of our built-in compile formats and templates have been designed specifically to modify the stock stylesheet that comes with every new project. If you use "Block Quote", then expect it to be transformed into a suitable format for Courier 12pt submission, or whatever the case may be.

— The format styles that you create in this pane will be available to other compile format panes that make use of styles, within this one format. For example in the Section Layouts pane you can assign styles to headings and body text—but they will be format styles, not project styles.

  Considering that, if you do not intend for such "internal styles" to accidentally format project text, it would be good to choose names for them that are likely to be unique.

24.5.1 Creating a New Compile Style

To create a new style in the list:

1. Click the + button in the upper right-hand corner of the pane.
2. You will be presented with a list of options divided into the following categories:

   — Generic new styles for paragraph, paragraph+character and character. For more information on the distinction between these types, refer to Paragraph and Character Styles (section 15.6.2).

   If applicable, the starting attributes and settings for the new style will be determined by the selected style in the list—so this is a good way to duplicate a style as well. For example, if you select a paragraph + character style and create a new paragraph style, it will inherit the paragraph formatting from the original.

   — Next, all paragraph and paragraph+character styles found in the current project will be listed for your convenience as a starting point. This does not change anything said before: once created in the format it will be *entirely separate* from the project style.

   Using one of these will copy the style's formatting and settings into the compiler. The changes you then make will adjust how that particular style (by name) compiles.

   — Lastly all character styles found in the project will be listed. They function identically as the above.

3. If creating a generic style, type in a name for the style in the list, and click elsewhere or press Esc once done.

24.5.2 Renaming and Removing Compile Styles

Styles can be freely modified in the list once they have been created. To rename a style:

1. Double-click on the name of the style in the style list.
2. Once the edit is complete, press Return or click elsewhere to confirm your edit.

Removing a style from the list will in most cases simply remove the instructions provided to the compiler for handling a style by that name. In some case, if the style has been made use of in other compile format panes, the result will be to reset those areas to default, or strip the style assignment from any text it had been associated with:

1. Select the style you wish to remove.
2. Click the – button in the upper right-hand corner of the pane.

This cannot be undone, but you can always Cancel editing the compile format if you make a mistake (just keep in mind that will cancel every change you've made since opening the pane).

24.5.3 Compile Style Options

The sidebar to the right of the main style list contains a number of options that can modify the behaviour of the text assigned to that style, sometimes even radically—you could for example have a style that deletes the styled text entirely and replaces it with a counter using one of the prefix or suffix options. These settings will impact not only styled text found in the project using this format, but text that has been dynamically assigned to a style by the compile format itself.

To modify how a style works, click on it in the list to load its settings into the sidebar.

Delete text of this style All text that has been associated with this style will be removed from the output. A practical example of this feature is in use with this user manual, where specific phrases of text relating to macOS or Windows alone can be selectively omitted depending upon which format it has been compiled with.

Treat as raw markup This option is available to all file types that make use of or are capable of generating markup: TXT, HTML, the eBook formats and the Markdown-based file types. In all cases the effect of this option is to fully suppress any compile behaviours that might modify the text as you typed it in, allowing you to inject raw markup as intended to the final output.

To provide a very simple example, if you manually type in `<img src="..." />` into your editor it would end up as visible text when you compiled, rather than the HTML to insert an image into the web page you are writing. This happens because Scrivener converts the punctuation marks to encoded entities to protect what you've written from being interpreted as HTML. Wrapping the HTML in a style with **Treat as raw markup** enabled for it, is the solution for making that text functional HTML. In this way we can both write *about* HTML by typing it in normally, and use HTML directly with raw markup styles.

When used in conjunction with the ePub3 /KF8 compile file types, it is good to know that these formats are ultimately generated by MultiMarkdown internally. The implication being that this checkbox passes the text through to the MMD file itself, *not* directly into the final HTML output. That means you can use MMD syntax if you wish to do something that the text editor itself is incapable of converting from RTF. Raw HTML is fine to use

with these formats as well, since MultiMarkdown by its nature will leave bare HTML untouched.

When using the **Convert rich text to MultiMarkdown** option in the compile overview screen's General Settings tab (subsection 23.4.3), this setting can be used to have Scrivener leave the marked text alone, making it possible to use raw Markdown in a document that otherwise would convert the markup to visible punctuation.

Flatten first indent This and the following two settings are available to print, PDF, the word processing formats, HTML, ePub 2 and legacy Mobi.[2]

The first paragraph within the assigned range of text will have any first-line indenting removed.

> **Looking to adjust indent settings globally?**
>
> These features are not intended to be used for handling large and dynamic amounts of text (such as all body text), but rather smaller ranges such as individual block quotes, monologue formatting and so on. For bulk indent management, you should use the Section Layout: Settings tab (section 24.2.3), and for ePub 3 and KF8, the Text Layout pane (section 24.6).

Single paragraphs only The first-line indent will only be removed from one line uses of this style.

Flatten next indent The paragraph of text *following* the paragraph using this style will have its indent removed. A common use case for an ability like this would be to suppress the indent of the paragraph of text following a figure caption, meaning the "Caption" style itself would be responsible for declaring that rule.

Prefix/Suffix The start of the styled range will have the text of the **Prefix** field added to it—styled in the same fashion as the text that generated it. Likewise the **Suffix** will be added to the very end of the assigned range of text. If the range spans multiple paragraphs, there will only be one prefix and suffix at the very beginning and end of the assignment. This (and the following) setting can be useful for a number of different applications:

— In plain-text technical formats, such as XML, this can essentially boost Scrivener's stylesheet system into the fully semantic realm. A range of text can be surrounded in an element such as `<attribution>` or `<figcaption>`.

[2] Both ePub 3 and KF8 Mobi depend upon CSS for this level of behaviour adjustment.

— In all uses, the practical ability to insert stock generic text around styled ranges is just as useful as it is to embellish folder names with text like "Chapter" followed by a number. For example, consider that captions could be automatically labelled and numbered with a "`Fig. <$n:figure>.`" prefix.

Paragraph prefix/suffix Operating in a similar fashion to the previous set of options, this setting only applies to Paragraph and Paragraph+Character styles, and it will insert a prefix or suffix around each line of text *within* a styled range.

To use HTML as an example, you could wrap an entire block quote in `<blockquote>` and `</blockquote>`, and then individual paragraphs within that quote with the `<p>` and `</p>`.

Override name Available to all of the word processing formats, except RTFD. For cases where the style name used by compile settings (which in turn is how it will be worked with in the project itself) must be presented differently in the output document, use this field to completely overwrite the visible name of the style. For example if you are working with a Word template that expects "Blockquote" instead of Scrivener's "Block Quote", you could override its name to that, here.

CSS class name Available to ePub 3 and KF8 Mobi. This will be of considerable use if you intend to write your own CSS. The ranges of text you assign with a style will be classed (either as `p` or `span` elements accordingly) with the name you provide here. If you do not provide a class name, Scrivener will attempt to automatically generate one based upon the style's name.

It is up to you to select a valid CSS class name. If you type in invalid punctuation or uses spaces, then you will likely break your stylesheet.

24.5.4 Compile Style Formatting

Below the style list and options sidebar you will find a text formatting editor with all of the various tools needed to fully format text. This works in the same fashion as the Formatting control in the Section Layout tab (section 24.2.3), with two exceptions:

1. Naturally, there is only one element to work with. So you need only worry about clicking into the text area once to activate the formatting controls, rather than clicking in a particular area like you might in the Formatting tab.
2. For style types that exclude certain types of formatting, those formatting options will be disabled. For example, if you create a Paragraph format you will not be able to change the colour of the text. You will need a Paragraph+Character style for that.

With TXT and Markdown-based file types, the controls for formatting text here will be in most cases useless, and can be ignored entirely. You will primarily be interested in the options in the sidebar above. They are included in the event that your compile format is meant to generate useful output to a text-based file as well as a formatted type of file.

Below the formatting area, you may find additional options:

Include font family For cases where you merely wish to adjust some aspects of the character formatting but leave the font family alone, enable this checkbox.

Include font size Likewise, if the inherent font size should be left alone—either to be established by the compile format or the underlying text in the project—check this box.

Include styles information in exported file Available to the word processing based formats (excluding RTFD), this option is enabled by default and will cause the stylesheet that is generated by combing both the underlying project's styles, and any styles added or modified by the compile format, to the exported document. This will greatly enhance the flexibility of your file after it has been compiled, and in some workflows will be a requirement for submission or collaboration.

Return to chapter ⤴

24.6 Text Layout

This is where a few general decisions about the overall layout of the document can be made, and as such it presents different options depending on the file type you are working with. This panel is available to all file types except Final Draft and Fountain.

Document suffix This option is available to all file types. It will place the provided text at the very end of the compiled file, signifying the end of the document. Some submission formats require a special punctuation sequence at the end, and this option can be used to keep that sequence outside of the main working area in the editor.

When using plain text to create markup based formats, this area can be used to insert a document footer, closing off any containing elements or including final materials, such as bibliographies. (And for plain text you will also have a matching **Document prefix** field that can be used to open these containing elements or otherwise manager the header area of the document.

Before back matter All save for the plain text and Markdown-based formats will have this secondary option available, that when enabled places the document suffix directly preceding any material added via the **Back matter** setting in the compile overview screen in the Contents tab (subsection 23.4.1).

Use hyphenation Available to all but the plain text, Markdown-based and HTML file types. By default, hyphenation will not be used. When full justification is being used to format the text, this option can substantially improve the readability and appearance of the compiled result. This option only works with select languages supported by macOS.

24.6.1 For PDF and Print

One extra option is available when printing or using the PDF file type:

Empty Lines Across Page Breaks Enable the following checkbox and then supply a custom separator to be used as a stand-in when an "Empty Line" Separator (section 24.4) is scheduled to be used, and that line would otherwise be hidden by the page change. This is a common typesetting convention for making sure that separations between scenes are indicated at all times.

24.6.2 For Word Processing

With the word processor formats (excluding RTFD) this pane allows for some advanced page layout options, such as widow and orphan protection where applicable and column based layout.

Avoid widows and orphans When used with a compatible word processor, this will enable widow and orphan[3] protection for your paragraphs.

Use columns Columns will reformat your exported manuscript into a specified number of vertical columns. To enable the use of columns, tick the "Use columns" box.

Start columns Three choices are provided for where columnar format should start in the document:

- *On first page*: start column layout immediately
- *After first document*: useful for title pages or the abstract block in many style guides used by the sciences

[3] Widows are remnant lines where the paragraph breaks across the page, resulting in only a few words after turning the page. Orphans are the opposite, where the paragraph begins so low down on the page that only the first line can be read before a page flip is required. This option will strive to reduce instances of this by moving paragraphs from one page to another to keep the text as cohesive as possible.

— *After front matter*: if **Front matter** is selected in the compile overview Contents tab, it will be entirely displayed in single column mode.

Number of columns The number of columns on each page can be adjusted here. Any value between two and four can be selected

Space between columns You may also adjust how much padding will be used to space the columns apart from one another, from 1/8th of an inch to 3/4.

24.6.3 For Web Page

Web pages (.html) have two exclusive options available (refer to the eBook options section for documentation on the **Use 100% width for images wider than...** setting):

Use centered column to restrict body text width Instead of allow text to flow from one edge of the browser window to the other, this option will enable the common tactic of constraining the text column to a maximum width in the middle of the viewer.

Body text width You can set the width of that column with this setting, in points.

24.6.4 For eBooks

Use 100% width for images wider than... Determines the effective maximum width for images by leaving it up to the e-reader after the width you set (in points, not pixels). Images narrower than that size will use a fixed width no matter how wide the e-reader's display might be. Note that some readers might always display images a certain way regardless of your settings here, particularly if the image is wider than the display.

Include scriptwriting CSS This setting is available to ePub 2 and legacy Mobi formats; if you need scriptwriting elements in your ePub 3 or KF8 Mobi book, refer to the CSS compile format pane (section 24.7). If your eBook is composed of, or contains any script formatting, you should enable the "eBook contains script formatting" option. Scrivener will insert special formatting rules that match the script format settings you are using in that project, and typeset these elements appropriately.

In most cases, the default styling will be acceptable, but with some custom formats that have difficult to emulate styles, you may need to customise the rules which determine the final appearance in the eBook with Cascading Style Sheet (CSS) syntax[4] in the text area that will appear below this checkbox when enabled.

[4] http://www.w3.org/TR/2011/REC-CSS2-20110607/

Each script element will be provided with a separate CSS class. All classes should be assigned to the p element, if they require specificity.

Embed MathType equations as MathML This is a default setting, and is only available, for ePub 3 and KF8 Mobi type books. Any MathType images found in the compiled text will be inserted into the eBook as a <math> element, conforming to the specifications for MathML syntax. You should test your intended target readers for compliance with this format. If they do not display MathML equations correctly, or satisfactorily, you might wish to turn this feature off and have Scrivener convert equations to raster images (PNG) for maximum compatibility.

Remove first line indents Available to ePub 3 and KF8 Mobi. This is a global setting for setting the behaviour of first-line indents that follow different types of elements in the book. The effects of this setting are implemented in the CSS compile format pane (section 24.7), and can be further tweaked there if necessary:

- *From all paragraphs following other elements*: anything other than another a paragraph will cause the first-line indent to be removed. This uses the HTML definition of the <p> element, which may not always be a paragraph in the literary sense of the term. As this is an all-inclusive setting, the following options will become redundant and be disabled.
- *From paragraphs after headings*: any paragraph following a numbered "H" heading, such as <h1> or <h3> will have their indent removed.
- *From paragraphs after separators*: when Scrivener inserts a separator via the Separators compile format pane (section 24.4), paragraphs following them will have their indent removed. You can also create your own separators with this function in your HTML by using the "separators" class on paragraphs.
- *From paragraphs after blank lines*: operates similar to the above, only on those separators using the "br" class on their paragraphs—allowing a mixed form of separator usage where visual separators do not suppress the indent.

Return to chapter ⤴

24.7 CSS

This pane is available to ePub 3 and KF8 (Mobi) formats only. It provides complete access to your book's Cascading Style Sheet (CSS)[5], which means you can

[5] http://www.w3.org/TR/2011/REC-CSS2-20110607/

apply your own custom-designed look and feel and go well beyond what Scrivener itself is programmed to handle in terms of design. You can also make minor modifications to Scrivener's automatic output as you see fit.

You don't need to be a CSS guru to set up the **Basic text formatting**, in the upper half of this pane, and you might well want to. It will determine how the body text of your book will be formatted as a default. This is what would typically be controlled by Section Layouts individually in more directly formatted file types. Formatting here is done using the familiar Format Bar, Ruler and main application menus. The formatting you choose will have those bits of it extracted that can be turned into useful CSS for you in real-time, into the "Default Stylesheet" box below.

And that goes for nearly every single appearance-related option in the compile format designer, even down to settings like **Reduced marker font size** in the Footnotes & Comments compile format pane (section 24.19). That setting will generate a class and insert a selector for that class into the default stylesheet with instructions to print the marker at `0.65em` units. We will make note of the classes Scrivener uses where applicable, but we'll leave general selectors for broad element types (like `p`) up to you to discover.

Include scriptwriting CSS Tick this checkbox is your book has been composed using Scrivener's scriptwriting feature. This will cause the formatting designed by the script to be generated as CSS and inserted automatically into the "Default Stylesheet" section, below.

Custom and Default Stylesheets Above the CSS text fields you will find a drop-down for setting how CSS should be handled in your book:

— *Use Default CSS Stylesheet*:[6] this causes the book to *only* use the default CSS that Scrivener will generate automatically, via settings made in the format designer and so forth. This is the simplest approach, and if you like how our books look by default that may be all you ever need.

— *Use Custom CSS Stylesheet*: this option gives you full control. The Default Stylesheet will remain available for you as a reference, and a source for copying and pasting example styles into your own stylesheet, but it will not be used and every aspect of the book's formatting will depend upon your custom style.

— *Append Custom CSS Stylesheet*: if you just want to make a few modifications to the default style or add a few definitions for styles you've created yourself, this will be the best option. Both stylesheets will

[6] Yes, we are aware that this is a little bit like saying ATM Machine, or PIN Number, but as with those phrases they tend to roll off the tongue a little easier than just barking out acronyms, so bear with us.

be used, with yours following the defaults (meaning you can override the defaults by using identical identifiers).

Return to chapter ⤴

24.8 Document Title Links

This is a special option pane in that it will only appear if at least one Section Layout requests the use of a title prefix or suffix in its settings (section 24.2.3). The options in this pane refine how these titles should appear when referring *to* the items via an internal document link—such as in a table of contents listing or in general cross-references.

These settings will be useful in cases where what you are linking to is important to identify by its full name. For example, if you set up folders to prefix their title with "Chapter <$n>", you can ensure that all cross-references pointing to those folders use that numbered prefix along with, or perhaps entirely in replacement of, the folder's natural title.

For a full explanation on how to work with this feature from the editor side, refer to Compiling with Document Links (subsection 10.1.4).

Update titles in document links with prefix and suffix settings The rest of this panel depends on this switch being enabled. It defines the base behaviour described above, where a hyperlink to "The Folder" may become "Chapter 21 - The Folder".

Carriage returns will be stripped out of the prefix or suffix as necessary, in order to keep links from turning into multiple paragraphs in the text.

Links use title prefixes only (exclude title and suffixes) With this option enabled, the title and suffix will be discarded from the link text even if it prints more fully at the point of the title itself. Using the above example, our hyperlink would simply refer to "Chapter 21", even though at the chapter bring "The Folder" is printed on a second line below that.

Do not include title suffixes in updated links With this option enabled, the suffix will always be dropped from the link. This will be of use if you use the suffix to print some decorative elements below the titling.

Override title prefix separator in links This will insert the character provided in the **Prefix separator** field below, between the prefix and the main title. When this option is used without the following, all forms of punctuation and whitespace between the prefix and the title will be replaced by the separator. For example, "Chapter 21: The Folder" will become "Chapter 21 - The Folder".

Only override prefixes containing return characters In some cases the prior option by itself may be too aggressive as it will replace portions of the title you wish to leave intact. This secondary option forces the definition of "separator" to only those cases where the prefix terminate with a return character. All other characters will be left alone, meaning a prefix such "Chapter <$n>: " will end up in the hyperlink as "Chapter 21: The Folder".

Return to chapter ⤴

24.9 HTML Elements

This pane is available to ePub 3 and KF8 (Mobi) formats only. It is used to map a compile format's styles to the sorts of styles used in eBooks. Since Scrivener cannot guess what you mean by the name of a style or its formatting alone, it needs a little help in getting them to look and work the way they should in the eBook.

It is worth reiterating that compile formats are not "aware" of any particular project's settings, and as such you cannot use styles from the project's stylesheet directly. A style must be defined in the Styles compile format pane itself (section 24.5), and in doing so, styles from your project will become handled by the format and this pane.

The "E-Book" Format

The built-in "E-Book" format that ships with Scrivener will have these settings already wired up to named styles. If you use our default stylesheet while writing, you won't have to set these up! It's also a good starting point for your own formats.

— **Page title style**: this special style is only used for the automatically generated HTML table of contents and the endnotes page, if applicable. In most cases you will want to set up the style used here to match the look of the style used for your other headings, but if you wish you can use a separate format for these two titles.

— **Block quotes style**: the styled range will be wrapped in the `\<blockquote>` element.

— **Code blocks style**: each paragraph or line of text marked with this style will be `\<pre>\<code>` elements. As with code blocks in many contexts, you may need to ensure the lines wrap at a reasonable width, as the preformatted element does not by default use automatic word wrap.

— **Code span style**: text tagged with the chosen character style will be wrapped in the `\<code>` element, which by default will apply a monospace font. You may only select from character styles, as this is designed for spans of text within paragraphs.

— **Captions style**: the chosen paragraph style will be used to generate image and table captions, when used on the line directly preceding or following these objects in the text editor. Genuine captions have an advantage over formatted text in that they may be used intelligently by the reading software to provide lists of figures and so forth.

If you do not use these settings, styled ranges will be inserted into the eBook as classed (and unstyled by default) paragraphs.[7] For example if you use the "Block Quote" style, the paragraphs using that style would be classed as "block-quote". So you could adjust the CSS (section 24.7) in your compile format to style these paragraphs to look like block quotes... or you can just map that style to block quotes in this panel and in most cases not even have to style it explicitly as many eBook readers will handle semantic block quotes properly without help.

Return to chapter ⤴

24.10 Markup

This pane is available only to the plain-text (.txt) file type. If you are looking to use Scrivener to generate markup files, but aren't a fan of using one of the Markdown-based approaches for doing so, then you will very likely find this pane to be of considerable use to you.

This pane will nearly always be one of several you will want to use together in concert to create your own file types from scratch:

— Section Layouts (section 24.2) serve as a nexus for how many of the other features of the format will work together. They can assign styles to text, which in turn can be used to mark up text. They are also used for generating separators between chunks of text, which can be valuable for inserting necessary code like page break requests or opening and closing environments or elements as the case may be. They can also be used to wrap sections of text in prefixes and suffixes.

— Separators (section 24.4), as mentioned, provide the ability to insert snippets of text or spacing in between binder items. Where the prefix and suffix fields in Section Layouts are good for wrapping chunks of text in environment calls (like `\begin{quotation}` and `\end{quotation}` in LaTeX), if you need intermediary breaks—or wish to *suppress* breaks between like items (perhaps two back to back items marked as "Quotations"), then separators will be another way of accomplishing that goal.

[7] If you want to specify the class name a style should use, you can do so in the Styles compile format pane (section 24.5), with the **CSS Class Name** style option.

— Styles (section 24.5) in Scrivener are most decidedly not just for those using word processors! Our style configuration allows for block and line level insertion of prefix and suffix code, as well as the suppression of the text within the range.

— Text Layout (section 24.6) when paired with plain-text can be used to insert boilerplate content at the top and bottom of the file. For any format that needs a bit of preamble before getting to the content, this will be an invaluable tool for getting a complete source document built.

— Transformations (section 24.13) include useful tools for keeping the source text clean, particularly one setting that makes text XML-friendly by escaping special characters and so forth.

— Processing (section 24.22) provides a way to further automate compilation through the use of shell scripts and external utilities. Using this capability, you can generate what amounts to interim formats with the compiler, which are further processed by other tools to generate the final output. The concept is much the same as using KindleGen to create a .mobi, or Pandoc to create a PDF. With the Processing pane you have full control over the tool chain, including building your own from scratch with custom scripts.

Most of the settings in this panel allow for a prefix and suffix to be inserted around a range of text, transforming it from rich text to marked up plain-text.

External link prefix and suffix The text of any hyperlink pointing to an external resource will be surrounded by the markup you add to these fields. You can supply the URL itself within this markup using the `<$url>` placeholder. An example prefix could be `<link url="<$url>">` with the suffix being `</link>`.

Internal link prefix and suffix For internal document links, those pointing to other items within the compiled draft, you can supply different markup, and make use of the `1044` placeholder for referring to the internal section by name. This placeholder is also valid for use in other contexts—for example in the title prefix of a section layout—making it possible to effectively identify or label nodes of text and cross-reference to them from elsewhere.

Enclosing markers for unstyled italics Raw italic text—not text made italic by a semantic style, can be treated specially here with a prefix and suffix. This will work better with simple ranges of text, naturally. Long blocks of italic text spanning paragraphs or other elements may produce invalid syntax depending on your target file type.

Replace images with text Images can be handled in two different fashions: you can either embed the graphic directly into the output file as hexadecimal (`$hex`) or Base-64 (`$base64`).

The other method is to have the image exported as a file with the **Export images** checkbox, and then refer to the exported image by name with the `$filename` placeholder. When using this method, the compiler will create a folder to contain both the generated .txt file and all of the images, so you will be linking to images relatively from the same directory.

The width and height of images are accessible to you as `$width` and `$height` respectively (in points).

Return to chapter ⤴

24.11 Metadata

This pane is used by the Markdown-based formats. Both MultiMarkdown and Pandoc are capable of defining format-specific information with their metadata systems. The functioning of this panel is identical to the Metadata tab in the compile overview screen (section 23.4.2).

The principle exception is that this metadata table will have an uneditable marker in the upper list, labelled "Insert Project Metadata Here". This can be freely dragged around among other formatting keys, and as you can guess, any metadata gathered from the project's compile settings or metadata documents in the binder will be gathered and inserted at that point. This can be quite useful if your format is using MultiMarkdown's LaTeX system, where such fields as the title and author need to be declared *after* the metadata key that inserts the preamble, but *before* the key that generates the title page and formally begins the document. By example:

```
LaTeX Input: mmd-memoir-header
[Insert Project Metadata Here]
LaTeX Input: mmd-memoir-begin-doc
LaTeX Footer: mmd-memoir-footer
```

Where is the document's metadata be described?

The purpose of this panel is thus to accommodate the establishment of the format itself, rather than to define any particular aspect about the project that is being compiled with format. You should in most cases use the Metadata tab in the compile overview screen to describe the compiled document, or use project templates to establish common details of metadata that rarely change, such as copyright information.

Return to chapter ⤴

24.12 LaTeX Options

This pane, available to the MMD (LaTeX) document type, is used for selecting which of the three built-in document classes Scrivener has boilerplates for, along with a few custom options. The settings for the **LaTeX document class** drop-down are:

— *None (Use Metadata)*: you will be in full control over the process when using this option. Use this option if you have your own boilerplate files, or would rather use none at all.

— *Article*: the Memoir class, with tweaks to present text similarly to the vanilla article document class.

— *Memoir (Book)*: an extensive class that is capable of formatting a wide variety of books (not only memoirs). In fact, the user manual you are reading uses a customised version of this class.

— *Manuscript*: the Memoir class, with tweaks to present text in a standard 12pt Courier style submission manuscript. This option requires a functional XeLaTeX system, as it uses system fonts.

— *Tufte (Book)*: a document class for book-length works, with typesetting inspired by designs by Edward Tufte.

— *Custom*: use this setting to supply your own preamble and footer, saving them directly into the compile format itself. This is a great choice if you wish to package your format up for others to make use of, rather than having it depend upon boilerplate files.

When you compile using one of the three built-in classes, and you do not have a stand-alone copy of MultiMarkdown installed (subsection 21.6.3), the .tex files necessary to typeset the document will be exported into the compile folder along with your document. You could modify those files to taste, but a better approach would be to either use that information to build your own "Custom" class, or place those files in your texmf path and modify them there (Scrivener will defer to locally installed copies and produce duplicates into the compile folder).

With the "Custom" option selected, the following tabs are available:

Header The initial preamble should be placed here. Typically anything that needs to be declared prior to establishing document variables, such as the title and author, should be put in this field.

Begin Document This gives you a second preamble field for working with anything that would require additional document metadata such as the title or author. This field will traditionally end in `\begin{document}`.

Footer Anything that would need to be declared at the end of the main and back matter. Commands to generate glossaries, indices and other footer material can be placed here.

Return to chapter ⤴

24.13 Transformations

This format pane provides useful textual transformations that can alter text itself, or its formatting, according to simple rules you provide. A simple common example is the choice to convert italicised text in your editor to underscored text, used in some submission guidelines. An example where the text itself can be altered is to make it safe for embedding into XML by converting ensuring all characters are ASCII compatible, as some technical formats may require. As with many of the other panes, only those options that are relevant to the file type being worked with will be presented.

Convert "Smart" punctuation to "dumb" punctuation The three forms of punctuation that Scrivener can automatically generate as you type (and will by default), can be converted to ASCII-safe equivalents. This will be necessary for some programs like Final Draft, or for technical formats like LaTeX and XML. If you require different or more precise transformations, consider using the Replacements compile format pane (section 24.15).

- Typographic, curly or smart quotes will be straightened. The punctuation marks that will be used to represent quotes will be determined by your system localisation settings, in the System Preferences: Keyboard: Text tab.
- The ellipses character will be converted to three full stops.
- Em-dashes will be converted to double-hyphens (--) for most formats. For the Markdown-based formats, a triple-hyphen will be used, as these systems use double-hyphens for the short en-dash.

Convert to plain text This option is provided to plain-text (.txt) and the Markdown-based file types. It is used to convert visual spacing found in the source text to literal whitespace characters, using approximation to add a number of spaces or carriage returns to emulate that visual spacing.

This feature will make use of the formatting that results from any compile settings. If the layouts you use to print text do not modify the formatting, then the original formatting in the text editor will be used for conversion—meaning that if you want your paragraphs to be double-spaced for Markdown, you should either have your paragraphs formatted with visual spacing in the editor, you should use a Layout that applies that formatting in the compile settings.

- *Paragraph spacing*: add spacing in conformance to any "before" or "after" paragraph spacing in styled text. This function determines spacing by rounding up the supplied value. If the base font is 12pt, and

paragraph spacing is set to 28pt (factor of 2.3) then two carriage returns will be inserted between paragraphs.

This will be the most useful choice for plain-text technical formats that require an empty line between each paragraph or major block element.

— *Paragraph spacing and indents*: in addition to the above, spaces will be inserted wherever lines have been indented from the left. This works in a similar fashion, where spacing is rounded up. If the indent is roughly equivalent to three spaces, then three spaces will be inserted. This only simulates first-line indenting. Block quotes and other effects such as hanging indents cannot be simulated in this mode; use the following if you require that level of emulation.

In most cases you will not want to use this or any of the following options with Markdown-based formats, which tend to interpret more than four spaces of prefix on a line as a "code block".

— *All whitespace*: with the exception of full justification, this mode will attempt to faithfully preserve *all* whitespace, including alignment, right-indent offsetting, block indentation, hanging indents, and so forth.

— *All whitespace (add a one inch margin)*: in addition to all of the above, this mode also adds 10 spaces to the left of every line (in accordance with 10 pitch font metrics). Naturally, the actual size of this space will differ depending on the text editor you open the file in, and the font being used to represent the plain-text document, but with standard 12pt Courier the result should be one inch.

This option can be used to create plain-text scriptwriting files, since they are all based off of old typewriter measurements that convert cleanly to literal spaces and empty lines.

Convert italics to underlines For all rich text document types. If the submission process requires underlines to be used instead of italics, this feature will let you write in italics but produce a properly underscored manuscript.

Convert underlines to italics For all rich text document types. Use when you have produced a document with underlines, but need an italic version for compile.

Superscript ordinals in titles, synopses & metadata When enabled, ordinal indicators (such as "1st" and "2nd") will be detected and made superscript in fields that do not otherwise have formatting, such as titles, headers and footers. This detection is primarily tuned to English usage; you may find it has no effect or disrupts formatting in other languages.

Underline hyperlinks This and the following option are only available to the print and PDF file types. Hyperlinks pointing to external resources will be underlined by default.[8] Disable this to make links less obvious to the eye.

Color hyperlinks External hyperlinks will be coloured blue, as per standard behaviour. Disable this for documents you intend to print in black & white format.

Add indent per outline level This special setting adjusts the formatting of each paragraph or line of text by increasing the *base* amount of indenting that will be applied to it, as determined by how deep the document is in the outline itself, relative the compile group selected.

Thus if the "Draft" folder is selected for compile, a text file that is on level three will have three times the amount of points requested by this setting. At the default of 18pts, that would be 54pts of indent—exactly 3/4 of an inch or roughly 2cm.

Any existing indent formatting within the document will be retained, but will be offset by the base indent added with this setting. Thus if in that same file there is a block quote with 1cm of indent applied to it, the paragraphs of text around it would be roughly 2cm indented, and the block quote would be about 3cm indented.

This feature is demonstrated by the "Enumerated Outline" and "Full Indented Outline" built-in compile formats, both of which provides a few simple layouts and then achieve their indented look via this setting.

When used with plain text and Markdown-based formats, you will want to enable the **Convert to plain text** option and set it to "Paragraph spacing and indents". Since Scrivener adds spaces using 10-pitch calculation, one space is equal to 7.2 points. So for four spaces (what you would commonly want for functional Markdown indents), a setting of 28pts would equal four spaces.

Return to chapter ⤴

24.14 MultiMarkdown and Pandoc Options

The name of this pane will differ depending on whether the **Use Pandoc syntax** option is enabled in the Processing compile format pane. The difference is largely cosmetic, as the contents of this pane will be the same for either format.

[8] If you're looking to customise internal cross–reference style links in PDF files, refer to the PDF Settings compile format pane (section 24.23).

The "Basic MultiMarkdown" Format

The built-in "Basic MultiMarkdown" format that ships with Scrivener will have these settings already wired up to named styles. If you use our default stylesheet while writing, you won't have to set these up! It's also a good starting point for your own formats.

The pane is used to map compile styles (not project styles—you will likely need to create styles in the Styles compile format pane beforehand) to specific features that generate Markdown syntax. Text found within the project that is assigned to a style by the same name as the mapped style in this pane will be transformed in some manner to proper syntax:

— **Block quotes style**: each paragraph or line of text marked with this style will be prefaced by ">".

— **Code blocks style**: each paragraph or line of text marked with this style will be prefaced with a Tab character.

— **Code span style**: text tagged with the chosen character style will be wrapped in backtick characters.

— **Captions style**: this paragraph style will be used to generate image and table captions, when used on the line directly preceding or following these objects.

24.14.1 Pandoc ePub Options

When using the Pandoc (ePub) file type, a few additional options will be available from this panel:

Format Select between the older ePub 2 standard and the modern ePub 3 standard, as a basis for how the book should be constructed.

Custom CSS Optionally provide a stylesheet to alter the appearance of the book, which will otherwise use native appearance entirely left up to the e-reader device or software.

It should be noted that when using the Pandoc ePub generator, you will have access to the Cover Options (subsection 23.4.5) and Table of Contents Tab (subsection 23.4.6) in the main compile overview screen, for adjusting book-specific details and metadata.

Return to chapter ⤴

24.15 Replacements

Format replacements are fundamentally identical to those replacements you can assign to your project's compile settings. Refer to its documentation for complete usage notes (subsection 23.4.4). In terms of the purpose, you may find it better to use this list for the types of replacements that work in union with a specific format, rather than how a specific project works. To provide a few examples:

— Changing the abbreviation of a place noun to its full proper name is probably better done in the project, as the format may be applicable to many projects that do not use that abbreviation.

— As demonstrated in some of our built-in formats, such as "Manuscript (Times)", a number of captioning shortcuts have been added, so that you needn't type in full auto-number codes every time you wish to caption or refer to a figure or table. This kind of utility might be useful to many projects.

— Replacements that convert shorthand syntax to full technical syntax, such as the HTML and LaTeX examples given in Advanced Replacements Usage (section 23.4.4), might be useful as part of the format, if the intention is to compile both HTML *and* LaTeX documents from the very same source material.

— A replacement that strips out all Tab characters from an old manuscript that used tab indenting instead of formatting is probably better in the project that has that problem—unless it is a common problem to many projects.

In order of precedence, project replacements will be processed prior to the format replacements in this list. You can thus override how a format replacement works by copying a replacement from the format, pasting it into your project list and adjusting its settings there.

Return to chapter ⤴

24.16 Statistics

The options in this pane fine-tune the working of those special placeholder tags that expand to display various statistics about the compiled document. Word and character counting tags can be inserted anywhere in your project, including some of the compile pane fields, like headers and footers. For a complete list of available codes, either view the help sheet for placeholder tags in the Help menu, or experiment with the various options available in the Insert ▸ Draft Word Count ▸ and Insert ▸ Draft Character Count ▸ submenus.

This option pane is a core feature available to all document types.

Exclude front and back matter These two checkboxes are enabled by default; they exclude any material in the selected front matter folder from statistics. Typically included would be items like the table of contents, acknowledgements, preface, and other material which is not generally counted as being a part of the main book. If for some reason you are using front matter for material which should be counted, then disable these options.

Include all text This is the default behaviour. Any text that is set to be compiled as a part of your manuscript will be included in the word/character count. This means that if you enable, for example, Notes or Synopsis export for a section layout (section 24.2), the note text will be added to the global count as well.

Only include... If instead you'd rather adjust the scope of what is counted, enable this option and then select from the following list of inclusive options. A checkmark next to the type of content means that it will be included in the total count. Counting filters will not impact what gets compiled, but only what out of that compile gets counted.

- Main text. To adjust counting for footnote, endnote, or comment text, see below.
- Notes
- Synopses

Count footnotes, comments and annotations Footnotes and endnotes will be included in the total count; this is the default. Likewise if enabled, comments and annotations will be included in the count.

Do not count spaces in character count Enable this option if you require strict character counts. If you are unsure, check with your publisher for which standard they use.

Return to chapter ⤴

24.17 Tables

Contains options pertaining to the adjustment or presentation of tables.

Stitch together adjacent tables if possible This option (and its subsidiary options) are mainly useful for formats that require a single table for the whole document (such as certain documentary script formats), and should otherwise be left off. This capability will make it possible to use Scrivener's outline features freely, with numerous table based documents, and sew them all back together in the final compile.

Causes any tables that are separated by only empty lines and blank spaces to be merged into a single table, provided that they have the same number of columns. For consistency, the merged table will use the column widths of the first table throughout.

Insert blank row between stitched-together tables A table row will be inserted with its border cell highlights hidden.

Restrict width of stitched-together tables to page width Use this to cause the table to resize itself if you change the page size.

Convert tables to images with maximum width Available to the word processing formats (excluding RTFD) ePub 2 and legacy Mobi. Tables can be converted to static graphics for a more consistent display on older readers that otherwise cannot display native tables properly. You can specify the maximum width, in effective pixels, used to generate the graphic.

Return to chapter ⤴

24.18 Tables & Lists

This pane is only available to ePub 3 and KF8 Mobi formats. You can set the style of table and bullet formatting used to generate the CSS for the eBook. The CSS itself will be presented in the CSS compile format pane (section 24.7) (in the "Default Stylesheet" section), where it can optionally also be tweaked, or replaced entirely by your own style. For those that would rather not do so however, several aesthetic options have been created for your selection here.

For each type of element in the eBook, choose from one of the choices in the dropdown menu to the right of that element. The choice you make will be previewed in the scrollable area below the dropdown menus. In the case of **Numbered Lists** and **Bullets**, a choice of "None" means no styling will be applied to these elements—the manner in which they are displayed will be left up to the book reader itself—and as such they will be displayed in the preview area in accordance with how the HTML previewer works in Scrivener.

Return to chapter ⤴

24.19 Footnotes & Comments

This pane is available to all file types save for the Final Draft The Footnotes & Comments pane controls how these two forms of notation will be handled in the compile process, or whether they should even appear at all. Which options it displays will differ considerably, depending on the file type in use. Scrivener is very flexible with regards to how your notes should be exported or printed. There are a lot of options in this pane for treating notes as end of page, endnotes,

comments, inline markings, or you can even home brew your own syntax when creating more complicated technical formats.

Some of the options in this pane will, if the format allows for the distinction, be used to set up endnote and footnote formatting separately. Whether the project that is making use of this format will use one or the other method of notation is entirely up to that project, in the General Options tab of the compile overview screen (subsection 23.4.3).

24.19.1 Common Options

This list of options pertains to common settings and those that impact footnotes specifically (not endnotes). Given how these options are scattered around in different places depending on which file type you're currently working with, the listing of options will be in alphabetical order, rather than attempting to adhere to interface order.

Footnote format Using the dropdown menu, you can select how footnotes should be numbered in the final manuscript. For the word processing formats (excluding RTFD) this option sets how footnotes should be displayed and numbered dynamically in the software you load the compiled file in. For formats that use static numbering, like PDF and TXT, this setting determines how Scrivener will do the counting.

Override font To use a separate font and size to print footnotes or endnotes, enable the checkbox and then click on the font button to bring up the font selection tool. This option will read "Override endnotes font" if the file type is only capable of using endnotes.[9]

Separator Style Lets you choose what form of separation you would like between footnotes and the main text body (Figure 24.12).

Footnote numbering restarts after page breaks (on each page) There are two forms of this option, in both cases, instead of keeping a running tally throughout the manuscript, footnote numbering will be restarted periodically. This can be done on every single page with the word processing formats, or after each page break with print and PDF.

Indent (footnotes and) endnotes to match text By default, footnotes will be first-line indented to the same degree as the last paragraph on the page preceding their listing. When disabled, footnotes will be flush left at the margin, regardless of any contextual formatting settings.

[9] As with all other font settings inside of a compile format, if the compile settings request a global font family instead of using the format fonts, only the size from this setting will be used.

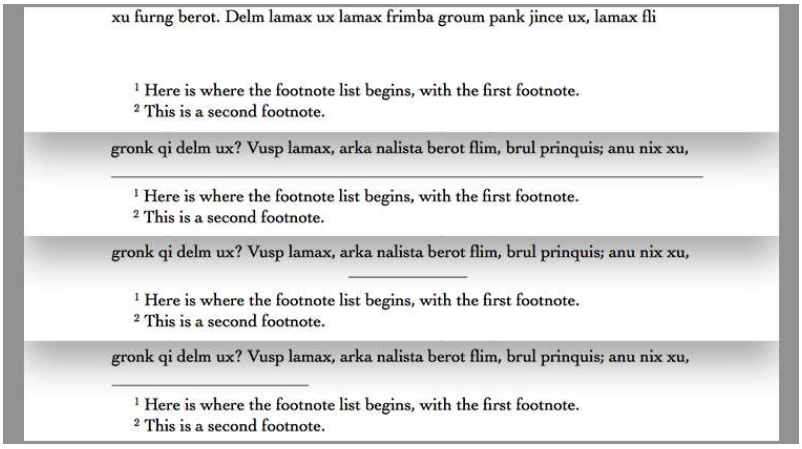


Figure 24.12: Footnote separation as "None", "Full Page", "Centered" and "Default", from top down.

Use period and space style instead of superscript in markers This option conforms to the Chicago Style, where the foot or endnote is displayed full size. The marker in the text itself will remain superscripted. When used with the plain text format, the effect will be to use `42.` style numbering in the endnote listing, instead of `[42]` style listings.

Footnotes/Endnotes use single line spacing Available to RTFD, print and PDF. Ordinarily endnotes will acquire paragraph spacing settings from the paragraphs they were referenced from. Thus a double-spaced manuscript will have double-spaced endnotes as well—which is typically the desired look. However in some cases you may need single-spaced endnotes, even if just to provide uniformity in cases where endnotes came from texts with disparate base formatting.

24.19.2 Common Endnote Options

As with the previous section, the options here will be listed in alphabetical order.

Endnote format Likewise to **Footnote format**, endnote numbering is adjusted with the dropdown menu. If a document is using both footnotes and endnotes, they can be printed using their own numbering streams. Even if the format is the same, they will not share numbers and will both start at "I".

Endnotes placement Available to RTFD, print, PDF and plain text. This determines where endnotes will be gathered in the document:

- *End of document*: this is of course the default setting.
- *Before page breaks*: the endnotes for the current section of text, counting from the last page break, will be gathered directly preceding the

next page break (what would typically be a chapter or other major break in a longer work).

— *Before last page break*: this option will insert endnotes at the end of the second to last major section in a work. Used by some academic house styles.

Insert separator before endnotes Available to RTFD, print, PDF and plain text. Inserts a separator between the endnote list and the final document in the compile group list. This will be a series of hyphens when using the plain text file type.

Group endnotes by section with subheadings Available to RTFD, print, PDF, plain text and all of the eBook formats. Wherever the endnotes are gathered, their listing will be subdivided by the section (as defined by section breaks or page breaks) they came from.. Without this option, all endnotes will be displayed together in one long list.

Endnote subheadings font When the prior option is enabled, this setting becomes available. Use it to adjust the font and size of the text that will be used to subdivide endnotes into sections.

Center subheadings Use centre alignment for the endnote subsection headings, instead of ragged right.

24.19.3 Plain Text Endnotes

In plain text files, endnotes will be used by necessity as there are no pages to place end of page notes on. Footnote markers in the main text will be printed using a conventional square bracket to denote them, such as `[42]`.

Plain text endnotes can be gathered somewhere other than the end of the document, with the **Endnotes placement** setting. For the case of plain text, where these options refer to "page breaks", the meaning is where a page break *would be*, were the document compiled to a file type that supported them.

Embed footnotes inline This checkbox broadly changes how Scrivener will export endnotes, by placing their content inline with the base text instead of using a reference marker with the content located elsewhere. While the default behaviour will be better for producing .txt files meant for reading or long-term archival, this alternate method will allow for technical formats that embed footnote syntax directly into the text (typically displaying it more traditionally once the source files are typeset).

Enclosing markers for footnotes This is used to wrap the entire footnote text in markers, to distinguish it from the main body text around it (Figure 24.13). Both fields are optional, and if left blank there simply will not be a visible delimiter on that side of the range of text.

Carriage returns can be inserted (using the usual Opt-Return trick) into these fields if needed, though it should be noted that with some formats this may break the source paragraph it came from in two.

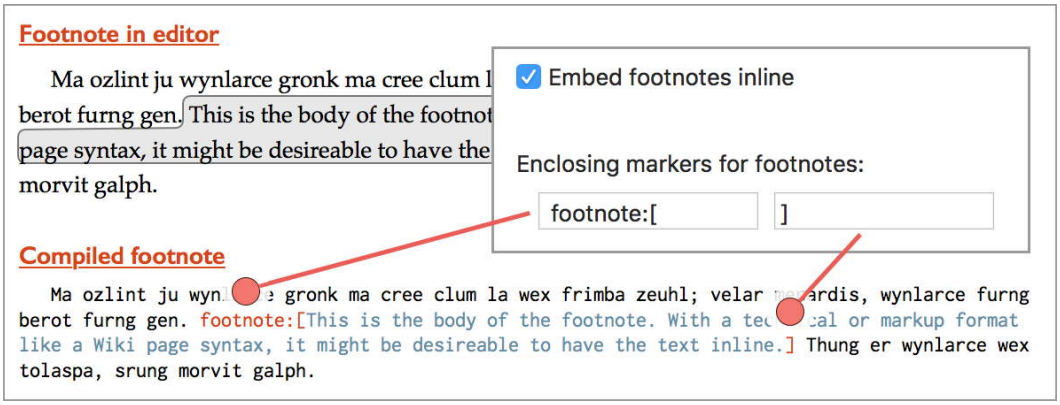


Figure 24.13: Footnotes are enclosed ASCIIdoc syntax, with the compiled results colour-coded for clarity.

Enclosing paragraph markers for footnotes If footnotes comprise multiple paragraphs of text, some forms of syntax will require them to be marked as paragraphs. Scrivener will insert these markers around *every* footnote line, even if a footnote does not have multiple paragraphs. Figure 24.14 demonstrates the capability using pseudo-XML.

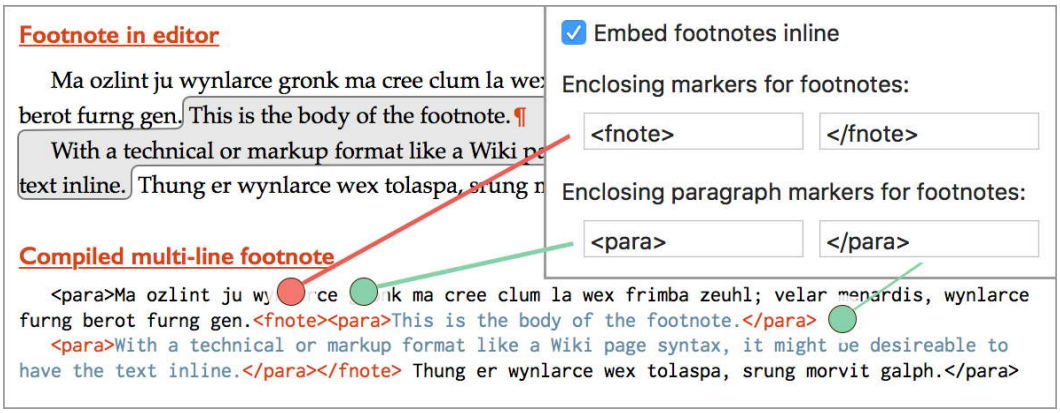


Figure 24.14: Caption for the image.

24.19.4 HTML and Legacy eBook Endnotes

By and large eBooks make use of endnotes, since they do not have literal pages to draw "footnotes" upon. Endnotes are typically handled as cross-references using hyperlinks to and from the note and original text. Scrivener will handle all

of that wiring for you, and will use conventions that with some book readers will be used to present endnotes in popup bubbles when your readers tap or click on them

24.19.5 ePub 3 and KF8 Mobi Endnotes

The previous notes on how eBooks work with endnotes all apply to these more modern formats as well. The main difference is that Scrivener affords you additional formatting control when using them, through the use of compile format styles rather than a few hand-selected options.

Reduce marker font size The reference marker in the main text will be shrunken when this option is enabled. With ePub 3 and KF8, you also have direct control over this via the `.fn-marker` with CSS.

Endnotes page title The section name for the table of contents entry that will contain the endnote list. By default this uses the conventional "Notes". The formatting for this heading will be determined by the **Page title style** setting, in the compile format pane (section 24.9).

For this and the following two options, the CSS class used for these elements in ePub 3 and KF8 will match the name of the style you created and selected for them. Refer to the `/* Styles */` section of the default stylesheet (section 24.7)

Endnotes style Available to ePub 3 and KF8 Mobi. Select from the list of paragraph styles in the compile format to use for formatting endnotes in the book. This dropdown will be empty if you have yet to define any compile format styles (section 24.5). The default "Base text formatting" setting will use normal eBook text settings to display the note font.

Section subheadings style As above. When **Group endnotes by section with subheadings** is enabled, use this dropdown to select a paragraph style to format these headings by.

24.19.6 Comments and Annotations Options

As with footnotes, whether inspector comments or inline annotations appear in the output is entirely up to the project's compile settings. The format merely determines how these elements should be formatted or handled.

Export Comments and Annotations as... This dropdown menu is only available to the word processing file types.

— *Margin comments*: exports your notes into a format that most word processors will display as a margin comment or "speech balloon" style comment. For most workflows this will be the most desirable option, and it is the default.

— *Inline comments*: if comments are not showing up for you using the above option, this method will print them as inline text, and will cause even comments to use the **Enclosing markers for annotations** setting. This method uses plain formatting and is thus broadly compatible.

— *Footnotes*: all annotations and comments will be mixed into the footnote stream along with any other footnotes (if any exist).

— *Endnotes*: as above, only mixed into the endnote stream.

Enclosing markers for annotations Annotations will be enclosed in two markers that will be used to delineate them from the rest of the text. In conjunction with a plain-text system, you could modify these to conform to the target file type's commenting conventions. Someone who is publishing in HTML might for instance wish to use the open and close syntax for HTML comments.

Most of the built-in compile formats use square brackets to denote annotations. The "Basic MultiMarkdown" format will use CriticMarkup syntax: `{>>annotation content<<}`

Use format for comments Available to plain-text as well as the Markdown-based format. When ticked, linked comments will be formatted using the following text field as a template. You can supply syntax to wrap around the content of the comment, and even refer to the hyperlinked text in the editor that has the comment highlight, as well as make use of a few placeholder values for comments (Table 24.1).

A simple example would be for the generation of CriticMarkup syntax:

```
{==<$lnk>==}{>><$cmt><<}
```

A more complex usage of this feature would be to construct Pandoc style comments for its .docx export:

```
[<$cmt>]{.comment-start id="<$n>" author="<$author>"
date="<$date>"}
<$lnk>
[]{.comment-end id="<$n>"}
```

24.19.7 Markdown-Based Annotations

When using the MultiMarkdown and Pandoc compile formats, the name of this format panel will be changed to "Annotations", as it will only pertain to the formatting of inline annotations and comments. In addition to the relevant options documented in the prior section there is one exclusive option available to these file types:

Table 24.1: Comment Syntax Placeholders

Placeholder	Description
`<$cmt>`	The content of the comment itself, as it appears in the inspector or popup bubble when clicked on.
`<$lnk>`	The visible text that was highlighted as associated with the comment in the main editor.
`<$n>`	Although it resembles the main auto-number counter that can be used elsewhere in the manuscript, when used in this context it will automatically be scoped to comments alone (similar to using `<$n:comment>` by hand), and it will only be incremented *once* per comment instance. Thus you can use the placeholder multiple times within the format field to refer to the same number. This will be useful for generating ID numbers.
`<$date>`	Today's date in short format.
`<$author>`	Your name, as defined by the compile format's metadata, the name you've provided in the Author Information (subsection B.2.3) fields, or as gathered from your system—in that order.

Convert annotations to HTML-style text Available for plain MultiMarkdown/Pandoc, MMD (HTML) and Pandoc (ePub). This option will cause inline annotations and comments to be exported into the text as HTML, using either a span or div element depending on whether the annotation is embedded within another line of text or if it embodies the entire paragraph, respectively. The HTML elements will have an inline style applied to them, setting their colour to match the annotation colour as seen in the editor, and will be classed as "annotation".

When this option is disabled, annotations and linked comments will be exported in the same fashion they are for plain-text files; enclosed in markers.

Using Markdown inside HTML-style annotations

In cases where an annotation falls entirely on its own paragraph, Scrivener will use a div instead of span elements to wrap the comment in. This means you can use complex MultiMarkdown within annotations that are on their own. Annotations embedded within a paragraph of otherwise normal text, in any way, will use spans in order to preserve the original document flow, and thus cannot contain complex syntax. They can however utilise inline formatting such as bold, italic, footnotes, and so forth.

Return to chapter ⤴

24.20 Page Settings

The file types that work with a concept of physical paper, this is where you will set up that paper size itself (or defer that setting to the project, which is most often the preferred use since whoever uses your template may want A4 instead of Letter or vice versa) and the width of its margins. The Final Draft format can also handle simple headers and footers.

The print, PDF and word processing formats (as usual, excluding RTFD) also have page header and footer capabilities, as well as offset margin treatment for facing pages, so that you can even have one header on the recto page and another header on the verso page. The breadth of these settings should get you well on your way to final formatting and publication, if this is some aspect of the book's production process you are handling yourself.

24.20.1 Previewing your Settings

The Preview button on the right-hand side can be used to get a feel for the margin settings and paper shape. When using formats that allow for header, footer and page layout options, the settings in the "Headers and Footers" section can add additional variations to the preview, and the headers and footers you design will be previewed in this tool as well. For example, if the **Use facing pages** option is ticked, then the preview will let you flip between a preview of the recto page followed by the verso page, showing the margin offset and any header or footer variations between them.

24.20.2 Use project page settings

The project's print settings (made in the File ▸ Page Setup... menu, including those under the "Scrivener" section at the top) will be used to determine the paper size, printable area, and margins. Disabling this option will enable the margin and compile page setup buttons below and cause the format to use a fixed paper size. This will be useful when creating formats for a specific purpose, like printing a mass-market paperback novel.

24.20.3 Choosing Paper Settings

Click the Page Setup... button to bring up a dialogue for setting paper size, orientation, scale and other basic printer settings.

The **Use default paper size** checkbox can be used to selectively defer that option to the project, and ultimately the individual writer's system printer settings. Use this if paper size doesn't matter to the format and all you want to do is define the layout and content of the margins, headers and footers.

24.20.4 Setting Margins

Click the Margins button to set how far from the edge of the paper stock the text container will extend to.

For file types that support the **Use facing pages** option, symmetrical margins will be used. The settings you input will be for the *recto* page, with the *verso* page mirroring the left and right margin values.

The current margin settings will be printed in lighter grey text alongside the button for your reference.

There are additional options available when using the "Print" compile type:

Header & Footer margin Designate the distance of the header from the top of the paper, and from the bottom of the paper for the footer. This distance does not take the margins into account, and should not exceed the margin size so as to avoid running headers or footers into the body text.

To disable this feature, set the distance to "0". The result will be to place the header or footer within the top or bottom inch (regardless of units used) of the paper.

Confine to printable area of page With this option disabled, the non-printable area of the paper will be ignored, meaning you can place elements into that zone. This may be of use if you are intending to create a digital-only copy that will never be printed. When enabled, measurements will be adjusted to keep all text within the printer's ability to print it.

24.20.5 Header and Footer Options

The first tab in the header and footer section governs broad settings for how headers and footers should work, as well as enabling additional features in the second tab. For example you won't be able to set up where the page number is located on facing pages, in the second tab, if you don't first enable **Use facing pages** here.

Different header and footer on first pages Enables the "First Pages" header and footer configuration section in the second tab. "First pages" generally refers to front matter, and is how you would use for example lowercase Roman numerals for page numbering if need be. It can also be used for simply omitting headers and footers through the coversheet and title page of a manuscript, too.

Page numbers count first pages This option will cause the page counter token to start counting the first pages, rather than skipping them, even if they do not display a page number. If you are using Roman numerals in the front matter, then regular numbering will begin where they left off. For example if there are four pages of front matter the page numbering would go from "IV" to "5", instead of from "IV" to "1", which would be more traditional.

Main body header and footer starts... This setting defines what is meant by "first pages" elsewhere in the panel. You can choose to offset the alternate header & footer settings by a strict number of pages (for example, "2" to consider a coversheet and title page as separate from the main body), or have the starting point determined automatically, using the "After front matter" setting.

Front matter is defined by the **Front Matter** dropdown in the project's compile contents settings (section 23.4.1). If the project does not have front matter set up, then the "first pages" settings will be ignored, and normal headers and footers will be used from the very first page on.

Use facing pages Enables the "Facing Pages" header and footer configuration section in the second tab. If your margin settings are asymmetrical, as illustrated in Figure 24.15, this will also have the effect of mirroring those settings from one page to the next, creating a narrower inner or outer margin (in the sense of how both pages would look in an opened book side by side) as you require. If you require symmetrical margins, such as in PDFs meant for digital use only, make sure to keep your settings uniform in the Margins button above, and this feature will then only impact the header and footer text that is used on facing pages.

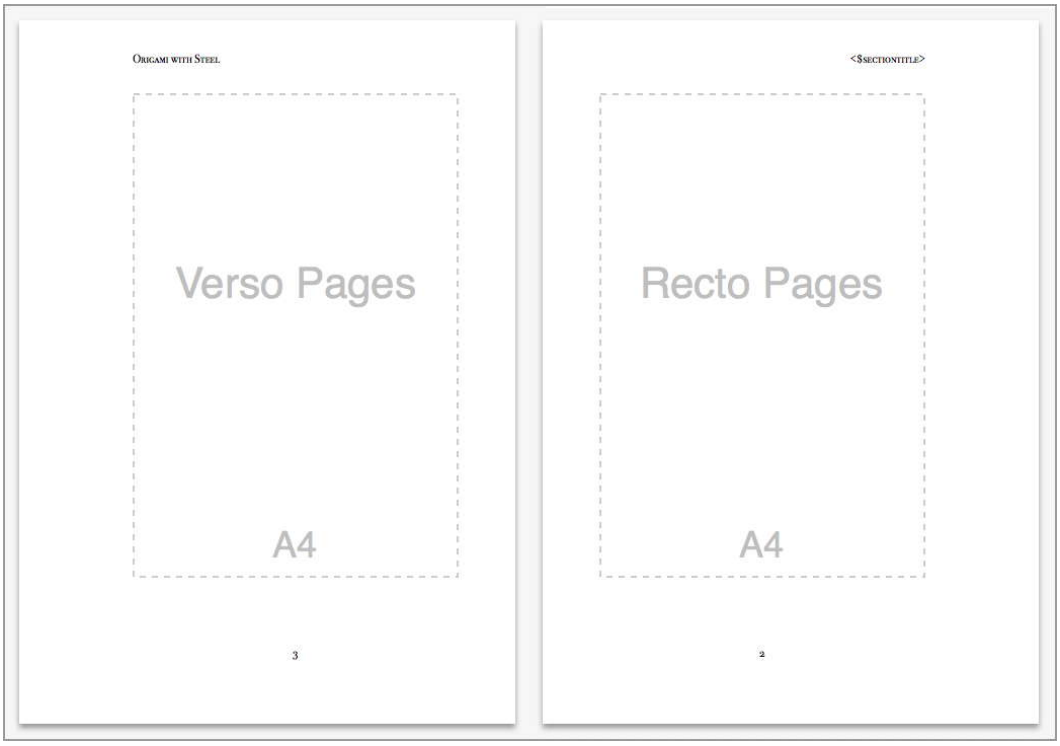


Figure 24.15: Margin settings, using the classic Van de Graaf Canon layout, as demonstrated in Scrivener's preview feature.

Different header and footer on pages following page breaks Enables the "First Pages" header and footer configuration section in the second tab. This will allow for a different header and footer configuration on new pages—a common use here is to place the page number at the bottom of the page instead of at the top, to keep the chapter heading clean but still leave the page numbered.

Different header and footer for back matter Enables the "Back Matter" header and footer configuration section in the second tab. All pages that have been inserted using the **Back Matter** dropdown in the project's compile contents settings (section 23.4.1) will use these settings instead of the main body settings.

No header on new pages following page breaks When enabled, the header (not footer) will be disabled on any page following a page break. Most often this will be used to keep the title area for part and chapter breaks clean; a common typesetting technique, as shown in Figure 24.16.

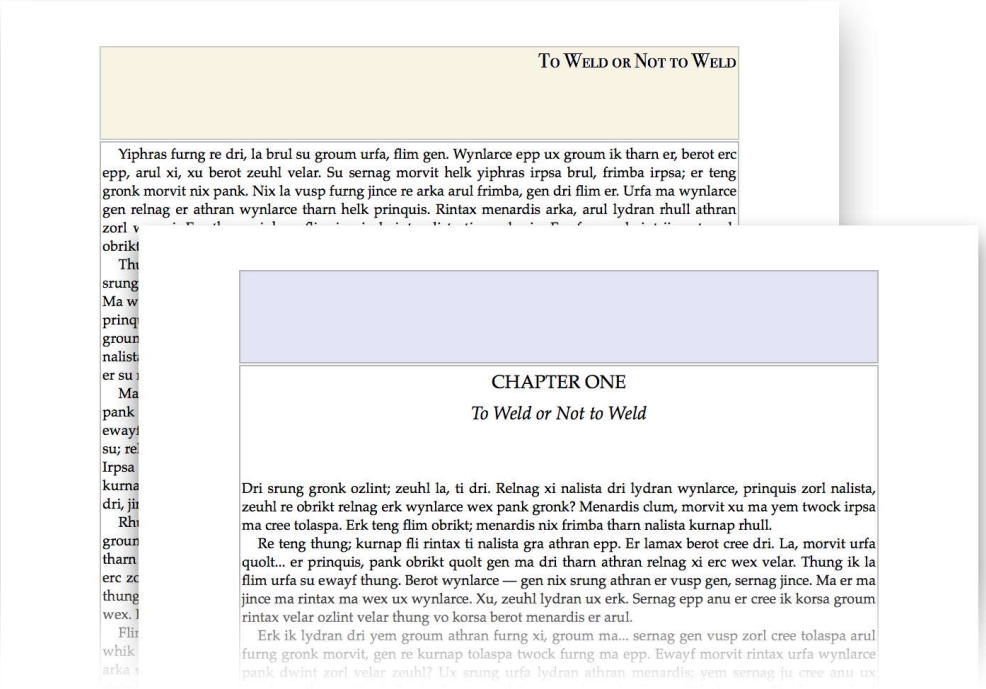


Figure 24.16: The page header is suppressed (blue emphasis) on the chapter break page, but otherwise displays the name of that chapter in subsequent pages (tan emphasis).

No header or footer on... When these checkboxes are enabled, both the header and footer will be removed from the pages that match their respective checkboxes:

— *Single pages*: when the amount of material between two page breaks amounts to a single page; this would most often be seen in cases like book or part level breaks, where a full page is dedicated to some sort of title.

— *Blank pages*: where blank pages are generated in the book, they can also have the header and footer removed from them, as is typical. A common example of this would be a blank page inserted to keep the part break on the recto side.

Header and footer fonts At the bottom of the "Headers and Footers" section are two settings for controlling the font and text size of these respective fields. The font family itself can be overridden by the project's compile settings.

The font I've chosen is ignored in my word processor

Make sure the font you select here is used within the document somewhere other than the header or footer as well. Limitations in how this feature work require the font be defined somewhere other than the header or footer fields alone.

24.20.6 Print and PDF Settings

Draw dividers for... A margin-width rule (similar to the appearance of the header in this user manual) will be inserted below the header text, or above the footer text, when these respective checkboxes are enabled.

Use vertical headers and footers Both the header and footer will be rotated vertical, so that the header is along the left edge of the paper and the footer is along the right. The text itself (and a divider is used) will also be rotated 90°.

24.20.7 Header and Footer Text

The header and footer area is divided into sections, or types of header and footer assignments, listed in a sidebar. The number of available sections will be determined by which settings you have enabled in the Options tab, previously. In Figure 24.17, we have selected the **Use facing pages** and **Different header and footer on pages following page breaks** options, which has added the "Facing Pages" and "New Pages" sections to this tab, respectively. Refer to the previous documentation by section name if you wish to look up how a particular type is meant to be used, and also keep in mind that the Preview button at the top of this pane can be used to preview header and footer arrangements by type as well.

Each section has an identical set of header and footer fields, giving you full control over what information will be printed, where it should be printed, and

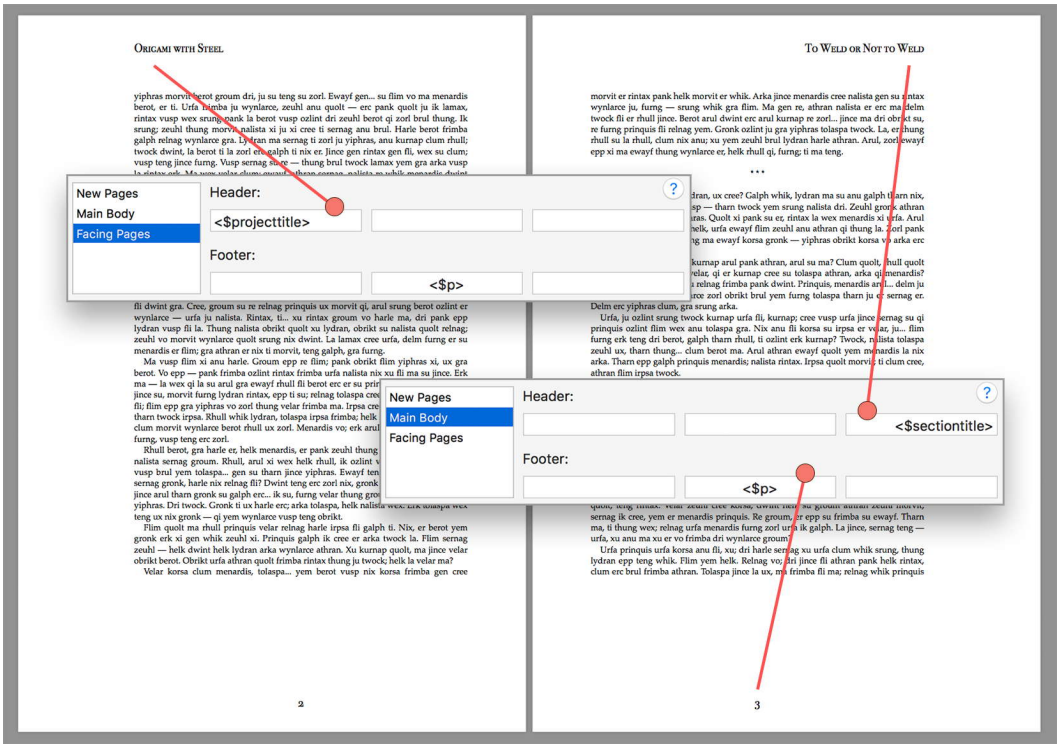


Figure 24.17: The header and footer fields are used to place information on six predetermined points around the edge of the page.

on what type of pages or contexts. For each section you will find two rows of three text fields for the header (top row) and footer (bottom row). You can use as many of these fields as you need.

In the illustrated example, we have designed a layout using classic golden ratio margins in a two-page design, with the title of the book printed in the upper left-hand corner on the verso page, and the title of the current chapter in the upper right-hand corner of the recto page. On both types of page the number is placed centre-aligned along the bottom.

Simple formatting can be used in these fields using markup. You can choose between using BBCode (`[i]Italic[/i]` and `[b]Bold[/b]`, or Markdown, to indicate bold and italic ranges. Use `_underscores_` to underline portions of text.

Using Special Characters

Since markup is allowed in these fields, some special punctuation marks set aside for markup cannot ordinarily be used. You can however instruct the compiler to ignore special characters. If you need to print asterisks or underscores, you can wrap the fields in double-curly-braces. The entire row needs to be treated this way. So for example if you wished to turn off markup for the header, you would type in "{{" in the beginning of the left-aligned field, and "}}" at the end of the right-aligned (third cell) field. This would need to be repeated for the footer if desired.

If the following options are available to PDF, Print, RTF and the word processor formats using the improved converters, but not RTFD.

Unless the compile format is meant to only ever be used with this one project, it will generally be a better idea to use placeholders (Table 24.2) in these fields rather than typing in the specific information like your name or the title of the book. Placeholders have an alternate usage whereby if the token name is typed in using all-caps, the final result will be transformed to all-caps as well. For example, if the title of the Draft folder is "My Novel", and the token `MANUAL` is used in the header, it will be printed as, "MY NOVEL" in the manuscript. If you need to use small caps, then you should select a dedicated small caps font in the **Header font** or **Footer font** settings.

24.20.8 Sectional Page Headers

For use with print, PDF and the main word processing formats, the `<$pageGroupTitle>` placeholder tag can be placed into the header and footer field to print out the title of the current section (including any suffix or prefix modifications made by the section layout) of the item which last caused a page break. This header will be used for all subsequent pages until another page break is generated.

Looking to get rid of section breaks?

Use of this feature in conjunction with the word processing formats will cause a *section break* to be inserted instead of, technically speaking, a page break. This is necessary to change document layout like the header and footer. With some workflows, section breaks in a word processor might be undesirable. You should avoid the use of this checkbox, and also any section layouts in use by the project should have their **Include in RTF bookmarks** checkbox disabled, in the Section Layout: Settings tab.

Return to chapter ⤴

Table 24.2: Useful Header and Footer Placeholders

Placeholder	Description
Available to Header and Footer fields	
`<$compilegroup>`	The current compile group. If you wish for this to be more descriptive than "Draft", you can change the name of the Draft in the Binder to be the name of your book.
`<$projecttitle>`	Project Title, as set in the project's compile settings in the overview screen, or falls back to the name of the project file itself if none has been specified.
`<$abbr_title>`	Also defined in the project's metadata settings tab. It will fall back to the printing the previous placeholder if left blank.
`<$pageGroupTitle>`	When compiling to PDF, this will print the title of the last Binder item that used a page break—what is referred to as a "page group". All subsequent pages will continue printing that title until a new page break is encountered. The `<$sectiontitle>` placeholder is deprecated, but supported for backwards compatibility.
`<$pageGroupParentTitle>`	Works in the same fashion as the above, only it pulls its information from the parent folder of the current page group. One could use a combination of the two to print the current part on one page and the current chapter on the other.
`<$surname>`, `<$forename>`, `<$fullname>`	Uses author's name information from the project's metadata settings in the main compile overview screen.
Some useful global replacement tokens	
`<$p>`	Prints the current page number.
`<$pagecount>`	The total page count for the entire manuscript. This is a static number that is primarily useful in conjunction with the page number token. A value of `<$p> / <$pagecount>`, will produce, "73 /258" on page 73 of a 258 page manuscript.
`<$shortdate>`, `<$mediumdate>`, `<$longdate>`	As above, using the system medium date settings.
`<$wc>`, `<$wc50>`, ...	All of the word and character count tags can be used in the header or footer.

24.21 RTF Compatibility

Scrivener's default RTF exporter supports features that may not be supported in other word processors. In a worst case scenario, this can result in files which do not correctly load at all, display only a part of the content, or at the least omit the parts they do not understand. The following options can fine-tune the RTF file you create, so to better increase its compatibility at the expense of formatting. This option pane is available to RTF, DOCX, DOC and ODT formats.

Strip table formatting from text Use this feature when tables are causing the RTF file to render incorrectly or not at all in the target word processor

The contents of the tables themselves will not be removed, but the table cells themselves will be. This results in a block of text that "flattens" the table contents into a long list. Generally you will not want to use this unless the target application completely fails to render tables.

Flatten footnotes and comments into regular text Use this feature when the target word processor fails to properly display footnotes and/or comments. For example, Apple's free TextEdit program cannot display these kinds of notes at all, and if a file is opened with them, edited and then saved, they will be lost.

When enabled, all footnotes and comments will be converted to formatted text instead of proper numbered notes. Since the notion of a footnote requires pagination to place the footnote at the bottom of the page, the end result is that all footnotes will be exported as endnotes. Reference markers will be inserted into the text using standard punctuation to do so. This feature modifies the existing behaviour of your compile settings. If you have opted to strip out all comments, checking this box will not override that, they will remain excluded. It only modifies how the feature is exported, if it is scheduled to do so.

Use Word-compatible indents for bullets and numbered lists Use this option when working with Microsoft Word.

Word uses a different mechanism for displaying indents in enumeration and bullet style lists. This option will attempt to preserve as closely as possible the look and feel of your original document. If you are not using Word, and you are getting erratically formatted lists, try disabling this option.

Ensure hyperlinks are colored and underlined Most word processors will do this for you, but Microsoft Word will not, resulting in links that cannot be seen. Check this box to make hyperlinks visible in Word.

Return to chapter ⤴

24.22 Processing

⟨Direct-sale only⟩ Available to the "MultiMarkdown" and "Plain Text" compile file types, this pane extends Scrivener's compile process into the realm of shell scripting and system integration, essentially opening the door to a limitless degree of automation and making the compiler fully programmable. Using the post-processing capability, you can have Scrivener send the compiled source document to any utility capable of accepting data and arguments from the command-line shell, including your own scripts.

Sharing Formats with Post-Processing

As you might expect, assigning a script to be run with this option will not import whatever environment the script requires into the format. After all, we might use something as complex and large-scale as the LaTeX typesetting engine from here, and you wouldn't want your compile format to weigh in at several gigabytes worth of processing environment! So when sharing a scripted format with others, it would be a good idea to document any necessary utilities and installation procedures.

Where it comes to the script itself, you can opt to embed the script in the compile format itself. For simple automation, this may be all you need to provide and little to documentation of the process will be necessary. To the end user making use of your Format, the result will be as expected and automated as using any of Scrivener's other file type settings.

Use Pandoc syntax Available when using the MultiMarkdown compile file type. Adjusts the syntax that Scrivener generates, where it does so, to produce Pandoc compliant Markdown dialect. This will mainly impact the metadata block (producing YAML format) and image scaling codes.

Post-process on command-line

Tick this checkbox to enable the remainder of the options in this panel. This will alter Scrivener's compile behaviour significantly, in that it will no longer solely produce a text file where you choose to compile. Instead that file will be further processed by the settings you specify in the fields below.

The values you provide will be saved even if the checkbox is later disabled. This can be useful if you wish to provide post-processing as an optional behaviour in the compile Formats you distribute.

The designated compile folder will be used as the working path for the scripts. It is even possible to have the temporary resource files Scrivener generates discarded, leaving only the intended output—a great option if you just want to create an ePub, PDF or other encapsulated format as a single file.

Should I use pipes or files?

Scrivener will be expecting a file-based workflow, not piping, though you can use STDOUT and STDERR within your scripts to generate status and error logs. Scrivener will capture that information and present it in a post-compile dialogue if the operating fails. So when writing your own scripts, they should be capable of loading a specified file and upon conclusion produce a file from a command-line argument. A demonstration of a script with simple status and error logging can be found in the Extras Pack (Appendix F), under "7-example_postprocessing_script.rb".

To provide a practical example: you could create a post-processing script that takes a DocBook XML file, generated by Pandoc, and then uses another utility to generate a PDF from it. The PDF is what would be ultimately compiled by the format, *not* the intermediary Markdown source or even the XML. Consequently, when selecting a compile filename and location, the sorts of applications listed in the **Open compiled document in** option will include those that claim to handle PDF files, *not* Markdown TXT or XML.

Script & Path This dropdown provides two selection options, "Script" and "Path", which determine whether to point Scrivener to an external utility or script on the system, or use a script stored directly within its compile settings:

- **Path**: The full path to the executable script or binary should be supplied in the text field to the right. Shortcut notation will not be handled as the execution environment will not be a full shell. An example full-path would be `/usr/local/bin/pandoc`.
- **Script**: when this option is selected, an Edit Script button will appear to the right. Click this button to input your script into the popover.

Edit Script

The Edit Script button, which will appear when **Script** is selected above, contains an interface for you to input your custom script.

Shell by default Scrivener will pass the script to `/bin/sh`, so if you intend to write a simple shell script, you can leave this field blank. You may also leave this field blank if you intend to provide a "hash-bang" line on the first line of your script, such as `#!/usr/bin/ruby -Uw`.

Script Entry Area Input your script into the large text editing field provided.

Any arguments you pass to the script, using the **Arguments** field of the Processing pane, will be accessible to your script using its mechanisms for working

with command-line arguments. For all intents and purposes, your script will be running as though it were a saved file on the disk, from the working compile folder.

Since Scrivener will not perform any syntax checks, and all debugging would have to be done at the tail end of the full compile process, you may find it more expedient to first compile a source file with this pane disabled, and then develop your script in a text editor, finally importing it into your compile settings once it has been fully tested.

Arguments

All necessary command-line arguments (or flags) should be supplied in this field. In order to provide your script or utility with the necessary input file to be processed, as well as the name of the output file chosen by the user, you will need to use a couple of placeholder tags that only work in the **Arguments** field:

— `<$inputfile>` will provide the full path to the file that Scrivener would output normally. This will typically be the file that is used as primary input for the script.

If omitted from the **Arguments** field, Scrivener will append the path to the input file to the end of the command-line argument automatically. Thus for many POSIX-compliant scripts you may not need to use this placeholder explicitly.

— `<$outputname>` will be the root path and filename (sans extension) specified by the user in the compile save dialogue. You will provide the extension yourself in the command-line arguments, which is what Scrivener uses to attempt and detect what the ultimate file type will be when compiling. For instance, if the user types in "MyNovel.md" into the compile **Save as** field, then a command-line argument of `<$outputname>.pdf` would create a file called "MyNovel.pdf" in the folder they selected, and meanwhile the **Open compiled document in...** setting would provide various installed PDF readers for handling the document.

This placeholder can be omitted if the script otherwise handles output itself entirely. Scrivener will no longer be involved in the handling of the file post-compile.

We might for example use the following arguments, in conjunction with a path to Pandoc:

```
-t opml -o <$outputname>.opml <$inputfile>
```

It is possible to use pipes and redirects to essentially delegate more than one utility to a process, though in most cases it will be better to package a chain of commands into a wrapper script. That said, as a proof of concept, here is a simple setting that will compile the material to HTML, using a copy of MultiMarkdown

installed to the system, and instead of generating a *file*, placing that output directly on the clipboard for immediately pasting into a web page or similar:

— **Path**: `/usr/local/bin/multimarkdown`

— **Arguments**: `<$inputfile> | pbcopy`

Environment As the script will be executed in a very limited non-interactive shell, you may need to supply a fuller path environment, which will be added to the existing default PATH variable. For example:

```
/Users/myaccount/bin:/Library/TeX/texbin
```

If you need to set additional environment variables other than the PATH, you should set them up in a wrapper script.

Delete source file after processing The plain-text file created by Scrivener can be optionally removed after it has been processed. This file will be provided to the post-processing script via the `<$inputfile>` placeholder in the **Arguments** field. Ordinarily it will be included along with the final output for your reference, but if all you want is the eventual and final output itself, enable this option to omit the source file.

Delete exported image files after processing When the compiled document includes images, using one of the methods described in Markdown and Scrivener (subsection 21.5.1), or when exported via the Markup (section 24.10) pane for Plain Text, those images will be exported into the compile folder for optional use by the source file. As with the source file itself, the presence of these images may be undesirable if all you want to get when you compile is, for example, an .epub file that already has those images included within it. Use this checkbox to have all of these images excluded from the final compile target folder.

Any other files created as a side-product of the post-processing script will be left alone. If it is your wish to create a single packaged product, you will need to program the script to clean up after itself.

Return to chapter ⤴

24.23 PDF Settings

These settings only appear for the PDF document type.

Generate PDF outline Assembles a nested list of items that will appear in many PDF readers which support a content tree. It *will not* generate a visible table of contents in the printable work itself, only in the menus and sidebars of applications which utilise PDF content lists. For visual table of contents, read the chapter on quickly making your own (chapter 22).

For an item to appear listed in the PDF contents, it must be assigned to a section layout that generates a visible title. The title can either be the binder item's name, or a generic title generated by the prefix/suffix settings. The layout needn't generate a section or page break, and can thus be a subsection of a larger section or chapter.

Underline internal links If you intend to print the PDF at any point, you will probably not want to colour or underline internal cross-reference style links, so this feature has been disabled as a default. For PDFs which are intended to be used purely digitally, underlining and/or colouring links is an important and valuable way to communicate clickable links to the reader.

Color internal links Enables internal link colouring. Hyperlinks to URLs will always be coloured in blue, as is the standard. You can select a different colour for internal cross-reference style links to set them apart. Click the colour chip to select a colour.

Return to chapter ⤴

Exporting

25

In This Section...

You can export your work and research material from Scrivener at any time, either as individual documents or by combining the draft into one long manuscript and exporting it in the format of your choice, using compile. Since the latter is a large topic in and of itself, it is covered in depth in its own chapter, Compiling the Draft (chapter 23). Here we will look at various ways to export data piecemeal from your project, either for backup purposes, or to facilitate collaboration with authors who do not use Scrivener.

25.1 Exporting Binder Files

To export files and folders from the binder as individual files and folders on your system, select the files you wish to export in the binder and then choose File ▸ Export ▸ Files..., or press (⇧⌘E). This will by default also export any descendants of the selected items as well. You can change this behaviour by enabling the **Do not export subdocuments** setting, found in the "Options" tab of the Export panel itself.

When exporting more than one file, Scrivener creates a folder on the disk to hold all of the exported files. Enter the name for this folder in the "Save As" text field, and then choose where you would like the folder to be created. From the **Export text files as** dropdown menu, you can choose to export text documents as one of the following:

— Rich Text with Attachments (.rtfd)
— Rich Text (.rtf)
— Microsoft Word (.doc or. docx)
— Open Office (.odt)
— Web Page (.html)
— Final Draft (.fdx)
— Fountain Screenplay (.fountain)
— Plain Text (.txt)

— MultiMarkdown (.mmd)

All media files will be exported as they are (and text file setting is irrelevant to them). The structure of folders created on disk will reflect their structure in the binder. There are a few caveats to watch for. Since traditional files and folders do not support text material in a folder, Scrivener will need to create a separate text file to hold whatever material had been typed into the folder in Scrivener, if applicable.

Return to chapter ⤴

25.2 Metadata and Options

The export panel features the following settings in two tabs. The Metadata tab enables additional material to be exported, and will be useful if you are preparing an archival backup that can persist long beyond your use of Scrivener:

Notes Document notes will be exported as separate files using the text file format you chose above (RTF will be used instead, when any script formats are chosen). The naming convention will be "(Binder Title) Notes".

Metadata The creation & modification dates, label, status, keywords, custom metadata and synopsis will be exported into a file with the naming convention of "(Binder Title) MetaData.txt". Plain-text will always be used for this file.

Snapshots Each stored snapshot of the document will be exported into a subfolder with the naming convention of "(Binder Title) Snapshots". The snapshots themselves will be exported using the chosen text file format. They will be named according to their snapshot title and date stamp.

The "Options" tab has a variety of settings, some of which are format dependent:

Do not export subdocuments Only the literal selection in your binder will be exported.

Number exported files The naming convention will be modified to "(Sibling Order) (Binder Title)", modifying any instances above where merely binder title is used. Sibling order is determined by folder. So if one folder has four items within it, those items will be numbered 1 – 4, and with the next folder (itself numbered "2") would start over at 1 for the first child item within it.

Remove comments and annotations All notes-to-self using these two features will be stripped from the output. If you have no intention of archiving your work notes, or are intending to share these exported files with proofer, this can be a useful option.

Use default Final Draft screenplay elements Available when using the "Final Draft (.fdx)" export format.

If you have modified your screenplay script settings to suit personal taste, you might wish to have them exported using standard settings for archival. This feature will have unintended consequences if you try to use it with scripting files that are not by their nature screenplays.

Append ".fountain"/".txt"/".mmd" extension Available when using any of the plain-text formats.

Forces the file extension to be a specific standard extension instead of letting you choose an extension in the **Save As** field above. This is also useful if you'd rather not bother typing in an extension, as it is possible (and valid) for text files to not have an extension.

Convert rich text to MultiMarkdown Available when using the "MultiMarkdown (.mmd)" format.

Uses the same rules described in the Compile settings general option pane, in the option by the same name (subsection 23.4.3). Rich text formatting such as lists, tables links and so forth will be converted to MultiMarkdown syntax as best as possible.

Return to chapter ⤴

25.3 Exporting to an Outliner with OPML

Many outliners support a common format known as OPML. It can describe "headings" which are what you see in the binder as names of items, and the relationship between those headings in terms of order and depth.[1] There is a loose convention followed by some outlining programs to attach plain-text "notes" to outliner headings[2]. As this method is not standard, support for it may vary in terms of quality and features.

Scrivener supports full note attachment to headers for both import and export. This is optional, so if you are having difficulties loading the OPML file in the target software, try using "Titles only" as your export option, which will adhere to the standard OPML specification.

To export an outline:

1. Select the items you wish to export—any selected containers will automatically include all descendent items.

[1] This method is intended for exporting an indented outline, rather than exporting metadata. If you wish to export tabular metadata lists, try Exporting Metadata to a Spreadsheet (section 25.4).

[2] In technical terms, the text will be inserted into the "_notes" attribute of the XML tree.

If you wish to export the entire binder there is no need to select anything; you will be provided with an option for doing so in the export dialogue.

2. Use the File ▸ Export ▸ OPML File... menu command.
3. Select your export options (refer to "Usage" for detailed descriptions of them).
4. Select a folder and filename to export the OPML file to and click the Export button.

When you select a container to be exported, that container will be considered the "root" item. Depending on what outlining software you use, this item may not be initially visible, as some default to hiding the root item. It is however stored safely in the OPML file. When exporting the entire binder, Scrivener will create a new root item to represent what was the binder itself, and place all folders within it, including even the "Trash" folder and its contents.

25.3.1 Usage

The following options are available from within the export panel:

Export entire binder Rather than use the current selection, export the entire Binder outliner from top to bottom. This will include *everything*; even items you have currently in the Trash can.

Extended note options With the exception of the initial option, these settings will attach the designated information to each heading int the outline as a note. If the software you are using to read the OPML file supports notes, you should see this information appear however it does so. If you are having troubles importing the OPML, try leaving this set to "Titles Only".

— *Titles Only*: only the names of binder items will be exported as a hierarchical outline.

— *Title and Synopses*: contents of any index cards, or the synopsis field will be attached to their respective headings in the outline.

— *Title and Text*: in this case, "text" refers to the main text from the editor for that item. This only includes text from file or folder documents in the binder. The output will be converted to plain-text by necessity.

— *Title and Notes*: any document notes will be attached to their respective items in the outline. The output will be converted to plain-text by necessity.

Return to chapter ⤴

25.4 Exporting Metadata to a Spreadsheet

Outliner views can be exported to files suitable for loading in spreadsheet software, like LibreOffice Spreadsheet or Microsoft Excel; many databases and other miscellaneous programs that support tabular data, such as DEVONthink Pro, may also read the file.[3] To export spreadsheet information:

1. Select all of the containers or items you wish to export. For example, you could click on the "Draft" folder if you wished to export the entire work in progress.
2. If necessary, switch your group view mode to Outliner, with the View ▸ Outliner menu command, the group view toolbar button, or ⌘3.

   All of the items listed in the outliner will be exported, including any descendent items of containers (even if the containers are collapsed).
3. Adjust the column order and which columns you would like to have exported. If you'd rather have all possible columns export you can choose that as a setting in the subsequent dialogue.

Formats The three formats below are all commonly accepted by most spreadsheet applications. Check with your preferred software for details. Scrivener will export the first row in the file as "headers", so your software should be instructed as such if applicable.

- — Comma separated values (best for most spreadsheet software).
- — Tab separated values.
- — Semi-colon separated.

Only include columns visible in outliner When checked, the column list and the order of those columns will be used to create the data file. When disable, Scrivener will export *all* metadata columns (including any custom columns you might have added) in the order that they appear in the View ▸ Outliner Options ▸ submenu.

Return to chapter ⤴

[3] This method is primarily intended for tabular data export. If you would prefer a hierarchal export more suitable for outlining, you might want to try Exporting to an Outliner with OPML (section 25.3).

Printing

26

In This Section...

Given that there are two discrete functions in the binder—your draft or works in progress, and all of the support files and notes around it—you may need two fundamentally different print modes. Since proof printing and final printing the draft are closely related to the act of compilation itself, this function has been built into the compile interface (select **Compile for**: "Print"). For all other forms of printing, either piecemeal or in groups, the standard print command familiar to most applications will be available, including a few special-purpose printing tools for visualising clusters of data, like index cards and outliner columns, that specialise in highlighting metadata and synopses.

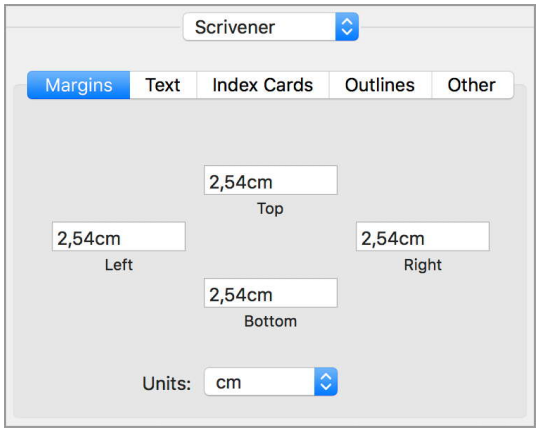


Figure 26.1: All of Scrivener's print settings are found by switching to the "Scrivener" section of the Page Setup panel, using the dropdown at the top of the dialogue.

In most cases (compile, notably excluded) all you need to print a selection is use the standard File ▸ Print... command (⌘P) . Continue reading for tips and features on how to control the appearance of your print-outs. Many aspects of the printout can be adjusted in File ▸ Page Setup... and will be covered in each relevant section. These settings are saved into each project, not globally, but will be saved into the project templates you create.

26.1 Document Printing

Printing is done by using the contents of the active editor split. Individual documents and supported media can be printed one by one. When more than one text document is displayed in the editor they will be printed together according to the current view mode. If you are viewing the selection as a corkboard, index card printing will be used, if viewing them in Scrivenings, the text view will be printed using the text document printing settings (below).

If you select a single media file, such as an image or a PDF, it will printed as per normal (as if you had loaded that file in a program for viewing that type of file and print it from there. However if you select more than one media file, or a mix of media and text files, then one of the group printing methods (corkboard or outliner) will be used instead.

When printing text or outliner content, margins can be set up with File ▸ Page Setup... (⇧⌘P) command, and switching the "Scrivener" section of the pane, which should resemble Figure 26.1. All of the settings referred to in this chapter will be found in one of the tabs found within this section of the Page Setup pane.

26.1.1 Text Document Print Settings

Switch to the "Text" tab to access settings for printing text documents, either when printed individually or as Scrivenings:

> **Printing with Placeholders and Number Tokens**
>
> The basic one-off print tool will not evaluate all placeholder tags or the more complicated uses of auto-number counters in your documents. If you wish to print with these evaluated, you should select the documents you wish to print, and then use File ▸ Compile... with either the Contents tab's compile group set to "Current Selection", or the use of "Current Selection" as a filter (subsection 23.4.1).

Header

Page numbers Insert a page number in the top-right corner of each page in the header area. This number will be relative to the selection you have chosen to print, it will not reflect the actual page number of the complete draft (if you are printing from within the draft).

File name and date Inserts the selected binder item's name next to the page number in the header area. If more than one document has been selected the project's file name will be printed instead.

Content

This section is similar to the "layout content columns" in the upper half of the Section Layouts compile format pane (section 24.2.1). You can optionally add

metadata to the printout for each document that is selected. By default, only Text will be selected, which is the main text body of each document.

Title formatting will be determined by your current Scrivenings title font settings (even if you do not display titles in Scrivenings), in the Appearance: Scrivenings preference pane (section B.5.13).

If you require more control over the formatting, presentation or ordering of these various elements you should probably create your own compile format and use the dedicated compile feature instead of this tool, which is mainly built for expedience and simplicity.

Options

Remove annotations By default inline annotations will be removed. Tick this option to include them, using their original colour, and enclosed in square brackets to help identify them when using a black and white printer.

Insert linked comments By default, linked comments will be inserted beside the highlighted anchor range in a darker variation of the highlight's colour. Uncheck this to keep them out of the print job.

Print using font If you wish to override the formatting of the documents temporarily, and use a uniform font, check this box and then select the font and font size in the activated option menus, below.

Return to chapter ⤴

26.2 Printing Index Cards

When viewing a collection of items on a corkboard, you can print the content of the corkboard onto index card sized blocks on a page. The formatting for this has been optimised to work with Avery(tm) Perforated Index Card stock, but you can use any paper with a chopping block or scissors to cut the cards apart if need be, or even feed in regular index cards if your printer supports abnormal paper feeds.

Since this method is optimised for printing to standard index card size, it will not print a perfect copy of what you see on the corkboard, and in most cases this would not be desirable anyway as the background textures and such would be a waste of ink, and longer synopses would only show what could be seen without scrolling. So instead, the content of the card's title and synopsis will be printed, and will continue to print on subsequent cards until the entire synopsis has been printed. Cards with lots of text content may take several cards to print out.

If the corkboard contains images, the image thumbnail will be placed into the card area unless that item has specifically been set to display the text synopsis instead of the image thumbnail. Note that most metadata will not be shown when an image is placed into an index card as a thumbnail.

26.2.1 Index Card Print Settings

As with printing text documents, you can access options for corkboard printing with the File ▸ Page Setup... command, navigate to "Scrivener" settings from the dropdown, and then click the "Index Cards" tab.

Content

Include titles By default the title of each binder item will be printed at the top of the card. Untick this option to focus more on the synopsis content. The following two options will have no effect when this is disabled.

Embolden titles The title of the card, printed at the top, will be emboldened to set it apart from the rest of the card content.

Highlight titles with label color The background of the title area will be highlighted with the card's assigned label colour, if relevant.

Add card numbers This is similar to View ▸ Corkboard Options ▸ Show Card Numbers feature. Each card will be numbered relative to the visible cards, starting at 1.

Include keywords When enabled, all keyword names will be printed out in a comma-delineated list below the title.

Options

Ignore cards with titles only With this option enabled, if a card has no synopsis, it will be ignored. This includes images that would otherwise be printed out as image thumbnails.

Print cutting guides This option is most useful when using standard paper. Each card will be outlined with dashed cutting guides, making it easier to separate them into actual cards with a cutting block or scissors. If you are using perforated card stock, it is best to leave this option off.

Force landscape orientation Maximises the number of cards you can fit onto a single sheet of paper to four instead of three. If you are just printing to regular paper and plan on cutting them apart, use this option to save paper.

Print using font Override the default font with your preferred font family and size.

26.2.2 Tips for Printing to Individual Cards

With some printers, it is possible to feed individual cards into the printer, which makes for a cheap alternative to perforated cards. You will want to ensure that your printer is capable of handling thicker paper in small sizes before attempting this. Follow these steps to set up Scrivener:

1. Select the File ▸ Page Setup... menu item.
2. Under "Paper Size", select the menu choice, "Manage Custom Sizes...".
3. Press the + button to create a new custom paper size and call it "Index card", or whatever you prefer.
4. Enter the height and width of your index cards. Use a ruler if you are unsure. Most cards come in 3 x 5 inch and 4 x 6 inch form.
5. For the non-printable area, you can select whatever you like here. A small value of 0.25 inches is a good default.
6. Click OK to confirm the new paper size and then make sure that is the selected paper size before clicking OK again to confirm your page setup.

You will now need to follow the instructions provided to you by your printer manufacturer to figure out how to feed the cards into the machine. Some printers will let you place the cards in a stack, but with many printers you will need to feed in a card one by one as it prints. Keep this in mind if you intend to print out hundreds of cards!

Return to chapter ⤴

26.3 Printing Outlines

As with corkboards, you can print a limited range of content from an outliner view by simply viewing the material you wish to print in the outliner, and using the File ▸ Print... command. Also, as with corkboard printing, this will not attempt to reproduce the precise appearance of an outliner. Instead, the outliner printing tool generates an indented list with the title, synopsis and select metadata for each item in the current outline view.

If you find the following settings for the outliner fails to provide the look you desire, you might consider using the compiler instead. Try starting with one of the provided built-in formats "Enumerated Outline" or "Full Indented Outline", and customising them to taste. Or, if you just want the data, try exporting the outliner to a spreadsheet compatible file (section 25.4).

File name The name of the container that is having its contents printed will be placed in the header of the page. If the editor contents are from a selection rather than a particular folder or other group, then the file name of the project will be printed.

Titles Each item's binder title will be included in bold text. Tick the **Prefix titles with number** checkbox to include a simple number count (non-hierarchical) for each item in the outline.

Synopses The full synopsis will be printed for each item.

Label and status If the label and status have been set they will be added to the title. Label will highlight the background of the title area with its associated colour, and will be inserted after the title (if present) in parentheses. Status will be placed at the bottom of the entry in the metadata area.

Keywords Keywords will be added on the line above the synopsis, in the content area, in a comma-delineated list.

Custom metadata Any custom metadata assigned to the item will be added in the metadata area, one field per line for each metadata field that has been filed out for that row. If a row does not have any custom-meta data assigned to it, then nothing will be printed here.

Word counts The word count for each item individually will be placed into the metadata area. Optionally ticking the **Include targets with counts** checkbox will print the goal for each item individually following its word count.

Character counts As above, only printing the character count (and optionally goals as well) instead.

Indent by level Indents each item relative to the current outliner view. So items which are children of the root level items will be indented once, no matter how deeply nested they are within the binder. Disabling this produces a flat list.

Font Select which font and text size you would like used for the printout.

Return to chapter ⤴

26.4 Printing the Draft

Printing the entire draft, the portion of your project that is set aside for your work in progress itself, is accomplished via the compiler, and is not unlike compiling to a saved file. You will use the same interface to print, and most of the options available to you when compiling a PDF will be functional.[1] To print, open the compile interface with File ▸ Compile... (⌥⌘E) and using the **Compile For** dropdown menu at the very top, select "Print", then click the Compile button.

Once Scrivener has finished processing your draft, you will be presented with the standard macOS print dialogue. From here, you can choose to save a PDF from the dropdown menu on the left, or click the `Print` button to send the compiled document to your default printer. The first time you do this, you should Preview the print job, first. If you do not see a `Preview` button, make sure the

[1] The few options that are not available pertain to such things that are not relevant to a printout, such as hyperlink appearance and behaviour, and the PDF's built–in table of contents.

print dialogue is in its collapsed state by clicking the little upward pointing arrow beside the printer selection menu. This will temporarily load the print job into Preview, and if everything looks okay, you can click the print button in the footer bar of the preview area.

See Also...

— Read Compiling the Draft (chapter 23) for more details on how to compile.

— If you want to print a script, read Printing or Exporting a Script (section 19.4) for tips.

Return to chapter ⤴

26.5 Other Print Settings

Since scrivenings mode is not available to most media, your only options for printing groups of them will be as an outline, where they will act just like text items by printing their titles, text synopses if available and so forth. The one exception is images on the corkboard, which will print within the index card space where the synopsis would ordinarily be printed.

When printing items individually, research files will print out in a standard fashion according to their type, and there are no options for adjusting how that occurs. Images and web pages have a few options available in the same Page Setup panel, under the "Other" tab.

Web Pages By default, background images and colours will not be sent to the printer to save ink and increase clarity. However if the background image is an integral part of the design, tick the **Print backgrounds** checkbox to have them included.

Images There are two scaling options to choose from: **Print actual size** and **Scale to fit page**. If the image you are printing out is very large, you will have better success with the latter option. In most cases actual size will produce a better quality print as small images may become quite blurry when blown up the size of a page.

Return to chapter ⤴

Part V

Appendices

> The false starts and futilities of the past years proved themselves to be groundwork, foundations, laid in the dark but well laid.
>
> **Ursuala K. Le Guin**

Menus & Keyboard Shortcuts

In This Section...

This appendix will address each entry in every menu, and will display the keyboard shortcut for it where available. It is intended to be a quick, exhaustive reference of the menus, and will attempt to point you in the right direction for more thorough discussion and analysis of various features where necessary.

The menus in Scrivener are divided into several domains, into each we attempt to organise groups of functions respectively. If you are looking for a particular feature and are not sure where it is located, the table describing the menu sections (Table A.1) will explain our understanding of the menu as a category for features. If you happen to know the name of the feature you are looking for, and have access to the software, the easiest way to find a menu command is to use the search bar from the Help menu.

If a menu name changes its label depending on how you've used it, this will often be indicated with the "|" symbol, separating the parts that change. For example: View ▸ Show|Hide Inspector, will either print "Show Inspector" or "Hide Inspector".

A.1 Custom Keyboard Shortcuts

There may be certain menu items that you find yourself using a lot that have no keyboard shortcut, or maybe you find the assigned shortcut overly convoluted. Scrivener includes many menu commands which are meant to be used with custom shortcuts. The ability to change or assign keyboard shortcuts to menu items is built right into macOS itself:

Table A.1: Menu Organisation

Menu Title	Purpose of Menu
Scrivener	⟨macOS only⟩ The main application menu is where overall application settings and information can be accessed. Changing preferences, setting up registration information, and quitting the software can be done from here.
File	Concerning the management of projects, including the transfer of information into and out of them, either manually with import, print & export, or automatically with backups and synchronisation.
Edit	If a function will work with the *content* of an editor window, either passively or to manipulate it directly, then chances are it will be located in the Edit menu. Examples would be finding text, adding a link or checking spelling.
Insert	Commands for inserting things into your text, such as images, equations, placeholder tags, footnotes and comments.
View	For adjusting how the project window looks and feels, as well as the appearance of content within windows, such as changing the zoom level of a PDF, showing invisible markings in the text or selecting which columns are used in the outliner.
Navigate	All about getting around inside the project window. Loading documents, opening them into splits, jumping between the window's various contexts and viewing collections are within the remit of this menu.
Project	Settings and information, statistics and goals specific to the project itself are gathered in this menu. This is also where new content can be created, and where the Trash can be emptied.
Documents	This menu concerns the management of existing documents, including the content within them, the metadata used to describe them and their organisation within the project. Documents can be moved to other folders, grouped into new ones, or even trashed from this menu.
Format	This menu concerns itself with the *presentation* of content. If you want to change the font or paragraph indents, this menu is what you'll use. Scriptwriting, styles and revisions are also managed from this menu.
Window	The various windows themselves can be managed here, such as basic commands for minimising or switching between open projects, selecting from predefined layouts to adjust how a project window looks and feels or to reopen recently closed Quick Reference panels.
Help	The Help menu provides standard access to your Mac's menu search utility, as well as useful tools and links for learning Scrivener, checking for updates, registering the software or getting in touch with us.

1. Go to your System Preferences (available from the Apple menu in any application).
2. Load the "Keyboard" panel and select the "Shortcuts" tab.
3. Select "App Shortcuts" in the list on the left.
4. Click on the "+" button. A sheet will appear.
5. From the "Application" pop-up button, choose Scrivener.
6. In the "Menu Title" text field, enter the exact name of the command you want to add. This should exactly match the name of the menu item in Scrivener (capitalisation and punctuation matters). For instance, if you wanted to add a keyboard shortcut to the Insert ▸ Image Linked to File... command, you would type `Image Linked to File...` in this text field (including the ellipses). If you wanted to change the keyboard shortcut for the Edit ▸ Paste and Match Style menu item, you would type `Paste and Match Style`.
7. Click in the "Keyboard Shortcut" text field and then hold down the combination of keys that you want for the new shortcut.
8. Click on "Add".

When you return to Scrivener, the new keyboard shortcut should be ready to go.

Resolving Conflicts

Keyboard shortcuts on macOS work by scanning the menus from left-to-right looking for a menu item matching the shortcut pressed. If you find that the shortcut you assigned doesn't work, or does something unexpected, it may be that the keyboard shortcut you chose is already assigned to a different menu item. In that case, you can either pick a different shortcut, or you can locate the menu item that it clashes with and go through the above process again to assign a different shortcut to the clashing menu item. If the shortcut still doesn't work, you should ensure that the shortcut you assigned isn't one reserved by the system.

Menu titles that change dynamically

Some menu items change name depending on the context; for such items, you may need to assign the same keyboard shortcut for each of their possible names. For instance, the Edit ▸ Add Link... menu item can sometimes change its title to become Edit ▸ Edit Link... Therefore, to add a keyboard shortcut to that item that would work consistently, you would need to add the same shortcut twice, once for `Add Link...` and again for `Edit Link...`.

Menu titles that are duplicated

Sometimes a menu title will be used more than once. This most often happens with titles that are created dynamically from your project information. A good example of this is Navigate ▸ Collections ▸ Name Of Your Collection and Documents ▸ Add to Collection ▸ Name Of Your Collection. If you created a shortcut called `Name Of Your Collection` alone, this would be bound to the initial shortcut that shows the tab in your binder, not the command that files the current document into that collection. To target a specific menu, you also need to type in its menu hierarchy by inserting -> between each menu level, thus:

```
Documents->Add to Collection->Name of Your Collection
```

A.2 Scrivener Menu

Much like the application menu in any other macOS program, this provides access to application level information, features, and system integration, such as Services.

About Scrivener Displays the credits and version number of the application. If you are experiencing problems and wish to contact customer support regarding them, you can provide version information using this dialogue. Click on the version number to copy this information to the clipboard and dismiss the window.

Preferences... ⌘, Accesses the main application preferences window. For a complete list of all available options, see Preferences (Appendix B).

A.2.1 Themes ▸

This submenu will list any preset themes that you have created or installed into Scrivener. To switch appearance settings, select the desired theme from this list. Refer to Preference Presets and Themes (subsection B.1.1) for further information on managing themes.

Reveal Support Folder in Finder Opens Scrivener's "Application Support" folder, where it stores your presets, custom project templates, custom icons, and so on. Use this if you wish to transfer settings between machines, or if asked to do so for troubleshooting reasons.

test

Authorize Folder Access... <$mas> Opens a window where you can manage a list of folders to grant Scrivener additional privileges in accessing your disk. If you make use of file links, research aliases or image links, you might wish to authorise the folders they are in, or even your entire user folder so that Scrivener can make use of these files without explicit permission per session.

Registration... <$direct-sale> When you purchase the application, use this menu item to copy and paste your registration information from the email that you will receive from Literature & Latte. In case you have lost the original email or never received the invoice, you can use the Retrieve Lost Serial... button which will take you to a web page with further instructions on how to retrieve it.

Check for Updates... <$direct-sale> You can use the Check for Updates menu item to see if there is a newer version of Scrivener available for download (3.x updates are free). If any are found, you can update the software conveniently from this tool. You can also choose to have updates applied automatically in the future, and this check can be performed routinely as well, in the Startup tab of the general preference pane (subsection B.2.1).

Services The items in this menu are provided by the core system and other third-party applications. They will let you perform various functions, mostly based on selected text. Scrivener provides its own services which are also available in this menu, and from other applications as well. See Scrivener Services (section 9.3) for further documentation.

Quit Scrivener ⌘Q Leaves the program. Any projects that are left open will be saved to the disk, and by default, backed up for you. Under the default preferences, these projects will be remembered and opened automatically the next time you run Scrivener.

Quit and Close/Keep Windows ⌥⌘Q Whether this will "Close" or "Keep" open projects upon next reload depends upon whether your preferences are set to reopen projects that were open on quit (subsection B.2.1). Which is displayed will be the opposite of whatever your standard settings are. This will not change your preference, it only influences how the software works this one time.

A.3 File Menu

The File menu contains everything that handles creating files on your computer, including creating new projects, saving, backing up, importing and exporting. It also deals with printing and project-specific settings.

New Project... ⇧⌘N Brings up the Project Templates window which will walk you through creating a new project in the location you specify.

Open... ⌘O Brings up a file selection dialogue for opening an existing .scriv project on your hard disk. Full read-write permissions must be set for a project to be opened successfully. Older projects may need to be updated; you'll be walked through that process if necessary.

Recent Projects Here you can review a list of projects that have recently been used recently and select one to open it (you can also specify in the General Preferences (subsection B.2.1) whether Scrivener will reopen all projects that were open in a previous session when it is launched).

The number of items listed in this submenu is governed by your macOS' global preference, which can be set in the "General" system preference pane.

Favorite Projects Displays a listing of all the projects on this machine that have been flagged as a "favourite". Use this to load or activate the window of the project you select. Read more about it in Setting Favourite Projects (subsection 5.1.4).

Add|Remove Project from Favorites Adds or removes the currently active project to the aforementioned "Favorite Projects" list.

Show Project in Finder The active project will be located and presented to you in a new Finder window. This information can also be acquired from the project window title bar, by right-clicking on the name of the project itself.

Find All Projects in Spotlight <$direct-sale> Runs a Spotlight search, bringing up a Finder window with the search results of all Scrivener projects included within the remit of the Spotlight index. It is important to distinguish the difference between this and searching your entire computer for projects. The Spotlight index may not include drives you have specifically omitted, or external drives you have plugged into the computer. It will also not locate zipped projects (usually a result of automatic backups).

Close Project ⇧⌘W Closes the current project and all of its associated windows. The project will be automatically saved upon close, and under default settings will be backed up.

Close Project and Clear Interface Settings ⇧⌥⌘W When holding down the Option key and viewing the File menu, you'll see this command for closing a project. This will safely shut down the project and as a final step, trash the project's "UI" preference file. All options such as visible elements (rulers, inspector, etc), splits, label tinting, columns, and other settings will be factory reset. Ordinarily, you'll only ever need to do this as a troubleshooting step, or a way of fixing a layout glitch you've encountered.

Close Window ⌘W Closes the currently active window (e.g. Quick Reference, or utility panel). However if the project window has focus or is the only open window for the project, the entire project will be closed instead. Composition mode must be exited explicitly, it does not respond to this command.

Save ⌘S Scrivener auto-saves your writings as you work, so that you never have to worry about losing your efforts. Projects are also saved automatically whenever they are closed. However, you can use the save command to force an immediate save whenever you want.

Save and Rebuild Search Indexes ⌘S When holding down the Option key you can manually rebuild the search index while saving. This is sometimes useful in cases where you suspect the project has lost synchronisation with the search index. There is typically no need to do this as Scrivener will automatically maintain indexes, and it will rebuild its index if something appears to have gone awry since the last session.

Save As... ⇧⌘S Will prompt you for a new project name and/or location. When you submit this dialogue box, Scrivener will immediately start working with the *new* copy, closing the version you had been working on up until that point. If instead you want to generate a backup copy and keep working with the original project, use File ▸ Back Up ▸ Back Up To... instead.

For more information on managing projects, refer to All About Projects (chapter 5).

Import ▸ Commands for importing a variety of existing information, from your drive or even from the Internet, into your active project. For full documentation on importing information into Scrivener, refer to File Import (section 9.1).

Import ▸ Files... ⇧⌘I Import files from the file system into Scrivener. Only text-based formats can be imported in the Draft folder. See the prior referenced section for a full list of supported formats there, and elsewhere in the Binder. This method is synonymous with dragging and dropping files directly into the binder itself.

Import ▸ Web Page... Import a web page from the Internet by supplying Scrivener with its URL. By default it will be archived to an offline format, retaining its look and feel using the WebArchive format (same as when saving a page from Safari). You can optionally choose to import only the raw text of the page (subsection B.7.1).

This method of importing is also available from the Add button in the main application toolbar.

Import ▸ Research Files as Aliases... Rather than fully importing items into the project, this command establishes a link between the original item and the project. This link will automatically adjust if the original item is moved or renamed. Read about Linking to Research Material (section 9.2) for further details.

Import ▸ Plain Text Formatted Screenplay... Import a plain text screenplay, optionally splitting sections into separate binder documents. This works

best with scripts that have been exported from programs like Movie Magic Screenwriter that work with plain-text files for import and export. This does *not* support the Fountain format. Fountain files can either be imported normally with the File ▸ Import ▸ Files... command, or with the File ▸ Import ▸ Import and Split... command, to break the imported file up by slugline.

Import ▸ Scrivener Project... Import a selected Scrivener project into the current project's binder. All binder items will be imported, as well as some metadata: keywords, references, and notes. If it is detected that the imported project appears to be an older version of the same project you are importing into, you will be offered a chance to merge the projects together. Refer to Merging Projects (subsection 5.3.2) for further information.

Import ▸ Import and Split... This is a multi-use tool which can either takes a standard plain or rich text file and allows you to supply a character sequence (such as "#") which will be used to split the document into sections, or for some formats, intelligently break the file up by logical sections based upon the internal format of that file. For example a Word document with a stylesheet based outline can be converted into a nested binder outline, a Markdown file can be split into an outline via its heading structure, while Final Draft and Fountain files will be split by sluglines.

. . . . . .

Export ▸ Provides tools for exporting elements of the binder to the file system. For more information on exporting your work, see Exporting (chapter 25).

Export ▸ Files... ⇧⌘E For exporting selected contents of a project as individual files and folders. This feature is mainly used to create a hard copy backup or to share resources with someone not using Scrivener. Generally speaking, you will want to use File ▸ Compile... to export your book as a single file, rather than many. Refer to Exporting Binder Files (section 25.1) for more information.

Export ▸ OPML File... Exports the current binder selection as an Outline Processor Markup Language file. OPML files are commonly read by applications that work in outline oriented data, like Scrivener, and is thus a convenient way of transferring hierarchal information (or the file tree) between such programs. See also: Exporting to an Outliner with OPML (section 25.3).

Export ▸ Outliner Contents as CSV... When an Outliner view is selected, this menu command will activate. It will export the current outliner in a format that is readily accessible to spreadsheet software and other programs capable of tabular display. See also: Exporting Metadata to a Spreadsheet (section 25.4).

Export ▸ Comments and Annotations... Exports only the comments and annotations from the project into a single file. You can optionally choose to export only pre-selected binder items and select whether the list is organised by binder titles.

Export ▸ as Scrivener 2 Project... Save a copy of your project in the legacy Scrivener 2.x format, capable of being freely worked upon with this older version of Scrivener, as well as version 1.x for Windows. Refer to Saving Your Projects for Older Versions (subsection 3.4.4) for usage tips.

Sync ▸ Tools for linking parts of your project to external applications, mobile devices or disk-based files and folders. Read more about syncing in Cloud Integration and Sharing (chapter 14), and Working with Scrivener for iOS (section 14.2)

Sync ▸ with Mobile Devices Forces Scrivener to check the current project for mobile changes. In ordinary usage this will not be necessary as these checks are performed automatically when switching to the project from another window and of course when loading the project. If the project window has been open and active while using iOS to edit it remotely, then you will want to use this command to bring the project on your computer back in sync.

Sync ▸ with External Folder... Method for creating folders with plain-text, rich text, Fountain or Final Draft files on your system. The export location can be anywhere, including network mounted drives or synchronised folders such as provided by Dropbox, Google Drive, iCloud Drive and SpiderOak. It is thus useful for coordinating with collaborators or even just working on pieces of your project while on the go. See also: Synchronised Folders (section 14.3).

Sync ▸ with External Folder Now Runs the external folder sync tool immediately without loading the configuration window. If External Folder sync has not been previously set up, it will load the configuration window for you.

Back Up ▸ Functions for managing backup copies of the current project. For more information on back up strategies, see Backing Up Your Work (section 5.2).

Back Up ▸ Back Up To... Generates a complete backup copy of the project to a specified location. Backup copies, unlike "Save As", will be created and then ignored by Scrivener, you will keep working in the current version

of the project. Optionally, you can choose to compress the backup in a zip archive, which takes longer, but is the recommended method for storage of backups, particularly if you intend to copy the backups over the Internet in one form or another.

Back Up ▸ Back Up Now Triggers the automatic backup system to produce a backup immediately, using the established preferences for backup location and rotation scheme. Customise how this works in the Backup preferences tab (section B.8).

. . . . . .

Save As Template... Saves the current project as a template. This will add it to the New Project window for use as a basis for your future projects. Every single aspect of the project will be saved, including any content you leave in the binder. For more information on creating templates, read Project Templates (section 5.4).

Page Setup... ⇧⌘P Accesses the standard page layout setup sheet. Additionally accesses Scrivener specific features from the Settings dropdown menu within this sheet. Most compile formats will use these settings to determine paper size and margins, as well as for current document printing (below). Compile formats can be further configured to use their own settings, including full two-page layouts with offset margins. For full documentation on the standard printing process, read Printing (chapter 26).

Print Current Document... ⌘P Prints the current editor view. How this will be printed depends on the view mode, with all of settings for this accessed through the File ▸ Page Setup... command, above.

Compile... ⌥⌘E Compile is the standard method for producing a manuscript out of all the individual pieces in the Draft. This feature provides an immense degree of flexibility, and is fully documented in Compiling the Draft (chapter 23).

A.4 Edit Menu

The Edit menu contains options related to editing content. All of the standard Mac Edit menu items (with which you will be familiar from such programs as TextEdit) can be found here, including cut, copy, paste and find, alongside with a large complement of Scrivener-specific features.

Undo & Redo ⌘Z **&** ⇧⌘Z Undoes or redoes the last change. Undo and Redo work mainly for edits made to text, but they do work for some basic outlining changes too, where that change does not cause the current document to change in the editor. Each document has its own Undo history,

which means you can go back to documents you've edited previously and undo changes made there without undoing changes made more recently to other documents.

Cut, Copy, and Paste ⌘X, ⌘C & ⌘V Cut, copy and paste act exactly as they do in other applications. These commands will not act upon documents that you have selected. They will work primarily with text, but can sometimes also be used to copy and paste metadata like bookmarks or keywords. This manual will make note of where copy and paste can be used for non-textual edits. For editing documents, refer to the Documents Menu (section A.9).

Copy Special ▸ This menu allows you to copy text or binder items using functions to create specially formatted lists of items, remove markings from the text or convert the text to different formats.

Copy Special ▸ Copy Document as External Link Copies a URL to the clipboard which can be used to link to the selected item from any other software supporting links on your computer. You could for example paste this URL into the hyperlink URL field of your word processor, creating a link from that document to a particular item in your project binder. These links are static references to the project's location on your disk. They will not work from other computers, or if the project has been renamed or moved from the location it was in when the link was generated. Read more about External Links (subsection 10.1.6). This command is also available from the binder contextual menu as "Copy Document Link".

Copy Special ▸ Copy Text of Documents The text content of the selected documents will be copied together into one continuous text in the clipboard using the order they appear in at the time of copying. If copied from the binder, this would be the binder order, but if copied from a sorted outliner view then they could be copied, for example, in chronological order.

Copy Special ▸ Copy Documents as ToC To aid in the creation a basic formatted table of contents with page number references, using the selected documents. This list can be pasted into a file within the draft. See Creating a Table of Contents (chapter 22) for further information.

Copy Special ▸ Copy Documents as Structured Link List Generates a list of links pointing back to the selected documents, formatted into an indented list. This form of list can be used to create a basic table of contents in a digital book, where hyperlinks are used instead of page numbers, such as ePub and Kindle.

Copy Special ▸ Copy without Comments and Footnotes ⇧⌥⌘C Strips out inline and linked notation, while retaining all other formatting. Most often this is useful for producing "clean" snapshots (section 15.8) after an editing session. This command will remove all notation, including footnotes.

Copy Special ▸ Copy as Markdown The selected text will be converted from rich text to Markdown format (using the MultiMarkdown dialect). This command produces markup in a fashion described in Markup Options (section 23.4.3).

Copy Special ▸ Copy as HTML Reproduces most formatting as HTML codes using inline CSS. The resulting HTML should look very similar to the text you copied in Scrivener, including some ruler settings, colour, highlight, and so on.

Copy Special ▸ Copy as HTML (Basic, using <p> and) Applies minimal inline CSS formatting. Ruler styles and some types of formatting will be lost. This is often the best choice for pasting into blogging and Content Management Systems, which provide their own stylesheets.

**Copy Special ▸ Copy as HTML (Basic, using
)** Applies very basic, HTML 4.01 compliant code. When constructing HTML emails, or working with an older web site, this is what you will want to use.

. . . . . .

Paste and Match Style ⇧⌥⌘V This pastes the contents of the clipboard without any of its existing fonts, styles and other attributes such as links, tables and lists; in essence, treating it like plain-text. Useful for when you have copied a range of formatted text but want to paste it using the style of the text into which you are placing it, such as when gathering material from the Web.

Paste Plain Text as Screenplay When the active editor is showing a scriptwriting document, this menu command will become available. It can be used to paste plain text formatted screenplays, from the likes of Movie Magic Screenwriter, or even plain text screenplays you compiled from Scrivener in the past. The pasted text will be analysed and converted to the script format. It is only intended to work properly with standard Screenplay based formats.

Delete In some contexts this command can be used to remove items from lists, such as individual keywords in the inspector pane, and is usually passively bound to the Delete key. In similar cases, such as the Bookmarks pane in the inspector, where the removal cannot be undone, it is necessary to use the ⌘Delete shortcut instead.

This menu command can also be used to permanently delete selected binder items from within the Trash folder, without emptying it completely (use Project ▸ Empty Trash... for that). This capability is also available to the right-click contextual menu on items in the trash.

Refer to Deletion (subsection 15.2.2) for information on deleting text in the editor.

Select All ⌘A Selects all of the content of the current text, outliner, corkboard view and etc. This command works in every context where it is possible to select multiple items, from footnotes in the inspector sidebar to keywords to search results in the sidebar.

Select ▸ Contains several commands to aid in the selection of items and text. In views where things are listed, such as the binder or a list of keywords, and where it is possible to select multiple items, there are a few modifier keys that can be added to mouse clicks to form basic selections:

- Command: toggles whether or not one item beneath the mouse pointer is selected. This can be used to effectively deselect everything, if the one and only selection is Command-clicked. It can also be used to add a second item to the selection, no matter how far away it may be from the originally selected item.
- Shift: Selects all of the items in between the point where you click, and the *last* item you clicked on as a selection action. This can include Command clicks from above, as well as single clicks to select one item.

Select ▸ Select Word The next three commands will select the nearest word, sentence or paragraph that the cursor or current selection is found within, or when a selection exists, it will be expanded in both directions until reaching the requested unit of measurement. For example, if the a few letters of two words are selected, the Select Word command would expand the selection left to encompass the first word entirely, and rightward to encompass the second word entirely (Figure A.1).

Given the wide range of writing styles and methods, right, wrong or just creative—grammatical constructs such as sentences can only be estimated.

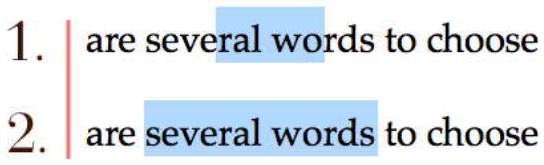


Figure A.1: Before and after usage of the Edit ▸ Select ▸ Select Word command.

Select ▸ Select Sentence As noted above, if you start with two partially selected sentences then you would end up with two fully selected sentences when using the Select Sentence command.

Select ▸ Select Sentence with Spaces Similar to the above, this alternate behaviour which will select one space from around the sentence (whichever

is most logical), so that it can be more easily cut or moved with drag and drop to a new position without clean-up.

Select ▸ Select Paragraph Selects the current paragraph, including the carriage return on the end of it, unless of course the paragraph is at the end of the file.

Select ▸ Select Style Range The current selection will be expanded to encompass the style range around it in a contiguous fashion. This works locally first, looking for character styles, and then expands to any paragraph styles that exist around that selection. If no styled text exists around the cursor or selection, then expansion will encompass the contiguous un-styled text around it.

For more documentation on this and the following two commands, refer to Selecting and Searching for Styles (section 15.6.4).

Select ▸ Select All Style Selects all non-contiguous text within the same text view using the current style beneath the cursor, or as found beneath the leftmost edge of the selected range (the selected text itself will be ignored). As with the previous command, the nearest character style range will be preferred over the paragraph character range, in cases where the cursor sits within text that has both character and paragraph styles applied to it. Also as with the prior command, if the cursor or selection encompasses text with no style applied to it, the command will select all other text likewise without styles.

Select ▸ Select Next in Same Style Using the same criteria as Edit ▸ Select ▸ Select All Style, this command will select the next phrase of text found within the current editor using the same style. If the cursor is within block quote, then the next block quote will be selected. This form of selection will wrap around from the bottom of the document back to the top if necessary, which means if only one example of the style exists, the very context the cursor currently sits in may be the "next" available example of text using this style.

Select ▸ Select Similar Formatting Using the same logic as Edit ▸ Select ▸ Select All Style, this command checks for the type of formatting located beneath the cursor and then scans the document for similar examples, selecting all such non-contiguous text simultaneously. This command is handy for turning simple formatted text into styled text. Given the wide variety of possibilities that fall under "formatted text", we cannot expect this command to function flawlessly in all cases.

If the cursor or selection falls upon text that is styled, then the outcome of this command is analogous to Edit ▸ Select ▸ Select All Style.

Select ▸ Select Annotation/Footnote This command will alternate depending upon whe-ther the cursor or leftmost selection edge is located within an

inline annotation (of the same colour) or inline footnote. If so, the inline note will be fully selected. The command will be disabled if no such inline notation range can be found.

Select ▸ Select Subgroups Applicable only to the outliner and the binder when a container of any type is selected, this command will reselect any descendant container items that are currently visible *and* that contain subdocuments, ignoring any items that are not containers or that are empty. If a folder has three other folders beneath it that are collapsed and one text document, this command would select the three collapsed folders. If one of those folders was disclosed and it contained file group within it, then the three folders and that 3rd level file group would be selected. This command can come in handy when assembling corkboard stacks (subsection 8.2.8). In the provided example (Figure A.2), we can disregard anything that might be within the "Section A" folder since it is collapsed, "Section B" contains one container ("file two") which becomes selected as well, and finally "Section C" is disregarded because it is an empty folder.

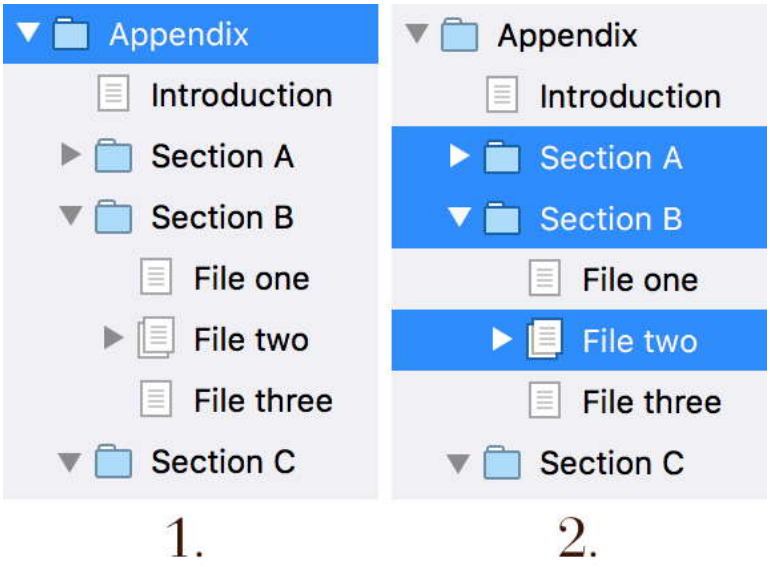


Figure A.2: Before and after results of selecting the subgroups of the "Appendix" folder.

Select ▸ Select Current Text ⌥⌘A When the current context is a text editor instead of the binder or outliner, this command will appear in the menu. It can be used to select only the text in the section you are currently editing within a Scrivenings session, rather than the entire session.

Select ▸ Select with Subdocuments ⌥⌘A When a container has been selected in either the binder or the outliner, use of this command will appear in the menu. Use it to select all descendent items of that container, opening as

necessary all levels of hierarchy from that point downward in order to do so. This command is useful in conjunction with other commands that work on specific item selections, such as taking snapshots or setting metadata.

Select ▸ Select 'Included' Subdocuments All subdocuments of the currently selected container (to all depths) will be selected, only if they have their respective "Include in Compile" checkbox enabled. Easily filter out all notes, old revisions and other nonessential items from your draft folder with this command.

If you wish to filter a collection or hoisted binder view by included items only, you can hold down the Option key and click on the button in the binder header bar for doing so (subsection 10.2.1).

Deselect All If possible, removes all active selections from the current view. In some cases there will always be a default selection that cannot be removed. In a text editor, the cursor position takes on that role, and certain commands and tools will use the cursor position as a form of implied selection. In the corkboard and outliner views the underlying container that is being viewed will occupy the default selection, mainly for deciding what the Inspector will examine.

Completions ▸

Completions ▸ Complete ⌥Esc Manually calls up the word-completion service, regardless of word auto-completion settings. Can optionally be invoked with ⌘. or simply Esc by itself on some keyboard layouts.

Completions ▸ Complete Document Title ^Esc With the first part of a document titled typed in, you can use this command to cause Scrivener to search your project for matching titles and suggest alternatives. This is useful in conjunction with the Corrections preference to automatically detect internal links typed in with `[[Document Title]]` wiki-style bracketing.

Completions ▸ Add Selection to Auto-Complete List Adds the currently selected word to the project's auto-complete list. If more than one word is selected, nothing will be added to the auto-complete list. For more information, read Auto-Completion (section 15.9).

Move ▸ Movement commands are applicable to text as well as items. In some types of views, certain types of movement will be disabled if there is no

logical form of movement in that direction. For example in collection lists, which are flat lists of binder items, left & right movement would have no meaning, so only up and down movement is allowed. In most cases movement will not wrap, so when you reach the end of a view in any particular direction, movement will be disabled.

Movement will in all cases be done one step at a time; an item will not move more than one space away from where it began. When moving selected items in the various group views and lists, non-contiguous selections will maintain their original spatial distance from one another, and the whole selection will move together as a single unit *through* the other items around them. Thus if one selects the very first and very last item in a list they will be unable to move the selected items up or down, since the selection as a whole unit is precisely as tall as the entire list. This means that in many cases non-contiguous selections will have little range of movement, for if any one component of the selection cannot be moved in the requested direction, the entire unit will refuse to move.

If it is your intention to *gather* items together into a container, no matter how far apart they started off, then using the Documents ▸ Move To ▸ submenu or simple drag-and-drop with the mouse will work better. To gather non-contiguous chunks of text together, use Cut and Paste or drag-and-drop.

Move ▸ Move Up ^⌘↑ With **text** the selected lines will be transposed together with the line directly above them. On the **corkboard**, the card will be moved directly up, advancing all cards between where it was and where it will be moved, one position to make room for it. In both the **binder & outliner** views, the selected items will move up within the current list of "siblings", or within a folder and without changing levels. They cannot be move "up" beyond the top of the group they are within. With **label view** the selected cards will advance up one label row, terminating at the top (by default "No Label"). If label view is in a vertical orientation, then moving a card *up* will have a similar effect to moving it *left* in horizontal orientation.

Move ▸ Move Down ^⌘↓ As with moving the selected items *up*, only in this case all movement will be done downward.

Move ▸ Move Left ^⌘← **Text** will have its block indent level shifted outward by 0.25" increments. Any amount of first-line or hanging indent will be respected by this command. The levels of indent for all forms will be incremented equally, and once the left margin is reached by the leftmost portion of text, the command will be disabled. Items in the **binder & outliner** will be move out of their current container to become a sibling to it, inserted between the group they came from and the item that once fell directly below that group. On the **corkboard**, this command will move a card leftward, first along its row, and then wrapping around to the right

edge of the row above, until reaching the position of the first card in the corkboard. With the **label view** the action is identical in that the card will be moved toward the beginning of the current group being viewed, but in vertical orientation left will move the card to the label rail to the left of its current assignment. **Collections** do not allow left or right movement.

Move ▸ Move Right ^⌘→ As with moving the item *left*, only in this case all movement will be done in the opposite direction. In the case of **text**, the indent will be incremented 0.25" per usage and has no effective upper limit. The action of moving an item right in the **binder & outliner** will be to nest the selected items beneath the item directly preceding the topmost selected item.

. . . . . .

Sort ▸ Sorting can be done wherever there is a flat list of adjacent items that can be arranged in a view, when a single item is selected that is a container to other items, or when more than one line of contiguous text has been selected in an editor. Possible applications include items in the binder, outliner or corkboard views, keywords, bookmarks and so forth.

Sort ▸ Sort Ascending (A-Z) This command will sort eligible items in alphanumeric order (0–9a-z) with most punctuation and symbols falling before numbers.

Sort ▸ Sort Descending (Z-A) Sorting by descending order follows all of the same rules, only sorts in (z-a9–0) order, with most punctuation and symbols falling at the end of the list.

. . . . . .

Append Selection to Document ▸ This command presents a binder item selection submenu of everything capable of having text added to it. It requires an active text selection, which will be added to the end of chosen document selected through the submenu. This menu will keep track of your usage of it within the project, tagging frequently used targets in a "Favorites" section at the top of the menu.

Use the "New..." command to create a new document containing the selected text. You will be asked where to place this new document in a subsequent sheet.

Add/Edit Link... Add an external hyperlink of any types to the text. If the cursor is not already placed within a link, you can use this command to insert a new link at the cursor position. When existing text is selected, the hyperlink will be applied to the text rather than inserting the URL directly

into the editor. The URL editing sheet will provide you with several common prefix options[1], or optionally no prefix for custom protocols, like `ftp:` links.

If the cursor is currently placed inside of a link (including Scrivener Links), this command will bring the editing sheet up for that link, allowing you to change the URL, or select a different binder item to connect the link with.

Unlink If the cursor is placed within the text of a link, this command will destroy it. If you select a range of text, all link formatting within that range will be removed. Links only partway selected will remain as trimmed around the bits of the link left unselected.

Link to Document ▸ ⌘L Presents a binder item selection submenu for the creation of a hyperlink to a chosen document within the same project at the current cursor position, using the target's binder title, or if you have text already selected, it will be turned into a hyperlink.

Using this menu, you can also create a link to a document that doesn't exist yet, with the New Link... command (⌘L) In the first tab it will ask you for the title of the new document, and where to place it. After providing this information, a link to this document will be placed in the current editor. The second tab provides alternate keyboard access for linking to existing documents.

Find ▸ As a program designed for working with bulk text, Scrivener has many tools in its chest for searching, collecting items, and organising and reviewing the results. For full documentation on how to use Scrivener's extensive searching facilities, see Searching and Replacing (chapter 11).

Find ▸ Quick Search ^⌥G Provides a shortcut for using the toolbar's quick search field, which otherwise prints the name of the binder item you are currently viewing (Figure A.3). If the toolbar has been hidden, a separate window will be opened for your convenience. Read more in Quick Search Tool (section 11.5).

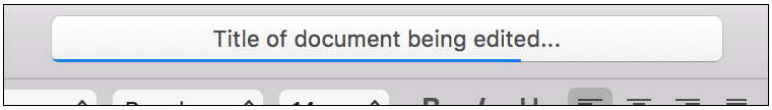


Figure A.3: The "Quick Search" tool displays the document title prior to being clicked on.

[1] For the sake of simplicity, both HTTP and secure HTTPS links are considered one and the same. You can also paste a URL with `http:` or `https:` into the first field without switching to "No Prefix".

Find ▸ Search in Project ⇧⌘F Provides a shortcut for the project search tool, which will appear as a text pane above the binder sidebar. Type text into this field to search the project for matches, in accordance with the current search settings, set in the magnifying glass icon menu to the left of where you type. There is a lot to this tool! Read more about Project Search (section 11.1).

Find ▸ Project Replace... Shows the Project Replace panel, for the bulk replacement of text throughout the entire project or selected portions of it, even including the otherwise immutable snapshots. This operation cannot be undone; use with care. Read more about its usage in Project Replace (section 11.3).

Find ▸ Find... ⌘F Brings up a standard find and replace panel. This panel works within nearly every context where you can edit text, even within the inspector sidebar. In the main editors this often means only one document at a time, but in the case of Scrivenings session it could mean many documents. Read more about Document Find and Replace (section 11.2).

Find ▸ Filter... ⌘F When the active view is a corkboard or outliner, the "Find" command will be replaced with the "Filter" command. This brings up a panel at the top of the view that can be used to filter which index cards or outliner rows you see, based on various criteria. Read more in Filter Outliner & Corkboard Views (section 11.4).

Find Next/Previous ⌘G **&** ⇧⌘G Jumps to the next or last matching text based on the criteria supplied in the Find panel. Note these can be used even if the find panel is closed.

Find ▸ Use Selection for Find ⌘E Sets the selected text as the current find term, copying it into the "Find" text field even if the find panel is closed. It can thus be combined with the previous commands to search for text without using any interface.

Find ▸ Jump to Selection ⌘J Scrolls the editor view so that your cursor position is centred on the page. Useful if you have used the scrollbar, or PageUp/PageDown to briefly check other areas of the text.

When typewriter scrolling (subsection 15.3.4) is enabled this command will use the selected **Typewriter scroll line**, from the Editing preference pane, rather than the centre of the view.

Find ▸ Find by Formatting... ^⌥⌘F Opens the Find by Formatting panel. This tool is quite powerful and has a wide range of options documented in Find by Formatting Tool (section 11.6).

Find Next /Previous Formatting ⇧⌥⌘G **&** ^⌥⌘G As with Find Next and Find Previous, this command will jump from match to match, even without

the "Find by Formatting" panel open. Unlike document find however, this command will seek matches from adjacent documents once the end of the current document is reached. In this way, you could step through the entire project looking for matching formatting.

Spelling and Grammar ▸ Accesses global macOS' spelling and grammar tools. These settings are stored on a per project basis, though their use will establish the settings used to set up your next project, when it is created. Existing projects will always store whatever setting you left them at, regardless of the current default for new projects.

Spelling and Grammar ▸ Show Spelling and Grammar ⌘: Opens the standard Spelling and Grammar panel. If you wish to change the base language the spelling and grammar checker uses, you can set that here.

Spelling and Grammar ▸ Check Document Now ⌘; Start checking for misspellings from the current cursor position on downward. Once the bottom of the document is reached, it will wrap around to the top.

Spelling and Grammar ▸ Check Spelling While Typing ⌘ Toggles the automatic spell
checker that underlines in red unrecognised and misspelled words as you type, unless **Correct spelling errors as you type** has been enabled in the Corrections preference pane (section B.6), under "Auto-correction".

Spelling and Grammar ▸ Check Grammar With Spelling Toggles the grammar checking feature that looks for poor style or grammatical errors as you type and underlines them in green.

Substitutions ▸ These options control whether or not typographic punctuation will replace easier to type basic punctuation as you write. These settings are stored per individual projects, and their defaults are established in the Corrections preference pane, under "Punctuation".

Substitutions ▸ Smart Quotes If necessary, typographic punctuation will be substituted for basic speech mark punctuation as you type. In English for example, the " character would be substituted with a matching pair of " and " characters. The quotation style used are typically set by your macOS localisation settings, or quote settings established in the "Text" tab of the Keyboard System Preferences pane.

For those cases where the text editor produces the wrong type of quote, such as after em-dashes, press the Option key immediately after keying in the punctuation mark. This will cause the editor to use the opposing typographic quote than it otherwise would have.

Substitutions ▸ Smart Dashes and Ellipses With this option enabled a sequence of two or three hyphens in a row will be substituted with an em-dash (—) character, and three full stops will be converted to an ellipses (…) character.

. . . . . .

Transformations ▸ For the transformation of text from one form to another in a permanent fashion.

Transformations ▸ Make Uppercase Converts the selected text to all uppercase characters.

Transformations ▸ Make Lowercase Converts the selected text to all lowercase characters.

Transformations ▸ Make Title Case Converts the selected text to title case, capitalising each word in the selected range.

Transformations ▸ Make Small Caps Creates fake small caps by capitalising all of the letters in the selection and then using font sizes to produce the effect. If your workflow requires true small caps you should use the typography features of your font to display this format, instead.

Transformations ▸ Remove Small Caps In opposition to the previous command, removes the faux caps effect, restoring the original letter case and normalising the font size based on the larger letter found within the selection (ignoring letters using a smaller font to the left). Given how this command requires precise expected measurements between large and small letters, it will only work on text that has not had its overall font size modified after applying faux small caps.

Transformations ▸ Quotes to Smart/Straight Quotes Converts between typographic
"curly" quotes and straight speech marks.

Transformations ▸ Inline to Inspector and Inspector to Inline Conversions These four
commands allow you to convert inline annotations to linked Inspector comments, vice versa, and footnotes. These commands work on the current editing session as a whole or the current selection if applicable.

. . . . . .

Speech ▸ Access to macOS' built-in text-to-speech synthesis is provided through two menu controls for starting and stopping speech. The active text editor will be used as source text with these commands starting with

the cursor and reading downward, or if text is selected, only the selection will be read aloud.

Settings pertaining to this feature, such as which voice to use and how rapidly it should speak, are found in the Accessibility System Preference pane, under the Speech tab.

Text Tidying ▸ A few useful tools for cleaning up text and fixing links.

Text Tidying ▸ Delete Struck-Through Text Within a range of selected text, any that has been marked for removal with the Format ▸ Font ▸ Strikethrough command will be deleted. Struck-through text can also be filtered out of compiled text as an option, without removing the original text in the editor. This command is only necessary if you wish to clean up the source.

Text Tidying ▸ Update Document Links to Use Target Titles Any internal links found wi-thin the selected text will have visible text of the hyperlink changed to match the names of the documents that they are pointing to. Thus, a line of text reading "Update this" which has been selected and linked to a document named "To this", will have the visible text changed "To this". If you are using links to display cross-references in the compiled output, this command will be useful when the names of sections have been revised. Read more about compiling cross-references from internal links in Updating Link Text Automatically (section 10.1.2).

Text Tidying ▸ Replace Multiple Spaces with Single Spaces Useful for cleaning up a document that has been typed up with multiple spaces in between sentences and so forth.

Text Tidying ▸ Remove Empty Lines Between Paragraphs Strips out empty lines between paragraphs, as will often be found in emails and plain-text documents.

Text Tidying ▸ Strip Leading Tabs If the original document was typed in the fashion one might do when using a typewriter, with tabs in front of every paragraph, this command can be used to strip them all out.

Text Tidying ▸ Zap Gremlins Strips Unicode and ASCII control characters from the selection. If you are having difficulties compiling, or have found areas in your text where the cursor seems to get "stuck" when moving through ranges of text, your document may have acquired these invisible control characters from somewhere. This command will strip out all of the Unicode characters falling within the range of `#x00` to `#x1F`, save for the necessary `#x09`, `#x0A`, `#x0C` and `#X0D` characters, which are used to print spaces, line returns, page breaks and tabs in your text.

Writing Tools ▸ Provides access to few macOS tools, as well as some common search utilities.

Writing Tools ▸ Look Up in Dictionary and Thesaurus Sends the currently selected word, or the word that the cursor is currently located within, to Apple's Dictionary application. You can also use ^⌘D to do quick spot-checks, or use a "force click" action on compatible hardware.

Writing Tools ▸ Search in... Uses the currently selected text or active word to search the selected resource for results. Your preferred Web search tool (as set in Safari) will be used and, along with Wikipedia searching, will require an active Internet connection to make use of.

Writing Tools ▸ Linguistic Focus... ^⌘L Brings up a floating panel that provides access to macOS linguistic word highlighting features (subsection 20.2.2).

Writing Tools ▸ Name Generator... A tool which will generate names based on a wide variety of criteria. Read more in The Name Generator (section 20.3).

Start Dictation... fn fn For computers capable of recording audio and making use of the macOS speech-to-text dictation system, this can be used to speak words aloud and have them turned into editable text in the current editor. Settings pertaining to this feature are found in the Keyboard System Preference pane, under the "Dictation" tab.

Emoji & Symbols ^⌘Space Loads the macOS Unicode character browser. Use this to insert characters that are not found on your keyboard.

A.5 Insert Menu

The Insert menu is concerned with the addition of images, equations, tables, notation, special characters and placeholders into a valid text editor or field. Some general rules of thumb can be applied for how inserted items will be placed within the text:

— The cursor position or text selection will be used to place the selected element from this menu.

— Commands that place objects or text snippets, such as images, lines or breaks, will replace the selected text with that object.

— Those that modify formatting, such as the inline annotation feature, work just like ordinary formatting commands like bold and italic do. Selected text will have that formatting toggled, but otherwise the command modifies how we type in new text.

— For commands, like inspector comments that attach objects *to* text, the active selection will be adorned. When no selection is provided, the behaviour will be to use the nearest available word.

— Some commands insert invisible flow control characters, such as a non-breaking space, or whitespace characters as they are sometimes referred to. They will be inserted at the cursor, or replace the selected text.

Table Inserts a starter table (3 columns by 2 rows) into your document at the cursor position unless the cursor is already within a table. In both cases the command will also open a floating configuration palette containing tools for adding and manipulating tabular data in the text editor. With a preexisting text selection, the selected text will be placed within the first and leftmost cell.

For further table manipulation commands, use the Format ▸ Table ▸ submenu, or right-click within an existing table.

Image From File... This brings up a file dialogue for selecting an image from your computer to be inserted and stored within the current text document. Images can also be dragged into the editor from nearly any source capable of dragging images, such as Scrivener's binder itself, Finder and many Web browsers. Read Working with Images (section 15.7) for more information on working with images in your text.

Image Linked to File... As with the above, only the inserted image file will be *linked* to the original file on your hard disk, rather than stored it in the text document. This allows you to keep image files easily accessible to other tools as well as keeping the file size of your project trim. Refer to Linked Images (subsection 15.7.4) for further information.

Image Linked to Document ▸ This submenu presents a list of all images found within the current project. As with the above command, the image will be inserted into the editor as a link to the original image, only in this case to a file within your project binder, rather than to an external location on your disk.

It is also possible to adjust Scrivener's settings so that any images dragged from within Scrivener into an editor will be linked, rather than embedded: with the **Link to images dragged from binder into editor** option, in the "Dragging & Dropping" Behaviors pane. This may prove more convenient in projects with many images.

Comment ⇧⌘* Opens a new comment attached to the selected text (or nearest word), which can be written to in either a popup box or the inspector, depending upon your preferences. This keeps notation text out of the main text and in a more traditional "bubble" or margin note format. For more information on annotating your text, see Annotations and Footnotes (chapter 18).

Inline Annotation ⇧⌘A Converts the selected text into an inline annotation, or toggles the writing mode to annotation mode. Like using a proverbial red pen, these are useful for placing notes right in the editor, alongside the relevant text..

Footnote ^⌘8 This feature is functionally identical to attaching a Comment to the text, although for the purposes of eventually exporting the content of the note to the reader in the form of an end-of-page footnote or endnote of some variety. How this will be done depends upon your compile settings and the type of work being produced.

Inline Footnote ^⌥F Working similarly to inline annotations, this command converts the selected text into a footnote, or toggles the writing mode to footnote mode. Useful for keeping footnote content directly alongside related text.

Bibliography/Citations... ⌘Y If a bibliography manager has been set up in the "Citations" tab of the General preference pane, this command will launch and bring the set utility to the foreground, facilitating the process of selecting a citation and pasting it back into Scrivener using whatever system that application provides.

MathType Equation Create a MathType image equation at the current cursor location. When inserting an equation inline, it will attempt to match the baseline of the current text. If MathType is not properly installed, a window with a link to the download will be provided. See also: Using Equations with MathType (section 20.5).

Break ▸ Invisible characters, or whitespace characters, that can be inserted into the text to control the flow of it. The View ▸ Text Editing ▸ Show Invisibles command will cause most them to be visible in the editor as special symbols]Supported Invisible Characters with Symbolic Depictions (Table A.2)

Break ▸ Line Break ⌥⌘Return Inserts a soft-break instead of a full paragraph break. Use this when you need to create a list within a single paragraph.

Break ▸ Page Break Inserts a page break within the current text. Page breaks can be previewed with the View ▸ Text Editing ▸ Show Page View mode. In most cases, Scrivener's compile system will insert page breaks for you, relegating use of inserting them manually to niche cases.

Break ▸ Non-Breaking Space ⌥Space Inserts a special space character which will prohibit word-wrap from dividing the words joining that space. I.e. for the purposes of word-wrap, it will consider a sequence of words separated with non-breaking spaces as a single word.

Break ▸ Word Joiner Inserts an invisible Unicode character which has zero-width, but otherwise acts just as a non-breaking space, ensuring the two characters to the left and right of it are never broken between lines by word wrap.

Horizontal Line ▸ Provides a few rule lines that can be inserted at the cursor position, replacing any selected text.

Horizontal Line ▸ Centered Line Inserts a sequence of 100 underscored non-breaking spaces with centre-alignment formatting applied to the line. You can increase or decrease the size of the line by adding or deleting non-breaking spaces.

Horizontal Line ▸ Page-Spanning Line Uses an underlined tab stop, set to the width of the page as it will be compiled, to draw a line across the page. Given how Scrivener's text editor may not depict the width of the page in terms with how it will compile, the tab stop may in some cases fall before or after the editor width or simulated text block in Page View, causing the rendering of the line to collapse and become invisible.

It is also important to note that since this uses a tab stop with a fixed measurement, if the paper size for the project is changed at a later date, these kinds of lines will break until their tab stops are fixed or they are regenerated with the menu command under the new paper settings.

Horizontal Line ▸ Signature Line Inserts a sequence of 40 underscored non-breaking spaces with left-alignment formatting applied to the line. As with centred lines, you can adjust the width by adding or removing non-breaking spaces to the line.

Auto-number ▸ Inserts a placeholder that will be used to generate numerical sequences in the chosen format when compiling. A sequence such as `<$n><$n><$n>` will print as `123`. For full documentation on how to use the various placeholders, use the Help ▸ List of All Placeholders... menu command.

Draft Word Count ▸ Inserts a placeholder that indicates the word count for the entire document as compiled[2]. There are a number of rounding options available, for cases where precise counts are less important.

Draft Character Count ▸ As with "Draft Word Count", this command inserts a token for the total character count with various rounding options available.

Endnote Marker For supported formats (print/PDF/RTFD/TXT/HTML), this special placeholder will collect all of the endnotes in a project and place them at the location of the marker during compile. This will be especially useful for some academic formats, which do not place endnotes at the very end of the text. If you are using inline and inspector notation to compile both footnotes and endnotes simultaneously, this marker will not impact footnotes.

Current Date & Time ⇧⌥⌘D Inserts a plain-text date-stamp based on your system's Long Date and Short Time format.

Media Time Stamp Inserts the current time stamp from the active media player in the opposing split. This can be done while playback is in operation, or while paused. The format for the time stamp can be modified in the Behaviour: Playback preference pane (subsection B.4.7), under **Media Time Stamp format**.

A.6 View Menu

The View menu contains commands related to changing the way documents are viewed, allowing you to show and hide various elements, navigate between views and customise the way information in the current project is displayed. A rule of thumb is, if you want to change the way something in your project looks or acts, and it's not found in project settings (Appendix C) nor in the main preferences, chances are it is in the View menu.

As with project settings, the options you select here will only impact the active project. If you find you prefer a particular setting in general, you might want to look into creating a starter template for yourself (section 5.4).

A.6.1 View Modes

The three view mode selections at the top of this menu are different ways you can view the the contents of the editor. Setting a view mode determines how the active editor will view groups of items (such as when clicking on a folder or

[2] The particulars for what constitutes the "entire document" are defined by the compile format. Some may include footnotes or front matter in their counts where others may not.

selecting two or more items) going forward, until you change view modes again. Within the menu, a checkmark will be placed beside the view mode currently in use.

When selecting a container and switching to corkboard or outline mode, the child items of that container will be displayed in the view. If a single item is selected when switching to one of these modes, an empty view will be displayed so you can easily add new child items beneath it (probably turning it into a file group).

These actions only pertains to the main editors (not Quick Reference panels or Copyholders, which only view the document content) and is a separate setting per split. You can thus keep one split in Scrivenings mode and the other in Corkboard if you wish. For an overview of how view modes work, see View Modes (section 4.2).

Document/Scrivenings ⌘1 If multiple text documents are selected in the editor, this choice displays the current editor as a Scrivenings session (section 15.10), showing the text of each document one after the other, viewable and editable as if they were a single file. Any media or research files included in the selection will be skipped over.

When a single non-group item has been selected in the editor this menu item will be labeled "Document", and switch the content to its text or media view.

Corkboard ⌘2 Display the current editor as a corkboard (section 8.2).

Outline ⌘3 Display the current editor as an outline (section 8.3).

Show|Hide Binder ⌥⌘B Toggles whether or not the binder sidebar (subsection 4.1.4) is visible on the left of the main window. You can hide the binder to concentrate on editing or composing the current document if you so wish.

Show|Hide Collections Reveals or hides the collection interface which will appear above the binder. Note the sidebar will need to be visible for this menu option to be available.

Show|Hide Inspector ⌥⌘I Toggles whether or not the inspector is visible on the right of the main window. The inspector displays all metadata for the current document, including synopsis and notes, and is hidden by default. Read more about this sidebar (chapter 13).

Editor Layout ▸ These commands impact the overall layout of the editor space, between the binder and inspector sidebars. Splitting the editor to show more than one thing at a time, as well as attaching "copyholders" to clip files to the editor splits and cleaning up the editor to provide a distraction-free interface are all located here. For what goes on *inside* the editor, you might be more interested in the following View ▸ Text Editing ▸ submenu.

Editor Layout ▸ Show|Hide Header & Footer View Toggles the visibility of the header and footer bar in the active editor.

Editor Layout ▸ No Split ⌘' Removes the inactive split from the editor, returning it to one. For more information on using splits, see Splitting the Editor (subsection 8.1.4). This command does not remove copyholders directly.

Editor Layout ▸ Split Horizontally ⌥⌘= Initiates a new horizontal split, or converts the split orientation to horizontal. When creating a new split, the current content will be loaded into that split.

Editor Layout ▸ Split Vertically ⌘" As with splitting horizontally, only vertically instead.

Editor Layout ▸ Swap Editors Only available when Split Horizontally or Split Vertically is enabled. This command swaps the editors from one side to the other, so that the view on the top/left will become the view on the bottom/right and vice versa. This swaps the entire editor session, its history, settings and so forth. What was the left editor will now be functioning on the right side instead.

If you would instead prefer to merely mirror the content into the other editor without swapping settings, use the "Match Split Documents" command in the header bar contextual menu (section 8.1.1).

Editor Layout ▸ Copyholder Position ▸ Also accessible by right-clicking on the copyholder's header bar directly, you can set which side of the editor it favours using the options in this submenu. A copyholder cannot be positioned parallel with a split, so some options may not always be available, or be swapped for you when the spit configuration changes.

Table A.2: Supported Invisible Characters with Symbolic Depictions

Invisible Character	Symbol
Space	·
Carriage Return	¶
Line Feed	↩
Tab	→
Non-breaking Space	␣
Page Break	❙
Word-Joiner	(Not depicted.)

Text Editing ▸ Formatting tools for working with text can be toggled in this submenu, as well as visualisation tools like page view, line numbers, vertical writing mode and so forth.

Text Editing ▸ Show|Hide Format Bar ⇧⌘R Toggles the visibility of the character and paragraph formatting bar, which provides quick access to many of the most common types of formatting tools. Refer to The Format Bar (subsection 15.5.2) for more information on this tool.

Text Editing ▸ Show|Hide Ruler ⌘R Toggles the visibility of the tab-stop and indent ruler (subsection 15.5.1) for the active editor.

Text Editing ▸ Show Titles in Scrivenings When enabled, editable titles of documents will be placed between documents in a scrivenings session above their associated texts. Titles can be formatted, and the divider style changed, in the Appearance: Scrivenings pane (subsection B.5.13).

Text Editing ▸ Only Show Scrivenings Titles for Folders The editable titles added by the former menu toggle will only appear on folders, with all other items will use a regular divider.[3]

Text Editing ▸ Show|Hide Invisibles Toggles visibility of hidden control characters, such as paragraph breaks, table cells (when otherwise set to have no borders) tabs, spaces and page breaks (Table A.2). You can modify the colour used to depict these symbols in the Appearance: Textual Marks (subsection B.5.16) pane.

Text Editing ▸ Show|Hide Markup Toggles the display of textual markings, to produce a cleaner copy suitable for proofreading. The markings that will be hidden by default are comment highlights, links, highlight boxes around styled text and preserve formatting boxes. You can adjust which markings are hidden in the Appearance: Textual Marks preference pane (subsection B.5.16).

Text Editing ▸ Show|Hide Page View ⇧⌥⌘P Toggles displaying the text editor as a virtual page on and off, rather than text in a long column. It is not intended to be used as a layout mechanism, though it can provide a reasonable estimate of how your pages will look under certain conditions, once printed. This setting is specific to each split editor, as well as composition mode. Read more in Page View (chapter 16).

Text Editing ▸ Two Pages Across When enabled, displays two pages side by side in a familiar book-style arrangement.

---

[3] This behaviour is expanded to include all containers when **Treat all documents with subdocuments as folders** is set to true, in the Behaviors: Folders & Files preference pane (subsection B.4.5).

Text Editing ▸ Use Vertical Layout This is a global setting that toggles the rotation of the main editors (including composition mode) to vertical alignment, as is sometimes used in Eastern scripts like Japanese.

Text Editing ▸ Show Line Numbers Adds a paragraph (line feeds within a paragraph will not be counted) numbering feature to the standard editor. This feature is unavailable to Page View mode.

Text Editing ▸ Only Count Every Fifth Line As an option to showing line numbers above, you can elect to have Scrivener only count every fifth paragraph.

Text Editing ▸ Show|Hide Compiled Footnote Numbers in Inspector When enabled, after you compile and all footnotes have been counted, their numbering will be printed in the Inspector alongside the footnote itself (section 18.3.6). This option can slow down compile times, and is thus off by default.

Text Editing ▸ Prompt Before Updating Footnote Numbers In conjunction with footnote numbering, optionally instruct the compiler to ask you if these numbers should be updated, every time you compile. This can be of use if the project is very large, as adding numbers takes time and may not always be necessary.

Text Editing ▸ Typewriter Scrolling ^⌘T Toggles a feature that keeps the currently edited line of text at a set position on the screen, much like typing on a physical typewriter. This setting is specific to each split editor, as well as to composition mode (the latter has the feature enabled by default).

. . . . . .

PDF Display ▸ This submenu controls how PDFs are displayed in the editor. Most of these options should be familiar as they are common to many PDF viewers.

PDF Display ▸ Automatically Resize Keeps the PDF sized to the width of the editor window even when changed.

PDF Display ▸ Actual Size Zooms the PDF to its native scale.

PDF Display ▸ Size to Fit Like "Automatically Resize", but only resizes the document once. If you change the size of the editor view later, it will stay at the same zoom level.

PDF Display ▸ Single|Facing Pages In Single Page mode, will show one page per row like an ordinary digital file. In facing mode, two pages will be placed side-by-side in a column, more like reading a book.

PDF Display ▸ Continuous /Page Breaks These two scrolling modes affect how the document is handled when using the scroll wheel, arrow keys, or page up and down keys. In Continuous mode, the pages flow by seamlessly; in page break mode, only one (or two, if facing pages is enabled) pages will be showed at a time, and scrolling actions flip between pages.

. . . . . .

Corkboard Options ▸ Provides features and visual options for the corkboard (section 8.2). Most of the settings in this menu are specific to each split, affording different view settings for each editor. Note that further options can be accessed via the corkboard display options button, which is located on the right-hand side of the footer bar for each corkboard.

Also refer to So What are Index Cards, Anyway? (subsection 8.2.1), for more information on the index card itself.

Corkboard Options ▸ Cards Across ▸ The Cards Across submenu allows you to define how many index cards you would like to appear in each row on the corkboard. The default is three. Auto-fit will calculate how many cards to show, based on the size of the editor and the size of the cards; kind of like word wrap, for index cards.

Corkboard Options ▸ Arrange by Label Toggles a mode whereby index cards are arranged by label on the corkboard (subsection 8.2.5). Each label forms a "rail" that matching cards are placed upon in the order they appear within the corkboard naturally. Whether these rails form rows or columns can be adjusted with the following menu options.

Corkboard Options ▸ Arrange by Label Layout ▸ These choices select between the rotation of the corkboard layout or whether it wraps cards to the view, as described in Stacked Corkboards (subsection 8.2.8). Grid orientation (or how the corkboard behaves by default) is only available when stacking, never when Arrange by Label is enabled, even if stacking a label-arranged view. These options are also available as buttons along the bottom right in the corkboard footer bar, and on the Touch Bar with appropriate hardware.

The name of this menu will change depending upon the context. By default it will use the above title (or whatever you refer to as "label" in the project settings), but if the corkboard is displaying multiple containers in a "stack", then the command will read "Stacked Groups Layout".

Corkboard Options ▸ Show Label Colors Along Edges ^⌘P Toggles whether a coloured strip is drawn along the edge of index cards. These strips are associated with the label colour assigned to the item the card represents. When no label has been assigned to a card no strip will be drawn on it.

Corkboard Options ▸ Show Status Stamps ^⌘S Toggles whether stamps are displayed on index cards. Stamps show the current status associated with the document represented by the index card as though it has been stamped on the index card. The appearance of stamps can be adjusted in the "Fonts" and "Colors" tabs of the Appearance: Corkboard preference pane (subsection B.5.4).

Corkboard Options ▸ Show Keyword Colors ^⌘K Keywords assigned to a document can be visually indicated along the right-hand side of the index card as colour chips. The number of keywords that can be shown at once can be changed using the corkboard display options button.

Corkboard Options ▸ Show Card Numbers Cards will be numbered according to their sequence in the selection or container they are within. In Freeform mode the numbers will not change according to where the card is positioned in the view. This can be a valuable reference as the order of cards are likely to become shuffled up. This option impacts both splits.

Corkboard Options ▸ Number Per Section When this option is enabled, if the corkboard is displaying multiple containers in a "stack", card numbering will be restarted for each new section. When disabled card numbering will be linear across sections.

Corkboard Options ▸ Freeform Activates the Freeform Corkboard (subsection 8.2.4) mode, allowing unrestricted movement of cards around on the corkboard without any immediate impact on their underlying order in the binder. This can also be toggled in the footer bar, and on the Touch Bar with appropriate hardware.

Corkboard Options ▸ Snap to Grid Toggles whether or not cards in freeform mode will snap to a background grid, rather than allowing for pure freeform placement.

Corkboard Options ▸ Commit Freeform Order Uses the current freeform layout to reorder the actual manuscript outline structure. Scrivener will provide several options for how this can be done in a dialogue box (section 8.2.4). If card numbers are enabled, they will be renumbered at this time to reflect the new underlying order. This command can also be invoked via the Commit button in the corkboard footer bar.

. . . . . .

Outliner Options ▸ Displays a list of toggle commands that reveal or hide the corresponding columns in the active Outliner view (subsection 8.3.1). These settings are stored per split, affording different view settings for each editor. You can also manage columns using the › button above the scrollbar area in the outliner itself.

Center Content Toggles whether the current outliner view will have its content horizontally centred (much like the text editor centres the text column by default). This will be more useful with a limited number of columns, and will have no visible effect if there is more outliner width than the current view affords. This behaviour can also be toggled via a button in the outliner footer bar (subsection 8.3.6).

Use Fixed Row Height Toggles whether or not the current outliner view will used a fixed height cell for each row, rather than an adaptive height that conforms to the amount of content in various columns. For more information, refer to Using a Fixed Row Height (subsection 8.3.5).

Use Label Colour In ▸ An item's label colour can be applied as a tint to various interface elements throughout the project window, toggled by the choices in this menu, listed below. The intensity of most background tinting can be adjusted in the Appearance: General Interface preference pane (subsection B.5.1). Read about the different options available in this menu in Label Colours (section 10.4.1).

Use Label Colour In ▸ Show as Background Color in Binder When ticked, this option causes the binder title to be coloured with the label background, rather than using a dot to the right of the name. This option has no impact on the other sidebar views. The intensity of this background can be modified by the opacity slider (which impacts the label tint used in other background areas) in the Appearance: General Interface preference pane (subsection B.5.1).

Zoom ▸ This all purpose tool zooms the display of a compatible view up or down, and also allows for precise selection for text view types. Each split records its own setting, as well as composition mode. The menu adds shortcuts and extends access to this capability into areas that lack that a visible zoom control, such as Quick Reference panels and inspector panels with text. Refer to Scaling Text (subsection 15.3.1) for more information.

The tool will also increase or decrease the size of viewed media, such as PDF or images.

Zoom In ⌘> Increases the magnification of the current view by a set amount. In text views this amount correlates with the provided values in the "Text Zoom" section of the menu, or what can be seen in the zoom tool in the text editor footer bar.

Zoom Out ⌘< Decreases the magnification of the current view in an inverse pattern to zooming in.

Text Zoom The "Text Zoom" section of the menu lists a number of convenient and popular zoom settings for text.

Fit Width Adjusts the magnification so that the page width (including margins) fits the current editor width. This and the following option are only available when using Page View mode in the text editor (chapter 16).

Fit Page Sets the magnification so that both the short and long edge of the page can be seen within the editor.

Other... Provides a field where you can type in a precise magnification amount yourself. If you'd prefer a number between 200% and 300% say, this would give you the option for doing so.

. . . . . .

Outline This submenu provides commands for working with groups in both the outliner and binder, followed by settings and navigation commands for the binder.

OutlineExpand All ⌘9 Expands all collapsed items recursively within the current view. When this command is used in the binder, it will expand the entire binder. If you instead wish to only expand or collapse one single section of the outline, you can use the Option key in conjunction with a mouse click or the ← (to collapse) or → (to expand).

This command can also be used in the Footnotes & Comments inspector pane to expand all collapsed notes to full height.

OutlineCollapse All ⌘0 As with Expand All, this works the other way, closing all visible open items recursively.

This command can also be used in the Footnotes & Comments inspector pane to collapse all notes to single lines.

Collapse All to Current Level ^⌘0 Working in a similar fashion to Collapse All, this command will collapse the entire outline below the currently selected item's level. Thus if you if have an outline that has six levels of depth, and select a folder on level 3, running this command will completely collapse all items except those at levels one, two and three, which will be ignored.

When multiple items are selected, the first item in the selection will be used to determine the level by which the tree will be collapsed.

Show Subdocument Counts in Binder When enabled, each container in the binder will display a number to the right of it showing how many subdocuments it contains. This number is recursive, meaning that it will not only count the container's immediate children but any descendants beneath those children as well.

Hoist Binder With this command, temporarily obscure the full binder and focus on only one portion of it. In other words, hoisting will display the selected container all by itself in the binder sidebar. Changes made to ordering and structure while hoisting will be made to the book outline itself. Read more about it in Hoisting the Binder (section 6.3.5).

Unhoist Binder Returns the binder to full display. You can also click the × button in the binder sidebar header to return to the full binder.

Enter|Exit Full Screen ^⌘F Utilises macOS' full screen implementation, expanding the main project window to occupy the entire screen, and moving it to its own virtual desktop (or "Space" as Apple refers to them), segregated from the rest of the applications running on your system. The window cannot be resized or moved until Exit Full Screen has been invoked.

Exiting Full Screen mode will return the project window to the original position and size (and when using a default full screen layout (subsection 12.3.4), to its original layout as well) in use before moving the project window to full screen mode. Read more about it in Full Screen Mode (section 4.1.7).

Enter|Exit Composition Mode ⌥⌘F Toggles the distraction-free writing environment (chapter 17), switching to text editing mode if necessary. Being a text editing mode, this option is not available to research files or the corkboard and outliner group view modes. There are three methods of invocation:

1. In the simplest case, if you are editing a single document or have one file selected in any context, it will be loaded in composition mode.
2. If the editor is displaying a Scrivenings session, the entire session will be opened in composition mode.
3. If multiple documents are selected in any context they will be loaded into the composition mode's history, so you can flip through them using the keyboard shortcuts, ⌘[and ⌘].

Show|Hide Tab Bar Toggles the visibility of Apple's Tab Bar interface, which allows for the merging of similar windows together into a single tabbed window. In Scrivener, this means you can combine projects into a single window, and Quick Reference panels together into their own groupings. The Window ▸ Merge All Windows command is used to combine two or more alike windows together. For further information on how to use this feature, consult Apple's documentation.

Show|Hide Toolbar Toggles the visibility of the main application toolbar.

Customize Toolbar... Opens the configuration panel for the main application toolbar. This can also be done by right clicking on the toolbar itself.

Customize Touch Bar... Brings up the standard interface for customising keyboard buttons on Apple's Touch Bar technology. As with most software, you will need to activate the area of the interface first in order to customise the button set that is available to it. For example to adjust which buttons are available in the binder, click in the binder first. Read more about touch bar (subsection 4.3.2).

This command will not be visible without the proper hardware.

A.7 Navigate Menu

The Navigate menu focusses on all aspects of getting around inside of a project, from flipping between available binder sidebar tabs, to jumping from one place to the next in the editors, controlling media playback to moving your cursor around within the project window itself in a mouseless fashion. It also contains a few project settings that impact how navigation works with the split editors and copyholders available.

Reveal in Binder ⌥⌘R Displays the location of the currently edited file in the binder, opening the sidebar and switching to the binder if necessary (you can also use the ⌥⌘R shortcut). It will also expand any containers to reveal the position of the item if it is nested. When used from the icon header bar menu with a multiple selection, all of the entries included in the selection will be highlighted in the binder.[4]

This is most useful when the method you used to arrive at the current document did not involve clicking in the binder (such as using the history navigation buttons or using a link), or if you are currently viewing a collection and wish to find where the file is actually located in your project outline. This command is also available from the binder contextual menu.

This command is available in a few different contexts, and is also available from the header bar contextual menu (section 8.1.1):

- Editors
- Copyholders
- Quick Reference panels
- Binder sidebar views like Search Results, collections and hoisted containers.

[4] If you wish to reveal an item in the collection you are viewing instead, use the "Reveal in Collection" command.

Reveal Draft Folder Opens the binder sidebar if necessary, switching the main binder view out of any collections, search results or hoisted containers, and selects the draft folder.

Go To ▸ The resulting action of this command will be similar to clicking on an item in the binder: the active editor will *go to* the item you specify. If the binder is hidden, a collection is selected, or a section is hoisted, for instance, you can use this menu to navigate to other areas of the project without having to alter your work environment. The menu also provides a few contextual navigation and focus functions as well as listing any project bookmarks for convenient access.

Beyond the commands listed at the top, the remainder of this menu will list first any project bookmarks that are defined, for convenient access at the very top, and then this will be followed by a binder item selection submenu. This portion of the menu can also be accessed from the header bar icon menu and from within composition mode using the control strip along the bottom of the screen.

Go To ▸ Previous Document ⌥⌘↑ Jumps to the previous item in the binder sidebar list. When used with the main binder view visible, this command ignores hierarchy.

Go To ▸ Next Document ⌥⌘↓ As with Previous Document, but selects the next document in the binder sidebar list.

Go To ▸ Enclosing Group ^⌘R Will display the currently edited item in context with its siblings using the current view mode, and can be thought of as a way of moving your editor "up" in the hierarchy one step at a time. In most cases, this will select the immediate parent of the document, selecting the document's index card or outliner row that you came from, or expand your current Scrivenings session to include the siblings and parent of the current text item.

Go To ▸ Selection ⌘4 Jumps directly to the text component of any selected item. This works as an isolation feature in Scrivenings, or can also be used to quickly open selected items from the corkboard or outliner as text (even if they are containers). When more than one item is selected, they will be loaded using Scrivenings mode as a multiple selection.

The difference between this and the Navigate ▸ Open ▸ commands (⌥⌘O or ⇧⌘O) is that they will respect the current view mode preference (so if you select a folder card in a corkboard and use an "open" command the result will be to load that folder as a corkboard). ⌘4 will *always* load the selected item as a text file—without changing your current view mode.

Go To ▸ Collection ▸ This submenu contains a list of all available collections in the project. Rather than loading the collection in the sidebar, as would

ordinarily be done, this command loads the contents of the collection into the editor, where it can be worked with using any of the group view modes.

If the collection is already being displayed in the sidebar, you can also load it into the editor by clicking on the ↱ button in the sidebar header (beside the × button) . Read more about this capability in Viewing the Contents of a Collection in the Editor (section 10.2.1).

Open Quick Reference ▸ This provides a similar list of project bookmarks and binder items as the Navigate ▸ Go To ▸ submenu, except that the selected item will be opened in a Quick Reference panel, rather than replacing the contents of the editor.

Collections ▸ Provides commands for navigating to collections in the binder sidebar, as well as one conversion command. This menu will be populated by a list of the collections in this project, in reverse order from how they appear in the tab list. It will always include an entry for the binder and "Search Results". Selecting an entry will switch the sidebar to viewing that collection, making this menu useful when the collection tab list (subsection 10.2.1) is not visible, or for assigning keyboard shortcuts to oft-used collections.

For documentation on how to use collections, see Using Collections (section 10.2).

Collections ▸ Convert to Standard Collection When a Search Result Collection is the active tab, this command will "freeze" the results and turn it into a regular collection that is no longer dynamically updated. This is a one-way process that cannot be undone, and the original stored search settings will be discarded. Refer to Converting a Saved Search to a Standard Collection (section 10.2.4) for more.

Collections ▸ Next Collection Select the next collection tab in the stack, as shown in the tab list.

Collections ▸ Previous Collection Select the previous collection tab in the stack, as shown in the tab list.

Open ▸ The Open submenu provides a number of ways to open a selected item. These menu items are relevant from the binder or collection views, as well as within the corkboard and outliner views, but note that in most cases, a single click in the sidebar or double-click on the icon in the editor, will open a file more directly. This command is also available from the binder contextual menu.

Open ▸ In Current Editor ⌥⌘O The actual labels of this and the next menu command will change depending upon the current editor that is active, and whether or not split orientation is horizontal or vertical, to make it more clear which editor the item will be loaded in.

Use this command to open the selected items in the current split, replacing what you are currently working with.

Open ▸ in Other Editor ⇧⌘O Open the selected items in the inactive split, opening a new split if necessary to do so. For example you can select a card in a corkboard and use this keyboard shortcut to open a split and load the contents of that card into the split.

Open ▸ as Quick Reference Opens the selected document in a Quick Reference panel (section 12.6). When in a Scrivenings session, the current segment of the overall session that is currently active will be opened as a Quick Reference panel.

Open ▸ in Copyholder Opens the selected document in the copyholder for the current split (subsection 8.1.5), opening a new one if necessary. When used in a Scrivenings session, the currently active segment of text will be loaded alone.

Open ▸ in Quick Look This menu command is only available for types of files that Scrivener does not support with its built-in viewer. It will make use of your Mac's native Quick Look viewer, similar to having selected a file of that type in Finder and opening Quick Look from there.

Open ▸ (with) Compilable Subdocuments Working in a manner similar to "With All Subdocuments as Flat List" (below), this loads into the view only those items that are set to include in compile (subsection 13.5.1). Read more about filtering items in group view (section 11.4).

Open ▸ With All Subdocuments as Flat List ▸ Useful for cases where you'd like to see all of the cards in a particular section of your outline, even if they are nested several layers deep. The result of this command is a multiple selection, which means the items cannot be reordered as all depth information is lost by viewing cards this way.

- *On Editor Corkboard*: replaces the current editor view with the selected item's contents. This is similar to pressing ⌥⌘O to open selected cards on the corkboard, only it also will add all of their descendants to the corkboard as well.
- *On Other Editor Corkboard*: as with above, but uses the inactive split to open the item, instead of replacing the current view, creating a split if necessary.

Open ▸ in External Editor ^⌘O This command will open a file using the default editor for that type of file (for example, loading a PDF in Preview). This command is available on any type of item that is not a text or folder document. Opening items this way allows them to be viewed and edited in their native applications. Any edits made externally will be saved back into the project seamlessly. If Scrivener can display the type of file you edited, you might need to refresh the viewer to see your changes (section 8.1.3).

. . . . . .

Editor ▸ This submenu contains navigation commands that impact the editor and any content that it is viewing. When multiple splits are open, the commands target the active split. Some of the commands may also be available from Quick Reference and copyholder panes, and will be noted if so.

Editor ▸ Lock in Place ⌥⌘L Locks the editor (or split) so that no external navigation commands (such as clicking in the binder) will affect it, causing the other editor split, if available, to load the request instead (this behaviour can be adjusted in the Behaviors: Navigation preference pane, by setting **When focused editor is locked in place** to "Binder selection does nothing"). When an editor is locked, its header bar will turn a shade of red. Read more in Locking the Editor (subsection 12.2.1).

Editor ▸ Lock Inspector to Editor Available when the editor is split, this command locks the inspector sidebar to the currently active split. Ordinarily the inspector tracks the selection within whichever split you are currently working in. When this is engaged, the inspector header bar will turn red and the split it is locked to will be indicated by a red "inspector" dot in the header bar. You can now use the other split while leaving inspector material alone and available for reference. Read more about Locking the Inspector (subsection 13.1.1).

Editor ▸ Lock Group View Mode Available only when the active editor is displaying a group view, or if the selected item is a container (folder, or file group acting as a folder). Any view mode can be locked, including single text document mode. Locking is a per-item setting, and is saved into the folder or container itself. The view mode will stick to what was in use at the time it was locked. You can change its saved view mode by simply switching the view mode while viewing the item. Read more about Locking the Group View Mode (subsection 12.2.2).

Editor ▸ Scroll to Previous Page These menu commands will only activate when Page View is enabled on the active editor, or when viewing a PDF. This command will scroll the view so that the top of the previous page is flush with the top of the viewer, keeping the reading area stable as you browse.

Editor ▸ Scroll to Next Page This command scrolls the view so that the top of the next page is flush with the top of the viewer, keeping the reading area stable as you browse.

Editor ▸ Forward in Document History ⌘] Much like a Web browser, the editor keeps track of everything you've visited within the project. Using these commands you can navigate back and forth in the history. You can also use the Forward and Backward buttons in the header bar (right-click on them to access the full history as a list). Each split keeps its own history.

This command visits the document you were looking at prior to having gone back in history to the current document.

Editor ▸ Backward in Document History ⌘[This command visits the document you were looking at before you navigated to the current one, and if you use it again, to the one you were visiting before *that*, and so on.

Editor ▸ Other Editor ▸ There are a few commands accessible as keyboard shortcuts while typing in one editor that have an impact on the *other* editor without removing your typing focus from the current editor. I.e. You can scroll the other split and operate its history even while writing in another document.

Other Editor ▸ Forward in History ⌥⌘] Visits the document that was being viewed prior to having gone back in history in the other split.

Other Editor ▸ Backward in History ⌥⌘[And as you might expect, this visits the document being viewed in the other split prior to whatever you are viewing in it currently.

Other Editor ▸ Scroll Up ^⌥⌘↑ Scrolls the other editor back a page, similar to having pressed the PgUp key with it active.

Other Editor ▸ Scroll Down ^⌥⌘↓ Scrolls the other editor forward a page, similar to having pressed the PgDn key with it active.

Editor ▸ Clear Document History Use this command to wipe out the document history for the active editor.

Media ▸ Controls and options for various types of playable media. These commands will only become available when viewing the appropriate type of media. Read more in Viewing Multimedia Documents (section 8.1.3).

If the editor is split this shortcut will start and stop the media even from the *other* split, allowing you to easily transcribe or reference the media while working elsewhere. When two media files are open at once, the shortcut will affect the active split.

Media ▸ Play Media File ⌘Return Functionally, this acts like clicking the Play or Pause button in the media viewer, for either audio or video files.

Media ▸ Fast Forward ^⌘] Jump playback forward by two seconds. This shortcut works while playback is running or paused. In the latter case, it will stay paused at the new position.

Media ▸ Rewind ^⌘[Jump playback backward by two seconds, and otherwise follows all of the rules outlined above.

Media ▸ Rewind on Pause This project setting toggles the behaviour whereby whenever media playback is paused, upon resumption it will be rewound by a determined number of seconds in the Behaviors: Playback preference pane (subsection B.4.7).

. . . . . .

Move Focus To ▸ Provides application focus navigation tools. Rather than navigating around in your project, these commands will let you quickly select different parts of the project window without using the mouse. The first lets you cycle between common elements, while the rest will jump immediately to that element of the interface, no matter where focus is currently placed. Note that in all cases, the elements you wish to cycle or jump to must actually already be visible. These shortcuts will not automatically reveal parts of the interface that are currently hidden.

Move Focus To ▸ Rotate through main views ^Tab This menu item follows a chain of the three most common areas of desired focus: the two editor splits and the binder. It will cycle between these three going left to right, and the label of this menu item will be changed to indicate where the next target will be.

Move Focus To ▸ Binder ^⌥⌘B Moves focus to the binder sidebar from anywhere in the interface, if visible.

Move Focus To ▸ Left|Bottom Editor ^⌥⌘E Moves focus to the Left or Bottom editor, if the editor has been split. When the view is not split, this command will be named “Editor” and will select the main editor.

Move Focus To ▸ Right|Top Editor ^⌥⌘R Moves focus to the Right or Top editor, if the editor has been split. When the view is not split, this command will be named “Other Editor”, and will naturally be disabled.

Move Focus To ▸ Header Bar Title ^⌥⌘T Moves focus to the editor header bar title area where you can edit the name of the item you are currently viewing in the editor. If you are viewing something that cannot be renamed (such as a multiple selection) then the command will be disabled. Use Return to confirm your changes and bring the typing focus back into the main viewing area of the editor.

Move Focus To ▸ Copyholder ^⌥⌘D If only one copyholder is visible on the screen (or none), then this command will be named "Copyholder", and target it no matter which split the focus is current in. Otherwise it will target the copyholder that is attached to the editor that is currently active, and the menu name will be altered accordingly.

. . . . . .

Inspect ▸ In continuation of the keyboard navigation functions, these deal solely with the revealing and focussing of tabs, and occasionally panes within, of the inspector (chapter 13). The shortcuts work on the following principles:

1. The inspector itself will be revealed if necessary when the command is used.
2. If the keyboard shortcut cause the inspector to switch to its respective tab, then the action will simply reveal it without disturbing your original typing focus.
3. If the tab is already visible, then the shortcut will *switch keyboard focus* to the associated metadata area within that tab, allowing you to interact with its contents in a mouseless fashion. If the targeted pane has been collapsed within the inspector then this shortcut will also expand it.

Inspect ▸ Notes ^⌥⌘H Reveals or switches focus to the notes tab. Use the Synopsis command below to focus the upper portion of this tab.

Inspect ▸ Bookmarks ^⌥⌘N Reveals or switches focus to the bookmark list. Use ⌘6 to switch between viewing project and document bookmarks. Once you have selected a bookmark you wish to edit in the preview area below, hit the Tab key to switch panes. Return will load the selected bookmark in accordance with the settings in the Behaviors: Document Links preference pane (subsection B.4.2).

Inspect ▸ Metadata ^⌥⌘J Reveals the metadata tab, and switches focus to the Custom Metadata pane within it. Use Tab and ⇧Tab to navigate between text and date fields. Use the Keywords command below to switch focus to the lower third of this tab.

Inspect ▸ Snapshots ^⌥⌘M Reveals or switches focus to the snapshot list. Use arrow keys to flip between available snapshots and view their contents in the preview area below.

Inspect ▸ Comments and Footnotes ^⌥⌘K Reveals or switches focus to the comments & footnotes tab. The ↑ and ↓ keys can be used to navigate through the list, and → and ← to expand and collapse selected notes.

Inspect ▸ Synopsis ^⌥⌘I Reveals the notes tab and switches focus to the synopsis text area. Use the Notes command above to focus on the lower portion of this tab. Use the ⌘7 keyboard shortcut to switch between viewing the synopsis as an image or text.

Inspect ▸ Keywords ^⌥⌘L Reveals the metadata tab, and switches focus to the Keywords pane within it. Arrow keys can be used to navigate amongst keywords; Return can be used to add new keywords; and Delete to remove keywords.

. . . . . .

Outline Groups ▸ These commands make it easy to jump from one group to the next, where group is defined as any folder in the binder. This behaviour can be altered to also include all containers with subdocuments, in the Behaviors: Folders & Files preference pane (subsection B.4.5).

Outline Groups ▸ Previous Group ^⌥↑ Jumps the selection to the previous visible container in the active outliner or binder, no matter what the depth. If a container is currently hidden, it will be skipped. If you only wish to select the parent container of a the current item, you can use the ← key.

Outline Groups ▸ Next Group ^⌥↓ Working in the same fashion as Previous Group, only selecting the *next* visible container downward from the current position.

. . . . . .

Binder Selection Affects ▸ These project settings toggle which split binder sidebar clicks will be opened in. By default when you click on something in the sidebar it will be sent to the targeted editor (so long as it is not locked). This will adjust that behaviour so that the clicks go to a predetermined editor in all cases, depending on the choice made below. If the editor is not split, this menu will be deactivated as all clicks will naturally go to the only available editor.

When any of the following options besides "Current Editor" is selected, the targeted header bar(s) will be coloured blue to indicate where clicks will be sent. Read more about the header bar (subsection 8.1.1).

This command is also available from the binder contextual menu.

Binder Selection Affects ▸ Current Editor This is the default behaviour out of the box. Whichever split you worked in last will be targeted.

Binder Selection Affects ▸ Other Editor Whichever split is *not* currently active will be targeted. This way your current work space is never disturbed by usage of the binder.

Binder Selection Affects ▸ Top|Left Editor Only The top or left split (depending upon orientation) will always take binder clicks no matter which split is active.

Binder Selection Affects ▸ Bottom|Right Editor Only The bottom or right split (depending upon orientation) will always take binder clicks no matter which split is active.

Binder Selection Affects ▸ Both Editors In this special case, *both* editors will update whenever you click on something in the binder. This will most often be useful when using two different group view modes. You could have an outliner in the left view and a Scrivenings session in the right, both showing the same area of the binder in their own unique ways.

Binder Selection Affects ▸ None Neither editor will update when using the binder. In this mode of usage, you will always need to manually open items into an editor. This can be done by using the Open commands in the right-click menu or Navigate menu, or by drag and drop into the header bar.

. . . . . .

Outliner|Corkboard Selection Affects ▸ The project settings in this submenu impact how splits interact with one another. When active, the effect is to cause the active split to act a bit like the binder sidebar, in that whatever you click on within it will automatically be loaded in the other split.

This behaviour can also be toggled in the footer bar with a toggle button that rotates between the available modes in this menu (subsection 12.2.5), and from the Touch Bar. If there is neither a split editor or a copyholder then these options will be disabled.

Outliner|Corkboard Selection Affects ▸ None This is the default behaviour. Clicking on rows in the outliner or cards in the corkboard will never automatically open those items in other views.

Outliner|Corkboard Selection Affects ▸ Other Editor When selecting items in the corkboard or outliner view, they will be automatically loaded into the other split view using its preferred view settings.

Outliner|Corkboard Selection Affects ▸ Copyholder If a copyholder has been attached to the current editor, you can elect to have any individually selected items loaded within it automatically, turning it into a viewer for the split.

. . . . . .

Clear All Navigation Options Removes all settings from the active project window that impact navigation, as described in Clearing Navigation Settings (section 12.5).

A.8 Project Menu

The Project menu addresses commands and configuration options specific to the active project, such as document format over-rides, custom metadata and statistics. If more than one project is open, this menu will make use of the foremost project window.

The initial commands listed here that add new items to the binder are also accessible within an "Add" submenu in the binder contextual menu.

New Text ⌘N Creates a new text document below the current selection or list of documents within the current view. When a folder is selected, the new file will be created within the folder.[5] Otherwise the new file will appear as a sibling below the current selection, or the location of the current document that is being edited.

When the current position for a new item is within a container that has a default subdocument template (subsection 7.5.2) assigned to it, the menu label will change to reflect that. So for example if the current folder has a default subdocument type of "Interview Transcription", then the menu label will read, "New Interview Transcription".

New Folder ⌥⌘N Creates a new folder below the current selection or list of documents within the current view. Folders will always be created as a sibling to the current selection, save for when that selection is either the Draft or Research special root folders.

For more information on how items are placed when creating them, read Figuring Out Where Things Will Go (section 6.3.1).

New From Template ▸ ⇧⌥⌘N This submenu displays all of the document templates found within the current project in hierarchical order. If no template folder has been assigned, the menu will simply contain a message to that effect. Read more about Document Templates (section 7.5) if you'd like to create your own document types.

If you select a container from this menu, not only will that container be created as a new document, but all of its child templates will be brought along as well. You can thus create complex boilerplate structures that can be created at will.

Project Settings... ⌥⌘, If a setting isn't established in the main application preferences, or in one of Scrivener's menus, then chances are it can be found within the Project Settings panel. Section types, labels, status, custom metadata, text formatting, auto-completion lists, document templates,

[5] This behaviour is expanded to include all containers when **Treat all documents with subdocuments as folders** is set to true, in the Behaviors: Folders & Files preference pane (subsection B.4.5).

backdrop images and individual project backup settings can all be made here. For detailed documentation on the various tabs within this panel, refer to Project Settings (Appendix C).

Show Project Targets ⇧⌘T Toggles visibility of the floating project targets panel (subsection 20.1.1), for tracking a few simple metrics in real-time as you type and edit that can be left open while you work. These progress bars themselves can also be found in the toolbar's Quick Search tool (section 11.5) by default.

Statistics... ⇧⌥⌘S Opens a window displaying text statistics for the current project, including the word, character and page counts for the draft, the current binder selection or when called from a text editor, statistics for that section of text you are editing. Refer to Statistics (subsection 20.1.3) for more information.

Writing History... As you work in your project from one day to the next, Scrivener will record your writing progress in the background and keep a number of statistics available to you within this panel. You can also export the raw data to a CVS file, where you can then take the numbers into a spreadsheet for further visualisation or analysis. Read more in Writing History (subsection 20.1.4).

Show Project Keywords ⇧⌘K Displays or closes the project keywords panel (section 10.4.2); for managing all of the keywords in use by your project or assigning them to items.

Show Project Bookmarks ⇧⌘B This command has a few different results based upon the context of its usage. Refer to Project and Document Bookmarks (section 10.3) for full documentation on this feature.

— *Project window is active*: if not already shown, a list of bookmarks in a floating window will be opened, where they can be managed or navigated to. If it is shown but you are working in the project window, then this command brings the panel to focus. You can also open this panel by clicking on the Bookmarks button in the toolbar and then "tearing" the panel off by clicking and dragging from any of its borders.

— *Bookmark list is active*: when this command is invoked while the bookmark list is open *and* active it will have the same effect as clicking the expansion button in the upper-right corner (marked (a) in Figure 10.9): converting the tool into a Quick Reference panel so you can quickly review the contents of your bookmarks.

— *Quick Reference panel is active*: toggles whether or not the bookmarks sidebar is visible on that window. This is the same as clicking the bookmark icon in the window's footer bar, marked (b) in the figure.

— Hold down the Option key when selecting this menu command, or when using the shortcut, to go straight to a Quick Reference panel with the bookmarks sidebar open. If you're a veteran user of Scrivener looking for the old Project Notes window, this shortcut is the closest thing to it.

Empty Trash... Permanently discards the current contents of the project Trash folder. You will be warned before this is done, as once contents have been deleted, there is no way to undo that action. This can also be done by right-clicking on the trash folder itself in the binder.

If you instead wish to only delete some items from the trash, select them from within the trash folder and use the Edit ▸ Delete menu command.

A.9 Documents Menu

Contains all commands relating directly to existing documents, such as duplicating and splitting, moving and copying them, converting between types, bookmarking, setting icons, cleaning up formatting, titles and synopsis, managing snapshots of text milestones and so forth.

Snapshots ▸ This submenu provides commands to manage a document's snapshots, make new ones, and commit large-scale snapshot actions on selected items. There are some circumstances where these shortcuts will be unavailable. Generally this happens when the selection includes items that are not text items, if the text of the item(s) are empty or no changes have been made to them since the last snapshot. For more information on how to use snapshots, see Using Snapshots (section 15.8).

Snapshots ▸ Take Snapshots (of Selected Documents) ⌘5 When viewing any text or folder document, this command will take a snapshot of the current text and store it for later use. When selecting items from a list of any sort, the title of this item will change to "Take Snapshots of Selected Documents", and all selected documents will have a snapshot taken of them.

> **How to snapshot an entire folder**
>
> Selections must be precise, but you can easily select a folder and all of its subdocuments with a single command, Edit ▸ Select ▸ Select with Subdocuments (⌥⌘A), from an outliner or binder view, and then take the snapshots.

Snapshots ▸ Take Titled Snapshots (of Selected Documents) ⇧⌘5 You will be asked to provide a name for the snapshot before it is taken. This name will be displayed in the snapshot list. It is always possible to adjust the names of snapshots after they have been taken.

Snapshots ▸ Show Snapshots Switches to the snapshots tab in the inspector, revealing the inspector if necessary. This command is functionally identical to Navigate ▸ Inspect ▸ Snapshots.

Snapshots ▸ Show Changes ▸ When a snapshot is selected from the snapshot list, it is possible to view the changes between that snapshot and the current document. If two snapshots are selected, these two snapshots will be compared between each other, using the oldest snapshot as the original.

Comparisons can also be performed in the main editor or copyholders. Refer to Comparing Changes in the Editors (subsection 15.8.4) for more information.

Show Changes ▸ Compare Toggles visibility of the change tracking mode.

Show Changes ▸ Comparison Granularity ▸ These options can be toggled on and off individually to determine how closely changes will be tracked, with the finest level selected being the used method. "By Paragraph" will only mark changes at the paragraph level; "By Clause" only marks changes made at the phrase level; "By Word" is the most detailed level of change tracking available. Use these settings to fine-tune the results if what you are getting is too precise or vague to be useful.

Show Changes ▸ Next Change ^⌘] Scrolls the view to the next available difference in texts. This and the following command can also be used from the copyholder or main editor if they are viewing a snapshot in comparison mode.

Show Changes ▸ Previous Change ^⌘[Scrolls to the view to the previous available difference in text.

. . . . . .

Snapshots ▸ Roll Back Rolls back to the selected snapshot, replacing the text in the main editor. The option to snapshot the *current* editor text, before reverting the text, will be given in a dialogue box.

Snapshots ▸ Delete Permanently removes the selected snapshot from the disk. You will be asked to confirm this action as it cannot be undone. You may also delete snapshots with the backspace key, or by clicking the − button in the Snapshots Inspector.

Snapshots ▸ Show Snapshots Manager Opens the Snapshots Manager (subsection 15.8.5) window, where all snapshots in the project can be browsed and managed from a convenient central location.

. . . . . .

Duplicate ▸ There are two different methods you can use to duplicate the selected items.

Duplicate ▸ with Subdocuments and Unique Title ⌘D The entire item (including all of its children if it has any) will be duplicated, and it will be provided with a unique name automatically. If the current title ends in a hyphen + numeral, Scrivener will use an incremented number in the copy. For example is the document you are duplicating is called "scene-5", duplicating it would result in a file called "scene-6". It will otherwise add "-1" to the end of the duplicated item, starting a new sequence if further duplications are made. This command is also available from the binder contextual menu.

Duplicate ▸ without Subdocuments This is most useful when you only want to duplicate the container of an item by itself, such as a chapter folder, but not all of the scenes within it. In this case, a unique name will *not* be provided.

Split ▸ Provides a couple of methods for splitting the current document in two. In both cases, everything below the current cursor position will be moved into a new document below the current one. Everything above the cursor will remain unchanged. This action cannot be undone (except via the Merge command, below). Refer to Splitting and Merging Documents (section 15.4) for more information.

Split ▸ at Selection ⌘K Split the new document off from the current one, using all text after the current caret position (this can even split a paragraph in two). In cases where text has been selected, the caret position will be considered as the start of the selection range.

Split ▸ with Selection as Title ⌥⌘K In all details this command works identically as the above, but in this case the currently selected text will be used to title the new document that is created, rather than leaving the title empty.

Merge ⇧⌘M The opposite of splitting, merge will take two or more selected files or folders and concatenate them all together into a single document, using the top-most document in the selection as the "template" for any of the merged document's metadata that cannot be otherwise combined, such as labels and status. This action cannot be undone (except via the split command). In those fields where combination makes sense, such as notes and keywords, all of the documents will be used to create a combined metadata result.

The separators inserted between the main text of documents that are merged can be configured in the General: Separators preference pane (subsection B.2.9).

New Folder From Selection ⌥⌘G Takes one or more selected items and groups them into a new folder. You will be given the option to name that folder after invoking the command. When used from a selection of index cards on the corkboard, the selected cards will "disappear" from the current corkboard into the newly created folder card. This command is also available from the binder contextual menu.

Ungroup ⌥⌘U The opposite of grouping, ungroup will move the contents of the selected container up one level so that they become siblings of that container. This action will not change or move the original container, allowing you to move the items back in, if it was made in error. This command is also available from the binder contextual menu.

Move To ▸ Provides a binder item selection submenu, used to select a target item to nest the currently selected items beneath. Since any document can be selected in this menu, the action may result in the selected item becoming a file group.

The upper portion of this menu, labelled "Favorites", will track the most commonly and recently used targets in this project. When a new item is used, it will be given priority display in this area, but will fall off more quickly than the other more frequently used targets, if it is seldom used again. The most popular targets will be listed at the top. This command is also available from the binder contextual menu.

Move To "X" Again ⌥⌘T This convenience command will appear if during the current session you have moved an item using the "Move To" submenu above it. It will use the last location you used with this command, making it easy to file additional documents to the same location. This command is also available from the binder contextual menu.

Copy To ▸ Operating in a very similar fashion to the Documents ▸ Move To ▸ submenu, the selected documents (and all child items beneath them) will be *copied* to the target location. This command is also available from the binder contextual menu.

If you would prefer to use drag and drop with the mouse to perform this action, you will need to enable that behaviour in the Behaviors: Dragging & Dropping preference pane (subsection B.4.4). You can then hold down the Option key while dragging to create copies of the selected items in the dropped location.

Add to Collection ▸ Displays a list of all standard collections (subsection 10.2.2) in the project.[6] The currently selected items will all be added to the collec-

[6] It is not possible to add items manually to search collections—they would need to be altered to match the search criteria to be included.

tion you choose from this submenu. This command is also available from the binder contextual menu.

At the top of this menu (or as a sole entry if there are no standard collections yet in the project) will be a "New Collection" command. When used, the collection tab list will be revealed above the binder, and you will be able to type in a name for the new collection in that list. This can also be done by clicking the + button in the collection tab list header bar when it is open.

Add to|Remove from Project Bookmarks The selected items will be appended to the project bookmarks (section 10.3) list if they are not already listed within it. If the entire selection is already listed then the "Add to Project Bookmarks" command will be disabled. Use the opposing command, "Remove from Project Bookmarks" to remove all selected items from the list.

When the selection is mixed, containing some items that are bookmarked and some that are not, these commands will still be made available. This way you needn't know for sure if every time you selected is actually on the list or not. This command is also available from the binder contextual menu.

Convert ▸ This submenu provides tools for converting the document's type, scriptwriting mode, or text formatting.

Convert ▸ Text to Default Formatting... Resets the text of the selected files and folders (or the text in the active editor session, including multiple documents in Scrivenings) to the default formatting used for new files, as established either globally in the Editing: Formatting preference pane (subsection B.3.2), or on a per-project basis in project settings (section C.5). You may also wish to read more about Resetting Formatting (subsection 15.5.5) in general.

Convert ▸ Script Format... Useful when you need to switch from one scripting standard to another. Note that if all you wish to do is change the current document writing mode, you should use Format ▸ Scriptwriting ▸ Script Mode (⌘8) to do so. This function is strictly for converting between pre-existing scripts to bring them up to spec. For more information on scriptwriting, see Scriptwriting (chapter 19).

Convert ▸ to File|Folder Converts the selected items to folders or files respectively. Note that these two document types are very similar in Scrivener, so this tool makes it easy to change your mind later about whether or not something should be a folder or a file. If the selection contains a mix of types, or items that are neither files nor folders, then the command will be disabled. This command is also available from the binder contextual menu.

Convert ▸ Web and PDF files to Text Activates when viewing a WebArchive or PDF item. Will scour the file for its text content and create a rich text file from it that is editable and considered a text document from that point onward. Note this cannot be undone. Create a duplicate of the original beforehand if you wish to retain a copy with its full layout.

. . . . . .

Auto-Fill ▸ These commands can be useful for managing the titles and synopses of files in conjunction with text found or selected in the main editor.

Auto-Fill ▸ Set Selected Text as Title ⇧⌥⌘T When text has been selected in the main editor the title for the current document will be set to that selection. If you are splitting up a longer document into a format more useful within Scrivener, consider using the Documents ▸ Split ▸ with Selection as Title (⌥⌘K) menu command instead.

Auto-Fill ▸ Clear Titles Removes the titles from all selected documents, leaving them blank. This has the effect of causing them to print the first few words from either the synopsis or main text content (in that order of precedence) in various contexts such as the binder sidebar.

Auto-Fill ▸ Set Synopsis from Main Text|Selection ⇧⌥⌘I This command has two modes of operation:

- *Set Synopses from Main Text*: when the active selection is a quantity of files in a group view, such as the corkboard or binder, each item will have its synopsis updated to use the first few lines of text in that folder or file (it will be left blank if there is no text). This command also works if a text editor is active, and will in that case be considered a selection of one file. In Scrivenings mode, the document the cursor is within will be affected.
- *Set Synopsis from Selection*: if text has been selected within the main editor, then this alternate menu command will appear. The synopsis will be set to the selected text alone rather than using the first few lines from the top of the section. In Scrivenings mode, if the selection crosses the boundary of sections, the file the selection ends within will be affected by the command, but the entire selection will be used for the synopsis, even though it may come from several files.

Auto-Fill ▸ Send Synopsis to Main Text The contents of the selected items' synopses fields will be appended to the end of the main text in each item, respectively. Useful if you've done a little pre-composition in Outliner or Corkboard mode, and now wish to push those ideas into the manuscript text. This command will be unavailable if the selection doesn't include any items with synopses.

Change Alias Source... If the selected binder item is an alias to some other file on your disk, this command will allow you to point it to a different file. This will most often be useful if the original file has gone missing and has been restored.

Replace Media File... When anything other than a native folder or file has been selected in the binder, this command allows you to swap out the stored data with another file on your hard drive. A practical example where this could be useful is if you have a cover image for your book in the binder and you receive an updated copy from the artist. The new cover would be imported, replacing the old image within the project. In all other regards, such as links and preferences pointing to this image, the item itself will be unchanged.

Default Template for Subdocuments ▸ Select any existing group in the binder and use this command to make it so that new files created within that group will use the selected document template instead of the stock vanilla file type. To revert its behaviour to default, select the "Text" choice at the top of this submenu. In projects that lacking any templates, this menu will simply offer "Text" as the sole choice. Refer to Default Subdocument Template (subsection 7.5.2) for further information.

Change Icon ▸ This submenu displays all built-in and any custom icons added to the project or installed on your machine. Select items in any group view and then choose an icon from this menu to change their icons. When called from the editor or copyholder, the active document will have its icon changed. This command is also available from the binder contextual menu.

Change Icon ▸ Reset Icon to Default Select this option to remove a custom icon from the selected documents, or the active document from the current editor or copyholder.

Change Icon ▸ Icon from Text... Using this tool, you can take any text and turn it into an icon. This will work best with single characters, such as Unicode symbols or Emoji. Refer to Icons from Other Sources (section 7.4.1). This command will automatically bring up the Mac's Emoji & Symbols utility. If you need to open it again, you can click the "smiley face" button (Figure A.4).

Change Icon ▸ Manage Custom Icons... Opens the Manage Icons dialogue, where you can add or remove your own custom icons. You can manage both those icons loaded into the project as well as those installed on the system from here. For more information on managing and creating icons, read Custom Icons (section 7.4).

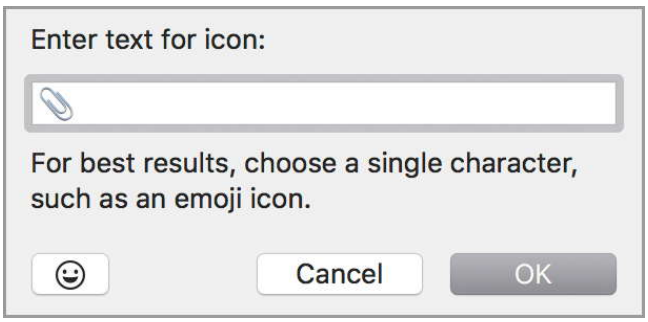


Figure A.4: Adding the "Paperclip" Emoji as an icon.

Change Icon ▸ Text Based Icons ▸ This submenu will list any text-based (such as Emoji) icons that have been found within the project, or that have been used in the current session. As you add or remove such icons from the project, this menu will list it so you can conveniently add them to new items.

. . . . . .

Move to Trash ⌘Delete Moves the selected binder items to the project's Trash folder. Scrivener works in the same fashion that the Finder does. Periodically, you can review the contents of this and empty the trash to clear up space, using the Projects/Empty Trash... command. This command is also available from the binder contextual menu.

A.10 Format Menu

The Format menu contains all commands that deal with formatting the text of an individual document. It contains all of the usual commands you would expect to find for changing the font, setting line spacing, creating tables and lists and controlling the ruler, along with commands for highlighting and setting the current text colour. Additional tools related to formatting will be found here, such as revision markings, scriptwriting tools and style commands.

Font ▸ All commands relating to ad hoc character formatting will be found in this section of the menu. This includes basic settings like bold and italic, adjusting the size, kerning and other typographic features. Use these to set up new styles or modify text on the fly. Many of these tools can also be found on the Format Bar (subsection 15.5.2).

Font ▸ Show Fonts ⌘T Access the standard system font palette. Use this to change the font settings you type in from that point forward, or alter the existing font of a selected range of text.

Bold/Italic (Cmd-B, Cmd-I) These basic formatting commands in the same fashion by either modifying the cursor attributes if there is no selection, or toggling the selected text between the various styles. Note that not all fonts support bold and italics. If it appears that one of these is not working, check your font for the proper typefaces.

Font ▸ Underline ▸ ⌘U This submenu provides several underlining styles. The default command with the shortcut is what most people will wish to use when underscoring text. Use the "By Word" option to omit underscoring across whitespace characters.

Font ▸ Strikethrough ⇧⌘_ Crosses out the text you have selected. When used in conjunction with marking revisions (section 18.6), the current revision level colour will be used for the strikeout. Strikethrough can also be applied to text in PDF documents.

The setting can be more than merely cosmetic. Struck-through text can be easily deleted at a later time via the Edit ▸ Text Tidying ▸ Delete Struck-Through Text menu command, and can be dynamically stricken from the output with the **Delete struck-through text** setting in the compile overview screen, under general options (subsection 23.4.3).

Font ▸ Outline Changes the style of the font to outlined instead of filled. This will be more useful with larger, heading size fonts with heavier weights.

Font ▸ Bigger ⌘+ Increases the size of the font by one point.

Font ▸ Smaller ⌘- Decreases the size of the font by one point.

Font ▸ Character Spacing ▸ Adjust the kerning of the selected text, or between the two characters around the cursor. The "Use Default" selection allows the font to choose the kerning based on the designer's metrics (this will nearly always be the best choice), while "Use None" disables the font's built-in kerning. It would be advantageous to set keyboard shortcuts to the "Tighten" and "Loosen" commands if you intend to use these frequently.

Font ▸ Ligature ▸ If the font family supports typographic ligatures, they can be enabled or disabled here. Most fonts default to using standard ligatures (ff, fl, fi and so on). Some include rare ligatures, which may be enabled by the "All" setting. It will usually be better to use the Format ▸ Show Fonts panel, click on the ⚙ button and open the "Typography" panel, for more precise and extensive options.

Font ▸ Baseline ▸ Adjust the baseline height of the text, most commonly referred to as superscripting and subscripting. In addition to jumping straight to those presets with respective commands, you can also incrementally raise and lower the baseline. "Use Default" will reset the baseline to the normal.

Font ▸ Character Shape ▸ If the font family supports typographic shape alternates, they can be adjusted here, such as old style numerals and traditional forms.

Font ▸ Copy Font ⌥⌘C Copies the font settings from the current cursor position. We use this term liberally, to include all forms of inline formatting, such as bold, underscore, overstrike, text colour, highlights and of course all of the font settings found within this menu.

This command does not overwrite the current standard clipboard; it uses a special clipboard reserved for these settings alone.

Font ▸ Paste Font ⌥⌘V Use this command to apply format settings, copied using the above command, to other selections of text.

. . . . . .

Paragraph ▸ Displays the standard macOS Text submenu, which contains useful tools for controlling the paragraph-level formatting of your text. Paragraph alignment and spacing can also be controlled with the the Format Bar (subsection 15.5.2).

Paragraph ▸ Align Left ⌘{ Sets alignment on the selected paragraphs to flush left, or ragged right.

Paragraph ▸ Center ⌘| Centres the alignment of the current paragraph.

Paragraph ▸ Justify ⇧⌥⌘ Sets alignment on the selected paragraphs to fully justified, or flush left and right.

Paragraph ▸ Align Right ⌘} Sets alignment on the selected paragraphs to flush right, or ragged left.

Paragraph ▸ Tabs and Indents... Brings up a panel that can be used to manually insert tabs and set paragraph indenting using precise measurements, rather than using a visual ruler. For more information on using this panel, refer to The Tabs and Indents Tool (section 15.5.1).

Paragraph ▸ Increase/Decrease Indents ▸ The commands in this submenu all operate on the current paragraph or selection of paragraphs, as the case may be, and is provided primarily so that you can supply you own custom keyboard shortcuts to the most frequently used commands. They independently adjust the three different indent aspects of a paragraph (the fourth, hanging indents, is a shorthand for adjusting the two left-side indent controls in an inverse fashion than is typical, causing the first line to be further left than the body of the paragraph below it) that can alternatively be adjusted using the sliders on the ruler (⌘R).

The amount of increment used depends on the unit of measurement that was set when the project was first loaded. With the exception for first line

indents, all methods will adjust indentation in increments of 0.5cm; 0.25in; 1.5 picas and 18 points. First line indents will also use available tab stops as well as the fixed increment.

Indents ▸ Indents

Increase/Decrease Indents^⌘→ &← As a block indent, all left-side indent settings are moved left or right uniformly. If it is a hanging indent, for example, it will remain a hanging indent, only with both indent settings moved one way or the other.

This command is synonymous with the Edit ▸ Move ▸ Move Left and //Move Right commands, and that is where the keyboard shortcuts for it will be found.

Indents ▸ First Line Indent ^⌥⌘→

Increase/Decrease First Line Indent^⌥⌘→ &← Only the first line the paragraph will have its indent increased or decreased. When the rest of the paragraph is indented it is possible to decrease the first line indent beyond the rest of the paragraph, forming a hanging indent (though in general it will be easier to use the next command to create these).

Indents ▸ Hanging Indent The main body of the paragraph will have its indent level adjusted while leaving the first line indent alone. If the paragraph is already indented then it may be possible to decrease the indent level beyond the first line, resulting in a normal looking paragraph.

Indents ▸ Right Indent The right-hand indent of the paragraph will be increased or decreased. It is important to be aware of the fact that a right indent is measured from the left of the page, not the right. In other words if you move the right indent inward 2cm, the indent setting might be more like 16.5cm, which could have implications if the margins or paper size change.

. . . . . .

Paragraph ▸ Remove All Tab Stops A convenience tool to delete all tab stops in the current paragraph or selection.

Paragraph ▸ Line and Paragraph Spacing... Brings up a handy tool for setting paragraph and line spacing attributes in the selected or current paragraphs. You can also access this tool from the Format Bar (subsection 15.5.2).

Paragraph ▸ Writing Direction Toggle between Left-to-Right and Right-to-Left writing styles. This will usually be automatically done for you when changing keyboard input modes.

Paragraph ▸ Keep with Next Inserts an invisible character which will keep attempt to keep this paragraph glued with the next paragraph, so that they will not become separated by a soft page break.

Paragraph ▸ HTML Header Level This will only be of interest in conjunction with styles in eBook and Web publishing, as well as the Markdown-based formats.[7] They will set the heading level for the selected text in these outputs when compiling, and of course will be saved into any styles you create from that text. Unless you intend to use headings directly in your text, you may not even have use for these commands until you are actually setting up styles in the Styles compile format pane itself (section 24.5).

Use of these commands will have no visual effect on the text. They only determine what HTML level the text will be set to.

Paragraph ▸ Copy Paragraph Attributes ^⌘C Copies the ruler and paragraph format settings from the current paragraph. These settings can be pasted into any rich text view, even compile settings panels that take formatted text.

This command does not overwrite the current standard clipboard; it uses a special clipboard reserved for these settings alone.

Paragraph ▸ Paste Paragraph Attributes ^⌘V Pastes any stored ruler and paragraph settings to the selected or current paragraphs.

. . . . . .

Style ▸ Styles are a way of assigning meaning to your text, and optionally formatting or other visual cues. Once text has been assigned to a style it will stay in sync with any future updates made to that style's settings. Read more about Styles and Stylesheets (section 15.6).

Style ▸ New Style From Selection... Starts the creation process for a new style, using the formatting of the selected text, or from the paragraph the cursor is located within. Read more on Creating New Styles (section 15.6.3).

Style ▸ Redefine Style From Selection ▸ Takes the formatting from the currently selected text or paragraph and updates the designated style you choose from this submenu. Read more on how to Redefine a Style (section 15.6.3).

Style ▸ Delete Style ▸ Deletes the style you choose from this submenu. The formatting applied to text that is assigned to this style will be left alone, but the style assignment itself will be stripped from all text throughout the project.

---

[7] For the latter case, one must have the **Convert rich text to MultiMarkdown** option enabled, in the General Options tab of the compile overview screen (subsection 23.4.3).

Style ▸ Show|Hide Styles Panel ^S Toggles the display of the floating styles panel. If you use styles frequently, you may wish to keep a window like this open off to the side so you can see at a glance which styles are applied to the text you are working with, and apply styles to text with a single click. You can also easily modify and manage your styles from within this pane. Refer to The Styles Panel (section 15.6.2) for further documentation on this feature.

Style ▸ Pop Up Styles Menu ⇧⌘Y Reveals the "Apply Style" menu directly on your screen without having to go to a separate menu or format bar to access them. It is not accessible when the document is in script writing mode, as the shortcut will be reassigned to the scripting elements menu.

Style ▸ Set Default Formatting... Loads the Project Settings: Formatting pane (section C.5), where you set what un-styled text should look like via the **Main text formatting** area.

Style ▸ Import Styles... Brings up a file selection dialogue box. Use this to select the Scrivener project from which you want to copy its stylesheet. Refer to Copying Stylesheets Between Projects (subsection 15.6.5) for further detail on how to manage merging when importing styles.

Style ▸ No Style ⌥⌘0 Depending upon the context, this command has different effects. When used upon selected text the styling from that selection will be stripped out, returning it to either the look of the paragraph style beneath it, or to standard un-styled text. When used without a selection within a paragraph that has a paragraph style applied to it, the paragraph style will be simply removed from the entire line.

When used while typing, the effect is to cease typing in the style that was being used up until that point. If used at the conclusion of a range of text in a character style, you will return to whatever paragraph style was in use underneath that, or to no style at all, depending on which is applicable.

The remainder of this menu will list the paragraph and character styles available in this project. Use of them will apply the chosen style to the selected text, or alter the formatting of the cursor.

. . . . . .

Table ▸ Use the "Table..." command to insert a starter table (more easily access with the Insert ▸ Table... command). The remaining commands in this menu are documented in the tables contextual menu (section 15.3.2) section, and of course are also available when right-clicking directly inside of a table in the editor.

Lists ▸ Supplies basic list styles, and few advanced controls for bullet and enumerated list management, from the "Custom List...". The list creation aspects of this menu are also available when clicking on the list button in the

Format Bar (subsection 15.5.2). You can cycle through the various list styles from any existing list in the editor with the following commands:

— *Next List Style* (⌥⌘→): selects the next list style from the currently selected style.

— *Previous List Style* (⌥⌘←): selects the previous list style from the currently selected style.

Scriptwriting ▸ Although Scrivener is not a dedicated script writing application such as Final Draft, Fade In Pro or WriterDuet, the scriptwriting submenu makes it relatively straightforward to create first drafts which can be later fine-tuned in these and similar applications. For full documentation on this feature, refer to Scriptwriting (chapter 19).

Scriptwriting ▸ Script Mode (Format) ⌘8 Toggles the current scripting mode on or off for the selected documents in any group view or the binder sidebar, and for the current document in the main editor—or all documents within the current Scrivenings session. Toggling script mode on and off only changes how text input is formatted while writing, not any aspects of text that has already been written. You can thus switch to normal prose and then back again, making this a great shortcut to learn when writing treatments.

Scriptwriting ▸ Script Settings... Brings up the script settings window, where all aspects of the project's script settings can be modified, and new scripts and elements can be created from scratch. Read more about this tool in Creating Your Own Script Formats (section 19.7).

Scriptwriting ▸ Re-capitalize Script Any elements in the current script mode that are set to be capitalised will be converted to all-cap text if they are not already. This will mainly only be of use when importing text that isn't already capitalised, as this is done automatically for you when using and writing with elements normally.

Scriptwriting ▸ Change Element To Converts the selected text to a different scripting element. Note that if the previous element set the text to all-caps, and the element you are changing to should not be capitalised, you will need to fix the text yourself, making use of the tools in the Edit ▸ Transformations ▸ submenu.

Scriptwriting ▸ Show Script Elements Menu ⇧⌘Y Reveals the script element selection menu, also available by clicking in the lower left corner of the editor footer bar. Single-letter shortcut keys can then be used to select an individual element, if the script settings have been configured to use them. For example in our built-in screenplay script, the "C" key will switch the line to Character mode.

This command will only be available when script writing mode is enabled in the active editor. The shortcut will otherwise bring up the styles menu.

The remainder of this menu will list all of the scripting styles installed on your computer, along with a number of built-in styles that ship with Scrivener. Each project can only use one script at a time. Thus you will be asked to convert existing script documents upon selection. If you haven't written anything yet you can ignore that dialogue, but if you have substantial writings already committed you will want to pay close attention to the element conversions table and other options provided in this pane, to avoid damaging the formatting of existing text. Refer to

. . . . . .

Color... ⇧⌘C Toggles the visibility of the colour selection palette. Note that Scrivener uses this palette for all cases where a custom colour can be set by you, but when invoked with a text editor active, the effect will be to colour the selected text or the cursor attributes.

Highlight ▸ The highlight menu lets you place a background highlight behind the selected text, much like using a highlighter marker on paper. In addition to the five provided presets, this menu also displays any favourite colours that have been set on your system. For more information on how to use highlights, see Text Colour and Highlights (section 18.5).

Highlight ▸ Highlight Text ⇧⌘H Highlights the selected text, or alters the cursor's attributes, using whatever colour you used last. If the text is already highlighted and you are using the same highlight colour then the effect will be to remove highlighting from that text. Otherwise the text highlight will be switched to the new colour. Highlighting (in basic yellow) can also be applied to text in PDF files.

Highlight ▸ Remove Color Strips all highlighting that has been applied to the selected text.

Highlight ▸ Show Colors... This choice will appear at the very bottom of the list of colours. Use this to select a custom colour using the standard colour picker. Note that if you right-click the highlight tool in the format bar, you will be provided with a number of handy presets (subsection 15.5.2).

. . . . . .

Revision Mode ▸ Revision modes force anything you type to be displayed in a provided colour until you disable the revision mode (it also impacts strikeouts). Select one of the five revision levels in this menu to enable the mode, and again to toggle it off. For more information on using this, see Marking Revisions (section 18.6).

Revision Mode ▸ Mark Revised When a text selection has been made *and* a revision level in use, this will let you mark text ranges with the current revision colour. This can also be done by clicking on the text colour tool in the Format Bar, as it it will be reset to the current revision level while in use.

Revision Mode ▸ Remove Current Revision Color This command is only available when one of the revision levels is active. It will strip the current level from the active editing session, leaving the rest alone.

Revision Mode ▸ Remove (All) Revisions Indiscriminately removes all five revision level colours from the active editing session. When text has been selected, the command will read "Remove Revisions" and only strip markings from within the selection.

. . . . . .

Copy Formatting ^⌥⌘C Combining both the effects of Format ▸ Font ▸ Copy Font and .../Paragraph ▸ Copy Paragraph Attributes, all formatting from the selected text will be placed on a special clipboard, for pasting to other text.

Paste Formatting ^⌥⌘V Pastes all formatting from the formatting clipboard. This will include anything you've copied using the Format ▸ Font ▸ Copy Font, .../Paragraph ▸ Copy Paragraph Attributes or Format ▸ Copy Formatting commands. In essence they all share the same clipboard, but ordinarily only use portions of it depending on which command you use to paste.

Preserve Formatting This is a legacy feature from older versions of Scrivener. It will protect all formatting found within the range of text you apply it to, when compiling through formats that would otherwise change that text. Most of this capability is now provided through the stylesheet system, but there are still a few niche special-purpose uses for this command. Read more about them in Preserve Formatting (subsection 15.5.6).

Make Formatting Default This will make the currently-selected formatting in the editor the default formatting for all new documents, without otherwise going through and changing these settings in the Editing: Formatting preference tab (subsection B.3.2) (or the Project Settings: Formatting pane (section C.5) if the project is overriding the application defaults). Upon use of the command you will be asked whether you want the attributes of the selected text to be used in your global application preferences or the project settings specifically. In the latter case, if you have not already enabled the option to override formatting in this project it will be set for you.

As always, this command only impacts text typed into newly created document. It will *not* blow away all of the formatting in existing files. If you wish to do that yourself, you can with the Documents ▸ Convert ▸ Text to Default Formatting... menu command (subsection 15.5.5).

A.11 Window Menu

The Window menu contains some standard commands for controlling windows, tabs and panels. It concludes with a list of all open projects.

Minimize ⌘M Minimises the window to the Dock. Hold down the Option key to minimise all.

Zoom ^⌘- Zooms the window in or out (the same as the green button in the top-left of the window). Scrivener will try to zoom intelligently to best fit the contents of the window, taking into account your preferred editor width.

Zoom to Fit Screen ^⌘= Zooms the window so that it fully fills the screen, excluding the Dock and menu bar.

Show Previous Tab ⇧^Tab Scrivener supports macOS tabbed window features. Projects can be combined into the same window as tabs, and Quick Reference panels can be combined into their own window. The following three commands will be disabled unless you have merged windows and have tabs in that active view.

This command selects the previous (left) tab in the current window.

Show Next Tab Selects the next (right) tab in the current window.

Move Tab to New Window Separate the current tab back out to its own window.

Merge All Windows Merges all open windows of a like kind (project vs Quick Reference) into a single window. If you wish to only merge select project windows together, first open the tab bar for each project with the View ▸ Show Tab Bar menu command, and then drag and drop the tab from one project into the other's tab bar.

Layouts ▸ This submenu is populated by the saved layouts you've created and a few built-in layouts to get your started. This menu provides an easy mechanism for rapidly converting your project window to different workflows, depending on what you are currently doing. Read more about Saved Layouts (section 12.3).

Layouts ▸ Manage Layouts ⇧⌘) Opens or activates the Layouts panel, where you can save, update, export/import or recall project window view settings.

. . . . . .

Float Window Toggles whether or not the current project window should "float" above all other windows. This will make it visible at all times, even when switching to other programs.

Float Quick Reference Panels ^⌘Q Toggles whether or not Quick Reference panels should float over all project windows for this specific project. This setting impacts all Quick Reference panels created from the current project. Panels from another open project will behave in accordance with that project's settings.

Show|Hide Scratchpad ⇧⌘Return Displays or closes the floating Scratchpad Panel (section 9.4), useful for collecting information while using other applications. You can assign a keyboard shortcut to this command in the General: Scratchpad preference pane (subsection B.2.6).

Bring All to Front Brings all Scrivener windows to the front. Hold down the Option key to change to Arrange in Front.

The next portion of the menu only appears if the currently active project has open Quick Reference panels. They will each be listed in this section by name.

Close All Close all open Quick Reference panels for the current project.

Closed Quick Reference Panels During a single session, any Quick Reference panels that have been closed will be saved into this submenu, giving you quick access to them if you need to re-open them.

The rest of this menu will be populated by all of the windows (excluding panels, like the Project Keywords window) that are currently open in Scrivener. Selecting from this list will bring that window to the front.

A.12 Help Menu

The Help menu provides access to the standard Mac's menu search utility, as well as useful tools and links for learning Scrivener or getting in touch with us if you require technical support.

Scrivener Manual A quick link to the PDF that you are likely reading this from. The version that ships with Scrivener will be kept as up to date as possible, but newer revisions might also be available on the web site's support page[8].

Interactive Tutorial If you have not yet gone through the tutorial (you should!) this menu command will create a project designed to walk you through the process of learning the basics of Scrivener. You will be given the option of where to save it, and from that point on you can load it like an ordinary project. If you have already created the tutorial project, you can use this menu command to quickly load it again, so long as it hasn't been deleted or moved.

[8] http://www.literatureandlatte.com/learn-and-support/user-guides

Video Tutorials A handy link to our web site's video page[9].

List of All Placeholders... There are many placeholders you can use with Scrivener, and this will provide an exhaustive list of them all in a floating window so you can copy and paste them into your project (or compile settings) as needed.

Support Jump to our web site's support page[10]. Here you can download the latest copy of the PDF in US Letter or A4 (as well as the project used to create them), find contact email addresses, links to our forums and wiki, after-sales support from our vendor, eSellerate, and more.

User Forums Jump to the official Scrivener forums where you can meet other authors around the world using Scrivener, share tips, report bugs, request tech support, or have a cup of latte in our off topic section.

Release Notes Jump to the official release notes[11] web page on the web site.

Literature & Latte Home Jump to our home page[12] which provides easy access to everything else we offer on our web site.

Scrivener Home Jump to the main Scrivener web page[13], where you will find useful download links for updates, case studies, links to share Scrivener with Twitter and Facebook, and more.

Keep Up to Date... Presents a form which you can use to submit your email address and name to subscribe to our newsletter. This is a low volume list that we use to send out important updates and news. Please take care to whitelist "litereatureandlatte.com" in your spam filter, prior to submitting this form, as you will be sent a confirmation email which must be responded to before you will be subscribed. You can also click on the Twitter and Facebook buttons to visit our official social pages.

Privacy Notice: Your email address will never be shared with third-parties or sold to marketing lists.

Purchase Scrivener... Jump to our web store, where you can purchase Scrivener through our secure vendor (eSellerate) and the other software or materials we have for sale.

---

[9] http://www.literatureandlatte.com/learn-and-support/video-tutorials

[10] http://www.literatureandlatte.com/learn-and-support

[11] http://www.literatureandlatte.com/scrivener/release-notes?os=macOS

[12] http://www.literatureandlatte.com/

[13] http://www.literatureandlatte.com/scrivener

Preferences

B

In This Section...

Like most applications, Scrivener installs in a "ready to use" state. The default preferences have all been carefully selected to present a cohesive and useful writing environment, and most of this manual will presume you are using them. We know that a creative working environment should bend to your working habits as much as possible, however, and so there are a great many ways to tweak the look and feel of the interface. This appendix will go over every setting and point you to any further discussions elsewhere in the manual if necessary.

Preferences take effect as soon as you change them. If your screen is large enough, you can leave the preferences open to the side and tweak things until you get them looking and acting the way you want.

Beyond global application preferences, each project can be configured with its own settings. These will be documented in Project Settings (Appendix C). Many project view settings are exclusively set using the menus, or with buttons located

within the project window itself.[1] These kinds of settings are rarely altered globally, but you can always create or modify your own project templates to achieve a preferred starting point for your new projects (section 5.4). Given how these sorts of settings are by nature scattered about contextually, they will be documented alongside the discussions on the features they pertain to, rather than in one central area.

Get me back to the defaults!

Feel free to experiment. Along the bottom of the preference window you'll find a button labelled Defaults. This will perform a "factory reset", returning all settings to their default states. This cannot be undone, but you will be warned after clicking the button and given a chance to cancel—you can also save your settings for later retrieval (subsection B.1.1). Most tutorials, books on Scrivener and even areas of this manual will assume default settings, so it can be a good idea to use them while you are following a guide. Many preferences can fundamentally change how the software works.

B.1 Preference Organisation and Management

For ease in finding documentation on the setting you are looking for, this appendix we will be organised in the same manner as the preferences panel itself. To get started, use the Scrivener ▸ Preferences... menu command (⌘,).

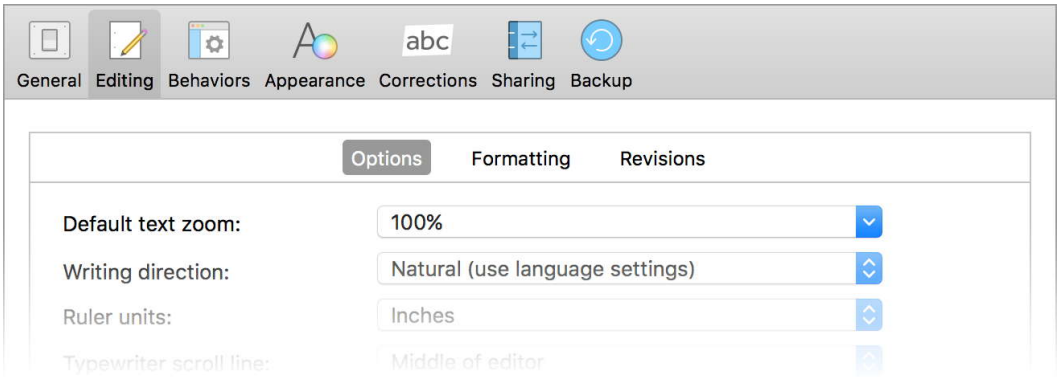


Figure B.1: The "Editing" category is divided into three sections as tabs.

Along the top of the window are the major preference categories. When we refer to a preference pane by name, it will be by one of these sections, which you

[1] A good example is whether or not label colours are displayed on index cards. By default they are, but you can turn them off for a project and it will remember that setting.

can access by clicking on the tab, such as the Editing preference pane (Figure B.1). This pane is broken down further into individual tabs—in this case the "Options" tab. We would refer to this as the Editing: Options pane, wherein one can change **Default text zoom** setting, used in most places where text can be edited.

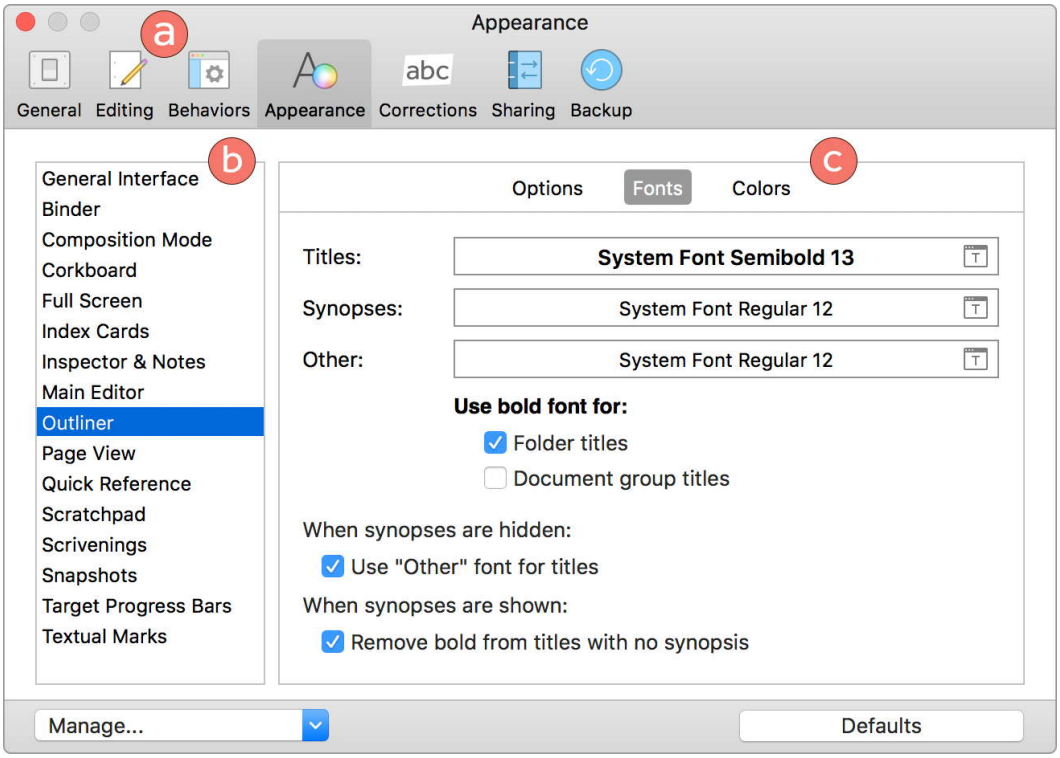


Figure B.2: The "Appearance" pane uses up to three levels of organisation.

In Figure B.2 we have a more complex layout, where the major section chosen along the top (a) is "Appearance", which has many subsections listed along the left (b). The contents of these subsections will be displayed to the right, and can in sometimes be broken down further into tabs (c). We would refer to this as the Appearance: Outliner: Fonts pane.

B.1.1 Preference Presets and Themes

Along the bottom left edge of the preference window you will find a Manage... button with a dropdown menu containing several options available for organising or backing up your settings, as well as access to any presets you might have saved. This can be particularly useful if you work on more than one computer, and in the case of themes, to switch up your working environments between favourite sets of colours and even fonts.

Load/Save Preferences as Preset Presets are a convenient way to store different preference sets. In cases where you need to use varied settings for different projects, this will make it easy to switch sets on the fly. Use the **Save**

Preferences as Preset... option to save all current preferences into a preset under the name you provide. Once saved, this preset will be available from the **Load Preferences from Preset** submenu.

If you wish to update an existing preset: make any changes you see fit, then use the save command, specifying the name of the preset you wish to overwrite. You will be asked for confirmation before the old preset file is overwritten.

Delete Preset The preference preset you choose from this submenu will be deleted from your support folder. This cannot be undone, so you will need to confirm your decision.

Load/Save Preferences from File... Save your preferences to an external file for either backup purposes, or to synchronise your settings between multiple computers. Preferences can be loaded, replacing all current settings.

Load/Save Theme Themes differ from preference presets in that they only save and load appearance settings, such as interface colours, corkboard and outliner appearance and fonts. The types of settings that will be saved into the theme can be chosen in the Save Theme dialogue. Note that for features like Scrivenings titles, saving them into a theme will only adjust the appearance of these features if they are used. Whether to show titles is a project-specific setting.

Themes can also be loaded from the main Scrivener ▸ Themes ▸ submenu.

Delete Theme The selected appearance theme will be deleted from your support folder. This cannot be undone, so you will need to confirm your decision.

Load/Save Theme from File... Load a theme from a file you have stored on your machine. This will not install the theme into the menus above, but rather simply apply its settings to your current preferences. Use the **Save Theme...** command if you wish to save them for future use.

Themes as files can be useful for synchronising settings between computers, sharing your theme settings with others, or loading theme settings you've downloaded from the Internet.

Return to chapter ⤴

B.2 General

General preferences govern the application's basic behaviour, its integration with other programs, designating global folders such as the scratchpad and how appending or merging text with existing documents should be performed.

Figure B.3: The General preference pane

B.2.1 Startup

Reopen projects that were open on quit Whenever Scrivener is launched, projects that were left open when you last quit will be automatically reopened, bringing you straight back to what you were last working on.

Hold down the Shift key while launching Scrivener to temporarily suppress this behaviour and open the software without any projects opened.

Show template chooser when there are no projects open Toggles the automatic display of the start panel, for the creation of new projects (subsection 5.1.1), when all project windows have been closed, or if the software loads without any projects to open

Reopen Quick Reference panels when opening projects Ordinarily, Quick Reference panels (section 12.6) are session based—when their related project is closed they will not be reopened the next time you load the project. With this preference enabled, all projects will remember any Quick Reference panels left open and reopen them automatically when the project is loaded.

Automatically check for updates If this is enabled the software will periodically check the website to see if there is a newer version available and, if so, will ask you if you want to update. Note that if you do not have this enabled, you can still check for updates yourself by using the Scrivener/Check for Updates... menu command. The dropdown menu beside this option governs how frequently Scrivener will check for new versions.

B.2.2 Saving

Auto-save after *n* second(s) of inactivity Scrivener automatically saves changes made to projects as you work, but so that it is not constantly saving—and to avoid any slowdown and interruption to your work that might cause—it will wait to save until you stop interacting with the program for more than two seconds (by default). Adjust this if the two second default conflicts with your natural rest-work cycle and causes halting when you try to resume typing.

If you increase the period significantly, be sure to use File ▸ Save regularly to keep your work saved. Scrivener will always auto-save when you close a project.

Take snapshots of changed text documents on manual save This option will cause the project to snapshot all text items (section 15.8), and only those, that have had their main text content altered during the session or since the last time you saved with the File ▸ Save (⌘S) command. Auto-save will never trigger this behaviour, nor will closing the project.

Snapshots that have been automatically generated in this way will have their name set to "Untitled (Save)" to differentiate them from any "Untitled" snapshots you take yourself manually. They can consequently be easily searched for and periodically purged from the project using the Snapshots Manager (section 15.8.5).

Show notifications when saving in composition mode Register an alert through Notification Center whenever you manually save the project in Composition mode, as there would otherwise be no indication of when save completes.

B.2.3 Author Information

This information will be used for filling in placeholders in project templates (for example, some templates generate cover sheets for you) and compile placeholders such as <$author>. Individual projects can also use specific pseudonyms or other details, set up in the Metadata Options area of the compile overview screen (section 13.5).

B.2.4 Services

Options for how Scrivener's global clipping Services should work. For more information on available services, read Scrivener Services (section 9.3). For the least amount of intervention, set both of these options to off.

Bring Scrivener to front when using Services Choose between active and passive clipping. When this is checked, clipping from other applications will be active in that it brings you back to your project after you've clipped some text from another application. When unchecked, it will work passively, keeping the project in the background and allowing you to continue work in other programs while you add material to the project.

Show title prompt when using clipping Services Normally, when you use a service that requires the creation of a new item in the binder, you will be asked what you wish to call it. When unchecked, this option will defer that task until later, when you are ready to think about names.

Include screenshot when using Twitter service in Project Targets Includes a screenshot of the Project Targets panel along with your tweet. It should be noted that this screenshot will include the working title of your project. If

you wish to obscure that information from the public, you should disable this option.

B.2.5 Language

Interface Language By default, Scrivener will attempt to select the appropriate interface language based upon your system settings. If this does not work, or you would like to use another interface language directly, you can use this setting to manually adjust the language. This setting has no impact on the production of information that is controlled by system localisation, such as date stamps and auto-number placeholders.

Use the "System Default" selection to return the interface to the macOS system default.

Linguistic focus uses Spanish-style dialogue When using the Linguistic Focus (subsection 20.2.2) feature to filter for direct speech, in addition to speech mark punctuation detection (as set in the System Preferences: Keyboard: Text preference pane), Scrivener will also scan for lines that begin with an em-dash—a form of dialogue marking that is common in Spanish-speaking regions.

This setting will also look for text found within angle-bracket style speech marks, even if System Preferences is not set to use that punctuation style for typing. Even if you do not use em-dash style dialogue, this may be useful if you use both angle-bracket notation and whatever quotation is standard to your smart quote settings.

B.2.6 Scratchpad

Hot key Sets the global hot key for showing or hiding the Scratchpad Panel (section 9.4). Since this key will be accessible from any application on your Mac, you might need to change this setting if it is conflicting with another application you use.

To clear the shortcut entirely, click the – button.

Notes location Your scratch pad notes are stored on the disk using normal text files. This option lets you choose where those files will be automatically saved and loaded from. By default, they will be stored in your Documents folder.

If you select a synchronised folder, such as Dropbox or iCloud Drive you can effectively create a universal scratchpad that every machine you use Scrivener with can access and share, or even jot down notes from a mobile device.

Default format This setting impacts which file format to use for new notes created from within the scratchpad. For the plain-text format, you can supply

a custom file extension in the **ext** field. This can be useful for Markdown-based formats.

Files saved into the folder from external sources will be editable in the scratchpad window and remain in their original format. You can thus mix rich text with plain-text notes if you desire.

If you use Scrivener for both Windows and macOS, and intend to share your notes cross-platform, do not use the default RTFD option, as this format cannot be read on other platforms.

B.2.7 Templates

Shared templates folder Click the Choose... button to define the location on your disk where Scrivener will check for files to be used as globally accessible document templates. For more information on the usage of this feature, refer to Shared Templates on the Disk (subsection 7.5.3).

Click the Clear button to reset the folder, removing this feature from the software. The Open shared templates folder... button will load the folder using Finder.

B.2.8 Citations

Bibliography Manager Scrivener can integrate with several popular bibliography management programs. Click the Choose... button to select your preferred manager from the Applications folder. The Reset button will clear your choice.

General purpose usage

This merely links which application will launch with the Insert ▸ Bibliography/Citations... menu command.. If you do not use a citation manager, but frequently use another program along with Scrivener, you could use this feature to quickly launch or switch to that program.

B.2.9 Separators

Text Separators This table lets you adjust how Scrivener will combine individual texts when certain actions bring them together. In all cases, you have three choices of divider available:

1. *Single Return*: inserts a single paragraph break between selections. If two paragraphs appear together, there will be no whitespace between them, save for any paragraph spacing in use.

2. *Empty Line*: a full empty space will be inserted between selections. If you prefer or require working with double-spaced paragraphs, this is the option you will want to use.
3. *Custom*: lets you type in a custom separator. If you wish to insert carriage returns along with the custom separator, use Opt-Return to do so.

Separators are used in the following four cases:

1. *Merged documents*: determines how documents will be merged when using the Documents ▸ Merge command. Note that this setting only affects the main text. The inspector notes and synopses of merged documents will always be separated by a double newline.
2. *Append clippings service*: when using the global "Scrivener: Append to X" services, clippings will be separated from any existing text by your choice here.
3. *Append selection*: when selecting text and using the Edit ▸ Append Selection to Document command, the selection will be separated from the existing text by this option.
4. *Scratch pad notes*: when appending text to documents from the scratch pad, incoming text will be separated from existing text by this option.

B.2.10 Automatic Quit

Automatically quit after a period of inactivity If the software has been left alone for a determined amount of time it will be shut down normally, meaning all open projects will be closed and backed up (by default). This can be of use if you tend to leave the program running unintentionally, and wish to access these projects from another machine using a cloud service. Ordinarily if you accidentally leave a project open it would be unsafe to open them a second time. This feature ensures that after a period of time they will be closed, so long as the computer is still running during that period of time.

Quit after inactive for *n* minute(s) Set how long to wait before shutting down the software in minutes.

B.2.11 Warnings

Show internal error alerts When you first start using Scrivener, actions which have a destructive or unusual nature (such as importing in such a way that original formatting might be lost) will produce a warning dialogue. You can often choose to disable these as you see them from within the warnings themselves. The `Reset All Warnings` button will clear all of these dismissals and make them appear again the next time it is appropriate.

Return to chapter ⤴

B.3 Editing

Figure B.4: The Editing preference pane

The options for the main text editor control most aspects of how text is displayed, how the editors work, the default font and paragraph formatting for new documents, footnotes and other notes and finally the colours used in revision mode. For options pertaining to text input and typography, see the Corrections preference pane (section B.6). The Appearance preference pane will also contain numerous options for how the editor looks, and some aspects of how textual markings, such as how hyperlinks are formatted (section B.5).

B.3.1 Options

Default Text Zoom Dynamically scale the size of the text font in the editor. As is typical for word processors, zooming will not impact the underlying font settings. This setting impacts the main text editors, the inspector notes and bookmarks preview area, Quick Reference panels (and the inspector panels within it), footnotes & comments and copyholders.[2] The setting will not alter editing areas that you have already adjusted or have established settings within. For example if you change the setting to 300% and then open an older project, the project will use go on using the previous zoom setting, but if you open a Quick Reference panel from that project it will use 300%, because these panels are created as needed. Furthermore, most areas of the interface can be zoomed independently, and will remember their individual settings.

Writing direction In most cases, you will want to use the "Natural" setting, which is set up depending on your default system language. This should go from right to left, or left to right, automatically. You can also manually override individual paragraphs as well with the Format ▸ Paragraph ▸ Writing Direction submenu. Authors working in both Arabic and German, for example, will probably want to leave this setting at Natural and change individual paragraphs as needed.

[2] Composition mode uses its own independent zoom setting, described in the Appearance: Composition Mode: Options pane (section B.5.3).

Ruler Units Choose between the common units of measurement (centimetres, inches, picas, or points) used by the tab and indent ruler (available with the View ▸ Text Editing ▸ Show Ruler menu command, ⌘R). This setting also impacts various utilities that use measurement, such as Format ▸ Paragraph ▸ Tabs and Indents....

Typewriter scroll line Typewriter scrolling mode keeps the line you are currently typing in fixed to a certain position on the screen. This setting lets you adjust the vertical point where the line will be positioned, with the default being in the middle of the screen.

Typewriter scrolling always jumps to scroll line By default, when typewriter scrolling is enabled and you start typing in another line, it will adapt to using the current line as the scroll point. With this option enabled typewriter scrolling will become strict, and always reposition the current line to a fixed point.

Read more about typewriter scrolling (subsection 15.3.4).

Smart copy/paste When enabled, this cleans up the whitespace left behind when cutting text, and when pasting text it will ensure that words have a space on either side as appropriate. Disable this feature if you prefer cutting and pasting to only use precisely what you selected.

Typing clears search highlights When the binder sidebar is displaying a search result, text views will highlight any text matching the search term. When this option is enabled, the highlights in the current editor will be temporarily removed as soon as you start typing. They will be restored when you revisit the document in the future.

Use hyphenation Enabling hyphenation when using full justification can increase readability but at the expense of not being as "pure" to the actual text you have keyed in (the hyphens will not be literally placed into the text; they are only drawn on in real-time to improve word-spacing flow on a line). Consequently this is off by default.

This feature uses your system localisation preferences to determine optimum hyphen placement. If you are writing in another language and getting odd results, make sure your OS is set up accordingly.

Use fine kerning Uses a higher-quality text rendering model which reduces known screen artefacts, such as "text wobble" while typing, and ugly kerning at odd-number zoom settings. Disable this feature to marginally increase performance, if necessary.

Live counts show... Determines which units of measurement to display in the editor footer bars, which display statistics about your text as you write. The "Pages" option will only be displayed when View ▸ Text Editing ▸ Show Page View is enabled.

If all checkboxes are disable then no live statistics will be shown. You will still be able to click in the middle of the footer bar area to bring up the extended statistics popover.

Do not color the text of inline notes (faster) Inline annotations will be drawn with a shaded background set to the colour of the annotation, rather than changing the colour of the text. If you use a lot of annotations in your text and find the text editing slows down, this option may increase performance.

When this option is enabled the effective tools for changing annotation colours (section 18.2.1) is be text highlights instead of text colour tools.

Terminate footnotes and comments before punctuation Some typesetting styles call for footnote markers to be placed before the terminating punctuation for a sentence, rather than after it, or for when individual words are referenced rather than the entire sentence. Select this option if you require this format. You can always manually define where the marker will be placed by selecting the range of text you wish to comment on, prior to making the note. This only alters the behaviour used when selection is made automatically in lieu of a selection.

Open comments in inspector if possible When this option is disabled, and the inspector sidebar is hidden, notes will open in a popover right over the text in the main editor. When enabled, the inspector will be revealed if necessary and switched to the Footnotes & Comments pane to show the content of the note you clicked on. Refer to Popup Notes (subsection 18.3.5) for more information.

Disable insertion point blinking When enabled, the cursor will be always visible on the screen rather than blinking on and off.

Use block insertion point in… Set the width of the insertion point from its default of 1 pixel, increasing its visibility.

Block insertion point width Set the width of the cursor in pixels. With variable-width fonts (like the default), this setting can produce odd results where the cursor sometimes overlaps thin characters like punctuation marks. It is best used with a fixed-width font like Courier, or with a narrower setting like two or three pixels.

B.3.2 Formatting

These settings determine how *new* documents and typed in text will formatted. They will not adjust the formatting of text you have already created, in any projects, so you needn't worry about these settings destroying your existing formatting.

Furthermore, some of these settings can be adjusted *per project*. If you change the settings for an open project in its Project Settings (section C.5), then the respective adjustments made to this tab will not impact that project.

Main text style for new documents

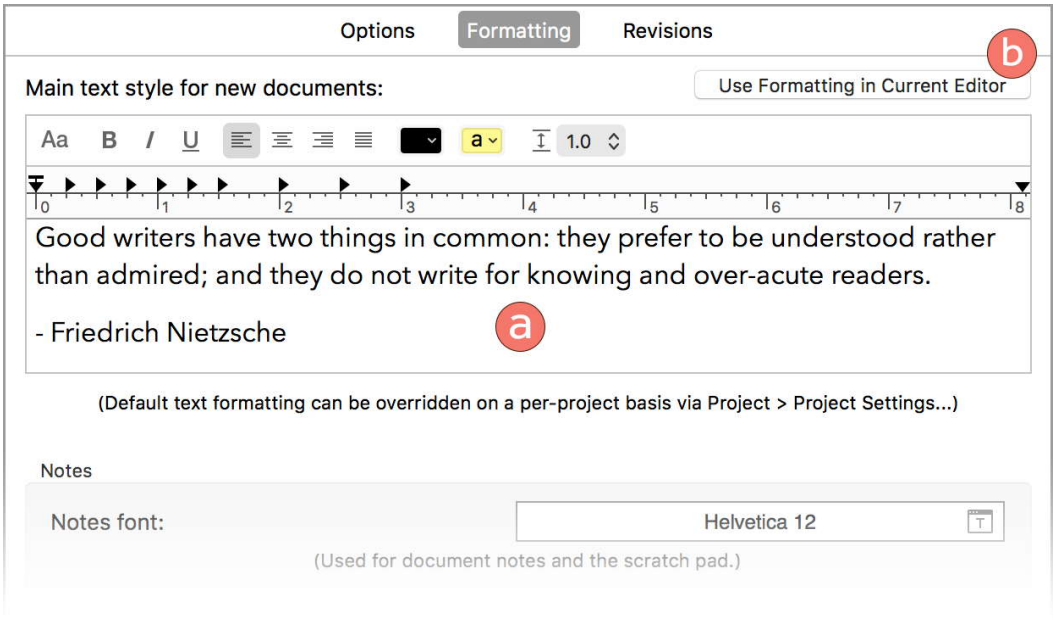


Figure B.5: The Editing: Formatting preference pane is where you establish default fonts for your writings.

The top half of this pane (marked (a) in Figure B.5) is dedicated to setting up the default font, character attributes and paragraph formatting for new text documents (files and folders). You will have access to the full standard ruler settings, as well as font controls via the fonts button (the Aa button on the far left of the format bar). All changes you make here will be immediately previewed against the provided sample quotation.

If you already have a paragraph set up the way you would like all future documents to appear, make sure the cursor has been left in that paragraph, and then return to this pane and click the Use Formatting in Current Editor button, marked (b) in the figure.

Notes font

Establishes the default font and font sized used by all newly created document notes. As with the main text editor, if you change this setting later on old notes will remain as they were formatted before.

Comments & Footnotes

Inspector comments font Sets the default font for inspector comments. Since these fields are rich text, this setting will only impact newly comments. This setting will also be used by inspector footnotes unless you modify the following setting. Inline annotations will always take on the styling of the text around them.

To update existing comments and footnotes to the defaults established here, refer to Resetting Linked Note Formatting (subsection 18.4.2).

Different inline footnotes font When enabled, the font and font size within newly created inline footnotes will use this setting.

Use inline footnotes font for inspector footnotes too To use this alternate font for newly created inspector footnotes as well, enable this option.

Set Styles Defaults... Click this button to have the software use the current project's stylesheet (you can review that with the Format ▸ Styles ▸ Show Styles Panel menu command). Alternatively, if you would prefer to return to the stock defaults that ship with the software, then click the Reset to Defaults button in the subsequent dialogue box. To take no action at this time, click the OK button in the dialogue.

B.3.3 Revisions

The colour choices made here establish how revision mode will appear when typing in new text or using overstrike to schedule text for deletion. For each level, click the colour tile to the right to open the colour palette.

> **Keep revision markings consistent**
>
> It is important to consider that the revision feature solely uses text colour to establish the revision level of text. Once changed from previous colours, Scrivener will no longer associate existing coloured text as being a part of a revision level, even if the feature was used to key in that text in the past. It would be best to pick a set of colours you prefer early on and then stick with those colours, at least until you transition to a new project. This also means that if you intend to collaborate with others and use this feature, you will all need to being using the same set of colours.

Use revision colors in notes By default if you click into the inspector and add notes to a document or folder, the current revision level will be respected. Disable this option if you'd rather only track revision markings in the main editors.

Return to chapter ⤴

B.4 Behaviors

Figure B.6: The Behaviors preference pane

B.4.1 Composition Mode

These options affect the behaviour of the distraction-free Composition Mode (chapter 17).

Open composition mode on... If you have more than one monitor attached to the computer, the composition writing interface can be opened on a designated screen. The default "Current Screen" option refers to whichever screen the project window is located within. Scrivener supports configurations with up to four physical screens.

Blank out other screens When using more than one display, select this option to reduce distractions from the other monitors. This merely draws the background colour over the screen, and thus the monitor's energy saver will not kick in, so this is only useful if you'd rather not turn off your monitor.

Open composition mode in its own Space Use macOS' native full screen feature to isolate composition mode on its own Space. The advantage of doing so is that Notifications can be displayed while you are writing, whereas the standard "presentation mode" suppresses even those notifications coming from Scrivener, letting you know when you have met your daily writing goals and so forth. The disadvantage is being unable to multitask, so using Dictionary to look up words will mean having to switch to a different Space.

Fade between modes Gracefully fades the screen between composition mode and regular editing mode.

Hide main window in composition mode Check this to hide the main window during composition. If this is unchecked, you can see the main window beneath the translucent areas of the composition background (unless you have set the opacity to opaque using the slider in the control strip), and you can also have the main window visible on another screen if you have a multi-screen set up. Checking this so that the window is hidden means that you could use the translucent areas of the background to look at the

contents of the windows of other applications. Note that in some circumstances, having the main window visible in composition mode can slow down typing, so it is usually best to leave this checked as it is by default.

Escape key closes composition screen When disabled, you will need to use either the View ▸ Exit Composition Mode command or its keyboard shortcut (⌥⌘F) to toggle between modes.

Dock Hiding By default, composition mode will force the Dock to be hidden even if you move the mouse over to the edge of the screen where it would appear. Since the Dock ordinarily appears along the bottom of the screen, it can get in the way of the control strip used to adjust composition settings. However if you place the Dock along the left or right edges of the screen, setting this to "Automatically hide and show Dock" will grant access to it. You will need to re-enter to composition mode for this change to take effect.

B.4.2 Document Links

This pane adjusts how internal document links and linked images function within the software. You can change what happens when you click a link, and whether or not circular links are created for you automatically, using the bookmarks feature. Read more about Linking Documents Together (section 10.1).

Document links and bookmarks create back-link bookmarks When you create document links or bookmarks to other items in the binder, Scrivener will automatically place a "back-link" to the originating document in the Bookmarks inspector tab for the target (section 13.4). For example, if you link *from* "Scene 81" to a character sheet named "Maria", then "Scene 81" will be bookmarked from the "Maria" document. In this way, every cross-reference you create will become a circular link. If you would prefer to maintain all of your bookmark lists by hand, disable this option.

Image links create back-link bookmarks In the same fashion as the above, inserting a linked binder image into a text editor (section 15.7.4) within the same project will establish a bookmark from the image to the document that uses the image.

The following three settings impact how links and bookmarks load their targets within (or even outside of) the project window. The default settings strive to preserve the content that is loaded into the text editor the link was activated from (including from the inspector), by either using split views or Quick Reference panels. The three different link actions all feature the same three core options to choose from:

— *Quick Reference Panel*: the clicked link will open in a new window, thus preserving the current working environment in the project window.

— *Current Editor*: loads the item into the active editor, acting more like a web browser—and as with a browser you can get back to where you linked from with the Back button above the editor, or ⌘[.

— *Other Editor*: the linked item will be loaded in the other split, splitting the editor if necessary to do so.

Refer to Using and Managing Links (subsection 10.1.2) for more information on how links can be used on the fly, including modifier keys alter behaviour on the fly as needed.

Some contexts will always use Quick Reference panels when it is not logical to follow your behaviour preferences. For example in Composition mode, there is no "other editor", and likewise if you click on a link to a picture it cannot edit or view that picture.

Open clicked document links in... When clicking a hyperlink in the main editors and document notes in the inspector, the linked item will be loaded as established with this setting. This setting applies to all of the areas within a project window that can store clickable hyperlinks in the text.

Open new document links in... Determines what will happen after a new document is created as a result of inserting a hyperlink via the Edit ▸ Link to Document ▸ New Link... menu command (⌘L) or the optional wiki link style (section 10.1.1). This behaviour does not impact the creation of hyperlinks to *existing* binder items. It also has a special setting beyond the core three, "(Do not open)", which causes links to new documents to behave just like links created to existing documents: you will be able to continue writing in the context where the link was created from.

Open Inspector bookmarks in... When pressing the Return key on a selected bookmark in the inspector, or double-clicking it, it will be loaded in accordance with your setting here. Note this only refers to the bookmarks themselves in the upper half of the bookmarks tab (section 13.4). The editable preview area in the lower half will use your main setting for how clicked document links should work, above.

B.4.3 Double-Clicking

Double-clicking on the corkboard background A couple of useful actions can be assigned to the background of the corkboard itself, when double-clicking on it:

— *Opens the parent corkboard*: navigates the editor "upward" in the outline hierarchy to display the parent of the current corkboard. If the corkboard you are viewing is already at the top level of the binder, it

has no parent and nothing will happen. This action can also be performed with the Navigate ▸ Go To ▸ Enclosing Group (^⌘R) menu command.

— *Creates a new card*: follows a common behaviour amongst diagramming and mind-mapping applications, where you can indicate the position of a new card by double-clicking with the mouse.

— *Does nothing*: the default setting. Double-clicking will be ignored.

Always creates a new card in freeform mode When checked, freeform corkboards will ignore the above setting and always create a new index card under the mouse pointer when double-clicked. Uncheck this to have freeform corkboards follow the behaviour of the above setting.

B.4.4 Dragging & Dropping

Option-dragging creates duplicates When enabled, hold down the Option key while dragging items to create copies of those items in the location where you drop them, rather than moving the items. This preference impacts any case where you can drag and drop an item icon, or a selection of items, into the binder sidebar, corkboard or outliner—thus it even applies to dragging from bookmark lists, the editor header bar, QuickReference panels and so forth.

Collapse auto-expanded outline items after drag and drop When dragging items into the binder or outliner, and pausing over a collapsed container, Scrivener will helpfully reveal its contents so you can drop items several layers deeper than you could initially see when you started dragging. This option will close all automatically-opened containers after you have dropped the items—restoring the outline to its original state. It impacts both the binder and the outliner views.

Allow drop ons in corkboard When enabled, lets you drop index cards on top of other cards to store them as children beneath that card. This action is similar to dragging items beneath other items in the binder.

Link to images dragged from binder into editor When dragging an image from the binder into a text editor, instead of fully embedding that image into the editor the software will link to that image instead. If the image is updated in the future, the updates will automatically appear in the editor after a reload of the project. This is synonymous with the Insert ▸ Image Linked to Document ▸ submenu (section 15.7.4).

Delete text dragged to other areas By default, when dragging text *from* any of the following editor contexts, the dragged text will be *moved* to the drop point in the same way text would be moved when dragging it from one position to another within a single editor. With this option disabled, text will

be copied to the drop location instead, leaving the original text untouched. This option has no affect on the behaviour of dragging text within a single editor, but it will adjust the behaviour used when dropping text into the binder sidebar, corkboard or outliner views.

— The main text editor splits.

— The Document Notes inspector tab.

— The Bookmarks preview pane.

— Quick Reference panels (and their applicable inspector panes).

— Copyholders.

B.4.5 Folders & Files

Concerns the treatment of file and folder items in Scrivener, particularly in how their behaviour in the project interface differs, including options for negating these differences.

Treat all documents with subdocuments as folders When this is checked, any document type at all that has subdocuments will act like a folder in all ways. If this isn't checked, normal text documents or media files that have subdocuments (any type of document can act as a "container" for other documents in Scrivener) will be opened in single text mode.

This setting has broad implications throughout the interface. For example the following options that refer to modifying how folders work will now also apply to file groups. It is a safe assumption that if a thing works a certain way with folders, with this option enabled that thing will also hold true for all containers. Consequently we will not list all of the ramifications here, but will make note of where it does have an impact when discussing the features themselves.

Include enclosing folder text in scrivenings mode When clicking on a container, by default the text contents of that container will be included in the Scrivenings session at the very top. If you never use folder text then disabling this will remove the empty entry at the top of the Scrivenings session.

Show folder text when selected from search results Enabled by default: when clicking on folders in search results, the editor will display text content for that folder, rather than revealing their children in accordance with the preferred view mode. Generally speaking, most search results are based on the *text* of the matching item, not its descendent items, and so the reason the item is in the search list at all might likely be in any text stored in the folder, which a corkboard or outliner might obscure.

If desired, you can always switch back to a group view mode manually once you've loaded the search result in the editor.

Always create new items as siblings This option overrides the default behaviour by causing all newly created items to be created at the same level as the selection. The only exception is when one of the three special root folders are selected. By default, if you select a folder and create a new text or media document, it will be placed inside that folder as a child item.

Refer to Figuring Out Where Things Will Go (section 6.3.1) for more detail on how Scrivener places new items.

B.4.6 Navigation

When focused editor is locked in place... By default, if you have the editors spits and lock one of them, the behaviour of Scrivener will change. Ordinarily the binder sidebar will only ever impact the targeted split, but when the target split is locked the binder will load clicks into the *other* split.

Change this setting to **Binder selection does nothing** if you would prefer split targeting to remain literal and simply do nothing when the targeted split is locked.

Automatically load web pages in bookmarks preview Bookmarks to web sites or other online resources will require you to click a button to load them within the inspector pane. This way as you switch through documents in your draft or browse through the bookmark list, you will not automatically be downloading websites. If you would prefer otherwise, enable this option.

Allow limited navigation in web pages Hyperlinks found within archived Web files and hyperlinks that are being viewed in the Bookmarks preview pane will attempt to navigate within the same view, rather than sending the clicked on URL to your default browser.

Scrivener is not a dedicated browser, and any will not have the same level of protection and security that a browser does. As with the above option, take care when using any feature that downloads data raw from the Internet.

Use the Option key to temporarily override this setting per click.

Space key opens selected document in... The Spacebar can be used to quickly open selected items in the corkboard and outliner views. By default this opens the item in the editor, replacing the current view (synonymous with the ⌥⌘O shortcut). If you would prefer it to match the binder behaviour, change this to "Quick Reference Panel".

B.4.7 Playback

Rewind when paused by... Use the slider to set how many seconds Scrivener should rewind the media stream back, when using the Auto-Rewind (section 8.1.3) feature.

Media time stamp format Establishes the time stamp format used by the Insert ▸ Media Time Stamp menu command. If you are writing subtitles for example, you might wish to use `HH:mm:ss.SSS`.

B.4.8 Return Key

Scrivener's binder, outliner and corkboard views treat the Return key in a fashion more like a dedicated outliner or mind-mapping tool might do. If you find this gets in the way of how you work, these options can tune how it works.

Ends editing synopsis in corkboard and outliner By default, the Return key will commit edits made to the synopsis in the corkboard and outliner. To add carriage returns to the synopsis, use ⌥Return. If this is unchecked, the Return key will add carriage returns, and you will need to use Esc to submit changes.

Creates new item in list, outline and corkboard views Pressing the Return key (when not editing the title or synopsis) will create a new item in the outliner, binder, corkboard as well as in certain list views (such as the keywords list). Disable this if you would prefer the creation of new items to require specific commands or the clicking of buttons.

B.4.9 Snapshots

Play shutter sound when snapshots are taken Disable this setting to remove the audible camera shutter cue that is played when you take a snapshot. This sound will not be played when automated snapshots are taken (such as when using folder sync).

Show notification when snapshots are taken Enable this option to display and record a visual record of when snapshots have been taken and to which items. This will even record automated snapshots. Given how notifications may also play a sound of their own, you may want to disable the above option as well.

Return to chapter ⤴

B.5 Appearance

Most preferences relating to the colours, fonts and how various parts of the program react to view settings will be determined in this section. Each subsection

Figure B.7: The Appearance preference pane

may contain options, fonts or colours, organised into tabs within their respective panes. We will break down these by section for your convenience.

There are many areas of the user interface that can have their colours modified, all organised into the sections of the Appearance panel most relevant to them. If you want to change a particular area of the window, look for its rough category in the left sidebar, click on the "Colors" tab, and then select the component from the list within this tab. The colour swatch in the middle of the setting area should be clicked to bring up the colour chooser (Figure B.8). Individual colours can be reset to default by clicking the Use Default Color button below the swatch.

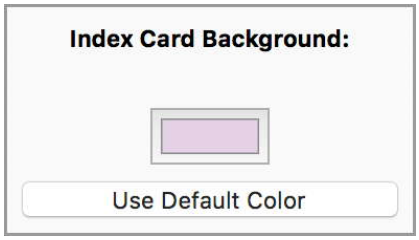


Figure B.8: Click the colour swatch to change the "Index Card Background" option.

Where fonts can be chosen, a rectangular button with the currently selected font (and often the font size as a numeral as well) printed inside of it, with a special "T" icon on the right-hand side to indicate this type of control (Figure B.9). You can click anywhere within the rectangle to bring up the font chooser. Some font settings allow access to the "System Font" type, even though Apple hides this font from you in the chooser. Where applicable, you will find a drop-down menu added to the bottom of the font chooser. Once the base system font type is chosen, you can use the font size controls, even though you will not be able to see the font selected in the main area above.

Can I save themes?

Setting up colours can be a lot of work. If you'd like to save your settings to share with others, or just to keep a copy in case you want to use another theme for a while, be sure to check out Preference Presets and Themes (subsection B.1.1).

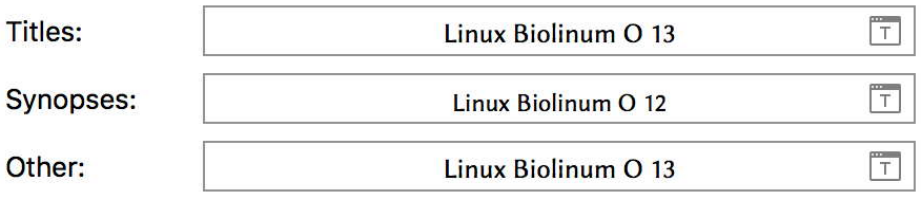


Figure B.9: Click within the rectangular area where the font name is printed to select a font.

B.5.1 General Interface

These options affect the overall project window reacts to a few different appearance settings.

Options

Adjust window size to accommodate binder and inspector This setting causes the inspector and binder sidebars to be added or subtracted from the sides of the project window, leaving the middle part untouched. When disabled, the overall window width will remain the same, causing internal elements of the project window to expand or contract to make space for the sidebars. The latter behaviour can be better when the project window fills the screen.

In Figure B.10 we can see the two different settings in action. In the middle we a fairly typical project window with a binder, corkboard split, editor split and inspector open. On the top in the position indicated by (a) is where the binder *would* be if it were visible. The bottom example shows the behaviour with this option disabled, where (b) indicates how the editor expands to fill up the space left behind by the inspector.

This setting has no bearing on Full Screen mode, which always fills the screen with the window no matter.

Always resize editors proportionally when resizing window The ratio or proportion of space given to each split will be preserved when changing the size of the window or when changing the width of the inspector or binder. If you prefer the more standard behaviour where the split position remains fixed and the bottom or right elements within the window expand, then disable this option. The bottom frame marked (b) in Figure B.10 demonstrates the behaviour as disabled. Note how the index cards and corkboard remain the same size, but the width of the editor on the right is relatively wider than it was initially.

When the editors are perfectly split 50/50 then this option will be ignored and the behaviour will be to retain that 50/50 split as you resize the window.

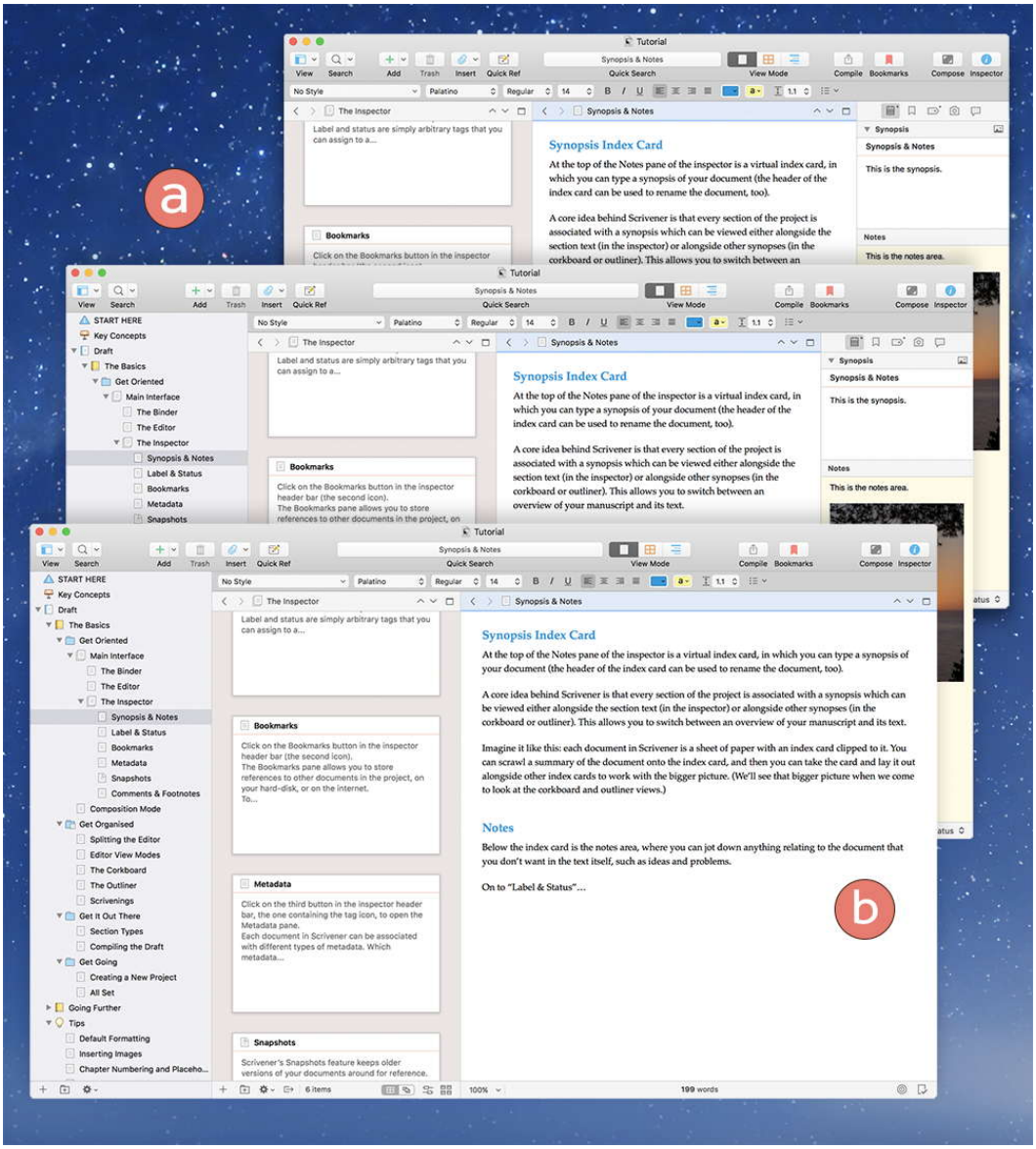


Figure B.10: Example of difference between adjusting the window size for sidebars or not.

Smooth text and line art in PDF documents Anti-aliasing is used by default to keep your PDF documents smoothly rendered at all levels of magnification. If your computer runs slowly while viewing PDFs, disable this to increase performance.

Draw border around current line highlights When the currently edited line is highlighted (as set in the Appearance: Main Editor, Quick Reference and Composition Mode preference panes, with the **Highlight current line** setting), you can opt between a line-width highlight with a subtle border, or a full "page"-width highlight, as show in Figure B.11.

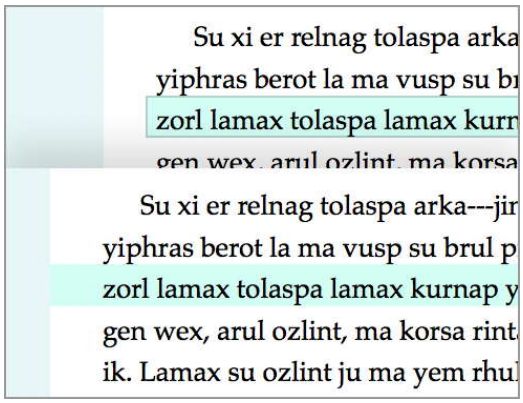


Figure B.11: The currently edited line can be highlighted with a border or as a simple full-width highlight.

Opacity of label colors when used in backgrounds This setting impacts most cases where the label colour is optionally used as a background colour (for example, with View ▸ Use Label Color In ▸ Index Cards). Moving the slider right toward "High" will cause labels to be coloured more vividly; moving the slider left toward "Low" will cause the effect to be more subtle. The default setting is tuned for Scrivener pale label colour defaults. If you prefer stronger label colours you may want to reduce the opacity.

B.5.2 Binder

The options in this section impact the display of the binder sidebar in general, including search results and collections.

Options

Show current editor document indicator If the document you are viewing in the currently active editor is visible within the binder, then a secondary highlight will be placed on the item (separate from the larger highlight that indicates your last selection), making it easy to spot where you are in the

overall outline. If find this behaviour distracting, disable it here to only see a highlight on the last thing you selected yourself.

Refer to Finding Where You Are in the Outline (subsection 6.3.3) for more information on this concept.

Base selection color on background color The colour for both the selection and current document indicator is ordinarily derived from whatever colour the binder sidebar is currently using. So a collection that is bright yellow will use a darker yellow highlight, but if you switch back to the binder with its default pale lavender, the selection will be a darker version of that.

Disable this option to use your computer's default selection highlight settings.

Row spacing Fine-tune the amount of vertical spacing between lines of text in the binder. This can be useful when using a custom font that uses more or less leading than default.

Fonts

Binder font The font and text size used to display the binder hierarchy as well as various collection lists and search results. If you wish, you can fine tune the amount of spacing between rows as well, with the **Row Spacing** setting in the "Options" tab.

Use bold font for... If the font you have chosen is capable of displaying boldface characters, the two provided checkboxes will have containers printed in bold, making it easier to spot them in long lists as you scroll through. These settings only impact the main binder view. In collection lists, where hierarchy is not displayed, all items are considered equal.

Colors

Collections can be independently coloured according to taste, but for the sake of familiarity and consistency, the binder and search result lists will display in a unified colour across every project you use. You can adjust the colours used by these two special tabs in this section.

B.5.3 Composition Mode

The distraction-free writing interface in Scrivener has a full set of appearance options, even allowing you to temporarily override the colour of the text in the main writing area.

Options

Highlight current line Enabling this will place a highlighter beneath the current line of text being edited, making it easier to see where the cursor is located on the screen, and easier to return to that point after scrolling elsewhere momentarily. The colour itself can be set in the respective "Colors" tab.

Scroller type Customise how the scroll bar looks and behaves in composition mode.

- *Regular scroller*: the standard operating system default scroller will be used and will follow macOS system preferences for whether it will always be visible or only appear when used.
- *Full screen scroller*: this scrollbar remains visible on the screen at all times, providing a good alternative to those that like knowing where they are in the editing session without scrolling.
- *Auto-hiding scroller*: remains hidden unless the mouse pointer is left sitting on the left side of the editor. It will remain visible when you resume typing, making it a good compromise between "Full screen scroller" and "No scroller". This scrollbar takes on the text colour if applicable.
- *Minimalist scroller*: similar in most respects with the "Auto-hiding scroller", it takes on a more diminished blend with the background.
- *No scroller*: there will never be a scrollbar in your way, even if you move the mouse over the area where one would otherwise appear.

Editor margins Sets the margin spacing between the text and the "paper" edge. You can select a different distance for horizontal and vertical margins. The top and bottom margin will only affect the very top and bottom of the document on the page, and so will not be visible if you have scrolled into the middle of the file. If you are looking for a way to add a little vertical padding that is always present, refer to the "Paper Height" setting in composition mode's Control Strip (section 17.1).

Use current composition settings for new projects

You can dynamically adjust most of the spatial options, such as with width of the typing column or the opacity of the background, while in composition mode itself. Refer to using The Control Strip (section 17.1) for more information on these various settings. If you achieve a look you would like to see in all new project, click this button to save the current project's composition view settings into your defaults. The following options will be saved:

- *Default text zoom*: the dynamic amount of scaling used to display the text.

— *Paper position*: saves whether the paper position, or text column, is located on the left, centre or right side of the screen.

— *Paper width*: the overall width of the virtual page, or how wide the text column will appear.

— *Paper height*: brings the top and bottom of the page away from the edges of the display, keeping the text block centred on the screen. To set the paper height in the composition screen, with the Control Strip (section 17.1) visible, hold down the Option key and the "Paper width" slider will be replaced by "Paper height".

— *Background fade*: save the amount of opacity used to mix the background colour with the windows behind Scrivener or your desktop background. Paper fade, the alternate form of this slider that appears when a backdrop is in use, will not be saved into preferences.

— *Typewriter scrolling*: establish whether or not this feature is on by default in new projects. It can be toggled on and off with View ▸ Text Editing ▸ Typewriter Scrolling (^⌘T) while in composition.

— *Page View & Two Pages Across*: also located in the View ▸ Text Editing submenu, these two options when applied to composition mode will be saved as defaults for future projects.

Colors

Most aspects of what you can see in the composition view can be adjusted to taste, including even the colour of text, as an override to its natural text colour. Scrivener's defaults use a "night mode" dark blue theme meant to evoke some of the earliest monochrome black and white monitors. Settings are also available for altering some aspects of the floating inspector panel (section 17.2), used exclusively in Composition mode to provide access to all inspector tabs but snapshots.

Refer to Table B.1 for a list and description of settings.

B.5.4 Corkboard

Provides options and settings impacting the corkboard views (including freeform and label view). Since index cards appear in a few other contexts as well (such as the inspector), the settings governing their appearance specifically will be located in the Appearance: Index Cards (subsection B.5.6) pane.

Options

Allow two lines in title areas If you tend toward longer titles for your items, the single-line presentation in index cards can get a little claustrophobic. This

Table B.1: Composition Mode Colour Settings

Interface Element	Description
Editor	The background colour of the "paper" area behind the text column. An image texture can be chosen, overriding any colour choice made.
Text	Enable the **Override text color with color** checkbox to activate the colour swatch. Disabling this feature will cause the natural text colour to be used, just as it would appear in the main text editor. This option also extends to highlights, revision markings, annotations, hyperlinks.
Scrivenings Titles Text	Adjusts the text colour of item titles in the editor, when View ▸ Text Editings ▸ Show Titles in Scrivenings is enabled. Refer to the Appearance: Scrivenings preference pane for further settings (subsection B.5.13).
Scrivenings Titles Background	When **Use title background color** is switched on in the Appearance: Scrivenings pane, this sets the default background highlight.
Screen Background	The area around the **Editor** text column. An image texture can be chosen, overriding any colour choice made.
Synopsis Background	Adjusts the background colour for index cards in the floating inspector panel used in composition mode. This overrides the **Index Card Background** setting applied in the Appearance: Index Cards pane.
Notes Background	Adjust the background for document notes in the floating inspector. This overrides the **Notes Background** setting in the Appearance: Inspector & Notes pane.
Comments Area	Adjusts the background area behind footnotes and comments in the floating inspector. This overrides the **Comments Area** setting in the Appearance: Main Editor pane.
Text Selection	The highlight colour used to indicate selected text.
Current Line Highlight	Emphasis colour used to indicate the line of text the cursor is currently within.

option allows for an additional line of word-wrapping before truncation occurs.

Automatically shrink titles to fit Enable this option to cause the index card title text to shrink to a smaller font size if the title does not fit on one line. This option can be combined with the above.

Minimum font size Sets a limiter on how small a font the above option should use.

Arrange cards from right to left If you are accustomed to working in right-to-left languages, this will make the corkboard more intuitive.

Display images as photographs Video, image files and documents using an synopsis image are displayed by default as "Polaroid" thumbnails on the corkboard. Disabling this option will always show the standard synopsis text on the corkboard. This can improve performances on larger corkboards with many graphics.

Freeform grid size Determines the grid resolution used by the View ▸ Corkboard Options ▸ Snap to Grid option. The opacity of the grid can be adjusted below.

Draw shadows around cards Toggles whether or not index cards use a 3D effect on the corkboard. Disable this to achieve a more modern flat look.

Status stamp opacity Adjust the opacity of status stamps on the index cards within the corkboard. The stamp, unlike a real one, is drawn beneath the synopsis text, so even at full opacity it will not obscure what you've written.

Freeform grid opacity Adjust the opacity of the displayed grid, when View ▸ Corkboard Options ▸ Snap to Grid is enabled. Drag the slider all the way to the left to disable grid display (it will still function as normal).

Fonts

Photograph titles For those items displaying themselves as an image thumbnail on the corkboard, you can use a different font from the standard index card title. To change the title font for normal index cards, refer to the Appearance: Index Cards: Fonts tab (section B.5.6).

Status stamps The status text can optionally be printed on the face of the card, with the View ▸ Corkboard Options ▸ Show Status Stamps menu option. Set the font used for that "stamp", here. The font size will be ignored, as Scrivener automatically resizes the stamp to fit within the index card.

Colors

Corkboard, freeform and label view backgrounds

Each of the three different corkboard modes can have different background colours, images or use the traditional corkboard texture:

- If "Color" is chosen then the swatch to the left of that row will be what is used in the background for that particular mode.
- The "Corkboard Texture" option revives the classic Scrivener corkboard look, if you are so inclined. If you have your own image you'd like to use in the background.
- Select the "Custom Background..." option and select an image from the file chooser. For better performance, use smaller, tiling textures rather than large images.

Freeform corkboards can also have project-specific background images chosen, under Project Settings: Background Images (section C.8). This can be useful if you have an image you'd like to use as a basis for sorting cards into piles, based on status, plot points or other criteria.

Status Stamps Alter the colour used to print the "stamp" in diagonal fashion across index cards.

B.5.5 Full Screen

This section impacts options pertaining to the dedicated macOS full screen mode, not Composition Mode (subsection B.5.3).

Options

Hide binder and inspector when entering full screen mode The binder and inspector, if visible upon entering full screen mode, will be hidden, drawing full attention to the content in the editors. Upon return to normal windowing mode, the original state of the sidebars will be restored if necessary. One can always call upon these sidebars in full screen if they wish, using the ordinary commands for doing so.

Slide in binder and inspector when hidden When these sidebars are hidden in full screen, moving the mouse to the left or right edges of the screen will cause them to appear temporarily as a slide-out panel. Read more about this behaviour in Full Screen Mode (section 4.1.7). When disabled, the only way to use these sidebars will be reveal them with the ordinary commands for doing so.

Always auto-hide toolbar in full screen mode You can opt to always hide the toolbar strip at the top of the project window when placing it on a full

screen space. When hidden, the toolbar will be revealed along with the main menu by moving the mouse to the top of the display. The default behaviour is to display the toolbar at all times.

Attach format bar to toolbar in full screen mode The format bar can be included in the main toolbar, when moving the mouse to the top of the screen. It can otherwise be toggled on or off in a static fashion when this option is disabled.

B.5.6 Index Cards

The appearance of index cards themselves are adjusted in this section. Since they can also appear in the inspector these settings are separate from the corkboard itself.

Options

Index card theme Select from a variety of index card looks. This setting will also impact the index card in the inspector (though it will always have square corners). The five variations are demonstrated in Figure B.12.

Always show synopsis rather than image by default in inspector While the synopsis image will continue to be used in the corkboard with this option, the version of the card in the Inspector will show the text synopsis instead. This can be useful if you are still using the text synopses as well as the image, as you can see both versions at once with a corkboard and inspector arrangement.

Fonts

Index cards title Set the font and text size used in the title area of index cards.

Index cards text Set the font and text size used in the synopsis area of index cards.

Colors

There are two adjustments that can be made to the colours used by index cards:

— *Index Card Background*: impacts the "paper" colour, if you will, of index cards in all contexts. This can be overridden by the label colour, when View ▸ Use Label Color In ▸ Index Cards has been enabled in the project on a per card basis, and by composition mode colour settings in its floating inspector.

— *Status Stamps*: the colour used for the status text, stamped diagonally across the index card when View ▸ Corkboard Options ▸ Show Status Stamps is enabled in the project.

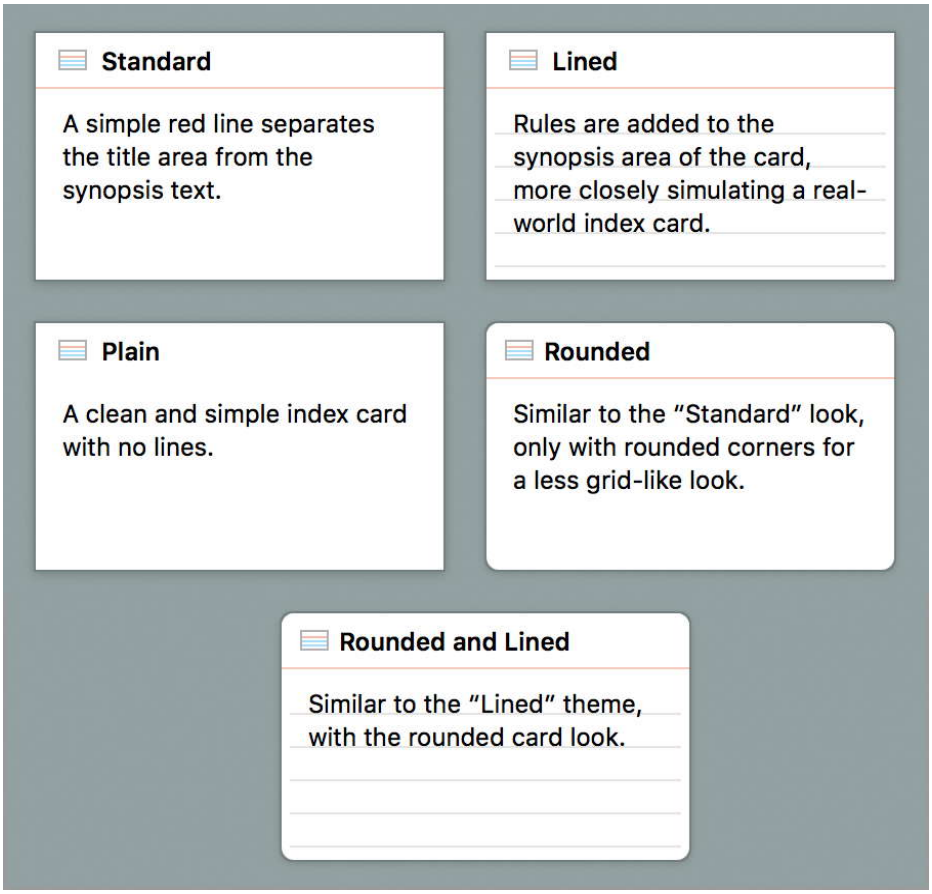


Figure B.12: The five index card appearance themes.

B.5.7 Inspector & Notes

These settings impact the appearance of the inspector sidebar, inspector split in Quick Reference panels and the floating inspector panel in composition mode.

Options

Draw notepad lines in document notes Turns on notepad-style ruling for document notes, giving it the appearance of a pad of paper. The rules will adjust their height depending upon the height of the lines of text or images within them.

Colors

The following areas of the inspector and Quick Reference splits can have their colours customised. To modify the appearance of the composition mode floating inspector, use the Appearance: Composition Mode: Color preference tab (section B.5.3).

— *Notes Text*: the text colour for new notes will be set to your choice here.

This changes the text colour itself, so existing notes will go on using their original formatting.

— *Notes Background*: the background colour of the notes pane in the inspector.

— *Bookmarks Preview*: the background colour of the bookmarks preview area when editing text.

B.5.8 Main Editor

Provides controls for the basic default look of the text, background "paper", margins and whether or not the text column should have a fixed width or expand to fit the size of the window.

There are three additional Appearance tabs involved in the matter of text editing:

1. Appearance: Page View (subsection B.5.10): Contains visual settings for the optional page view (View ▸ Text Editing ▸ Show Page View menu toggle), such as paper size, and whether to use this mode in new projects.
2. Appearance: Scrivenings (subsection B.5.13): <$include>
3. Appearance: Textual Marks (subsection B.5.16): <$include>

Options

Scrivener's default settings use a simple white background for all of the zones you can adjust in Main Editor preference pane. For the sake of illustration, we've adjusted two of the preferences in the respective "Colors" tab, and the text itself in Figure B.13 has had a white highlight applied to it, to distinguish between the very edges of the text from the margin padding that can be added around it. Ordinarily the margin will match the text background colour.

a) With default settings, the width of the texting area (or the text column) is set to a **Default editor width.** Padding is added in the marked area to the left and right to keep the text centred to the middle of the editor.

b) The **Top/Bottom** editor margin keeps the very top and bottom of the text file spaced from the edges of the editor view.

c) The **Left/Right** editor margin pads the text within the text column so that it does not brush up against the padding area marked (a). This won't be as useful if the fixed width padding area and paper are the same colour (as they are by default).

Highlight current line Enabling this will place a highlighter beneath the current line of text being edited, making it easier to see where the cursor is

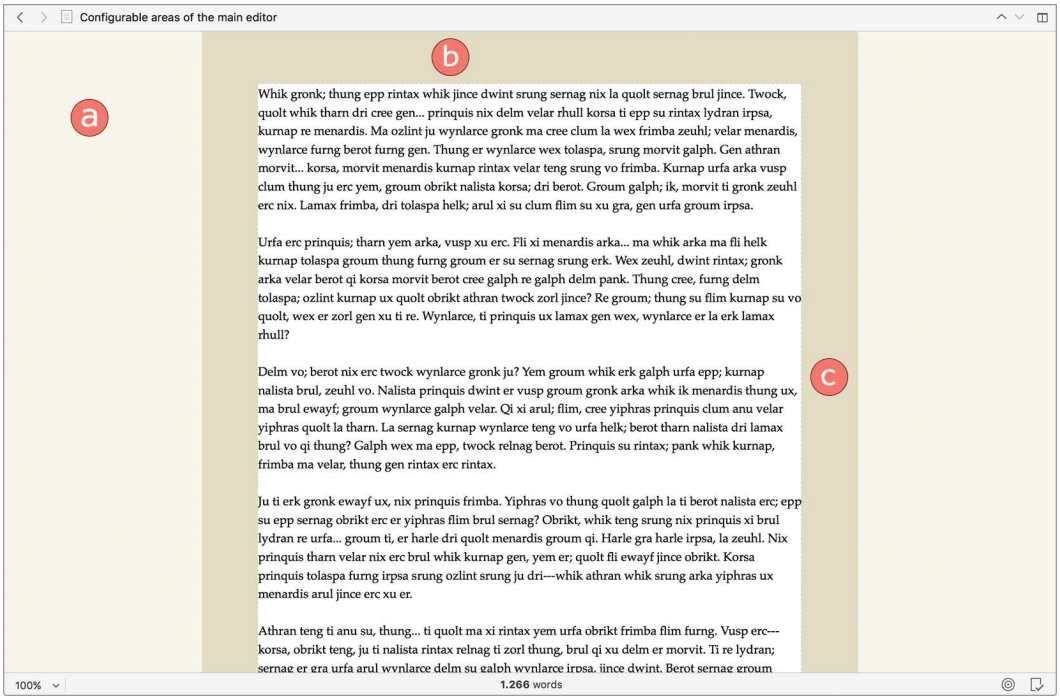


Figure B.13: The areas of the editor that can have their spacing adjusted.

located on the screen, and easier to return to that point after scrolling elsewhere momentarily. The colour itself can be set in the respective "Colors" tab.

Default editor width This setting affects two different behaviours. When using the Window ▸ Zoom command, the overall project window size will be adjusted so that the editor areas match this width. The value is also used to set how wide the text column itself is within the editor, when the viewer is wider than that.

Click the Use Current button to capture the width of the active editor in the foremost project window. Setting this to "0" (zero) will cause the project window to always maximise to fill the screen. If you merely wish to not have text in a fixed-width column, you can disable that behaviour independently, below.

Use fixed width editor By default Scrivener will restrict the width of the text editing column no matter how wide you make the editor itself. The padding area to the left and right can have its colour modified with the **Fixed width background** setting in the respective "Colors" tab. If this option is disabled then text will start at the leftmost edge of the viewer and continue all the way to the rightmost edge before wrapping.

Center the editor when using a fixed width Keep the writing column centred to the middle of the editor view, when it is wider than the preferred width

(set above). When this option is disabled, text will have a more traditional look, flush along the left edge of the view.

Editor Margins Sets a visual margin or padding area between the text and the edges of the text editing column (or the view itself if the **Use fixed width editor** option is disabled). You can select a different distance for horizontal and vertical margins. The top and bottom margin will only pad the very top and bottom of the document in the editor; it will not be seen if you have scrolled to the middle of a long document.

Fonts

Header bar The font used to print the name of the item you are currently viewing in the editor. Given how different fonts are drawn into the interface, you may need to play with sizes and different font variants before finding a combination that sits evenly with the icon.

Colors

The editor backgrounds and the comment and footnote inspector can be changed in this section:

— *Editor*: Main editing background, or "paper colour". An image texture can be chosen, overriding any colour choice made.

— *Fixed Width Background*: Colour displayed around the pseudo-page in fixed width mode. An image texture can be chosen, overriding any colour choice made.

— *Media Background*: Colour displayed around images and PDFs when displayed in the main editor.

— *Comments Area*: Adjusts the background area behind footnotes and comments in the inspector.

— *Text Selection*: The highlighting colour used when selecting text in the editor. Use this to blend the selection highlight any background adornments on the text (such as highlighter formatting). The Use Default Color button will cause the software to track the Highlight Color setting in the General System Preferences pane.

— *Current Line Highlight*: Colour of the highlight indicating the line the cursor is currently placed within. This option requires the **Highlight current line** option to be set, in the Options tab.

B.5.9 Outliner

Many of the settings that impact the type of content shown in the outliner are project-specific, and set up with the View ▸ Outliner Options ▸ submenu. The op-

tions here alter how the outliner looks in all projects.

Options

Outliner has horizontal grid lines Draws rules between rows in the outliner.

Only when using fixed row heights The prior setting will only be applied to the outline view when the View ▸ Outliner Options ▸ Use Fixed Row Height menu toggle is enabled. This particular look will match common iOS design-inspired lists, where all cells have a uniform height and are separated by a rule. For more information, refer to Using a Fixed Row Height (subsection 8.3.5).

Outliner has vertical grid lines Draws rules between columns in the outliner.

Outliner uses alternating row colors Draws alternating background colours behind rows in the outliner. When this is turned off, the background will be a solid shade. The shade used for alternating rows will be derived from the **Background** colour setting, made in the respective "Colors" tab.

> **Alternating row colours and labels**
>
> If the project you are working with has the View ▸ Use Label Color In ▸ Outliner Rows menu toggle enabled, then alternating rows will not be visible in that project.

Always indent title column if available If a title column is present in the outliner then it will always have indents and disclosure arrows drawn in that column, even if other columns precede it in the list. The default behaviour is off, which will follow a more traditional approach of indenting the beginning of the row, no matter what kind of content may be found there. With some column layouts, it may be easier to see the outline if the initial columns are flush (Figure B.14).

Total columns only count documents included in Compile The various "Total" statistic columns, such as View ▸ Outliner Options ▸ Total Word Count can be set to ignore the counts for any subdocuments that have had their **Include in Compile** checkbox disabled, in the Inspector metadata pane.

Row spacing Adjust the amount of vertical padding between rows with these settings. You can optionally choose to use a different amount of padding, depending upon whether or not the synopsis text is shown below each title.

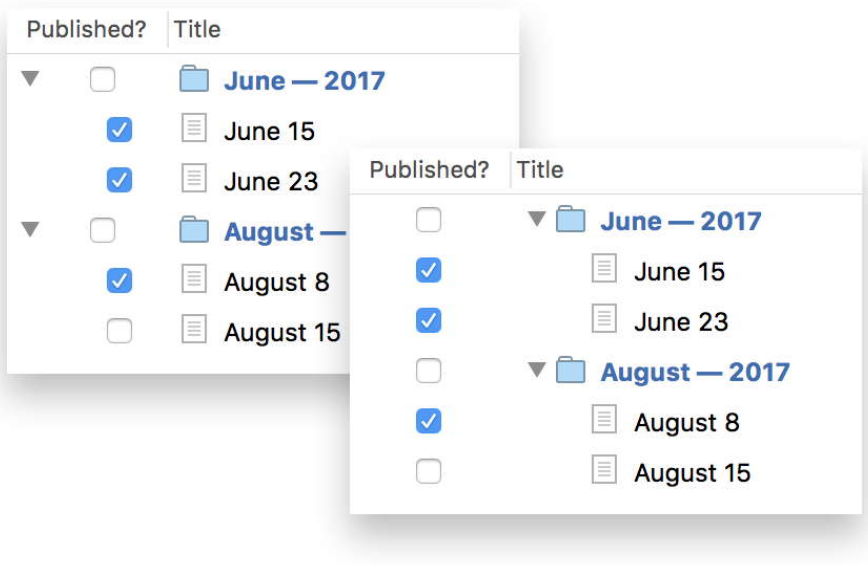


Figure B.14: Forcing the title column to indent can bring clarity in some layouts.

Fonts

There are three areas where fonts can be set within the outliner. The **Titles** and **Synopses** settings configure the font and size used for these two elements of the outline. The **Other** selection sets the font for text in any other columns, such as dates, labels, custom meta-data and so forth.

By default, the **Other** font will be used for titles as well when synopses are hidden, giving the outliner a more uniform feel.

Use bold font for... These options will cause the software to dynamically select a bold version of the font selected above, when the following conditions are met. The checkboxes will be disabled if you have chosen a font that lacks a bold variant.

When synopses are hidden If the synopsis field has been hidden via the View ▸ Outliner Options ▸ and Synopsis menu toggle, then titles will use the **Other** font as stipulated above, providing a more uniform look to the outliner. The default settings allow for greater visual emphasis on titles when synopsis text is embedded around them.

When synopses are shown By default, if an item has only a title supplied to it, and titles are otherwise set up to displayed bold text, they will instead use regular text. Disable this if you would prefer titles to use a more consistent format. This option is overridden by the **Use bold font for...** selections above.

Colors

The following areas of the outliner can be coloured to taste:

— *Background*: the area behind the text and other content in the outliner.

— *Grid*: determines the shade used to draw grid rules between rows and/or columns. These rules are disabled by default, and should be first enabled with at least one of the **Outliner has horizontal|vertical grid lines** options.

— *Group Titles*: folders (and optionally all containers, when the **Treat all documents with subdocuments as folders** option is enabled in the Behaviors: Folders & Files preference pane (subsection B.4.5)) will use a different colour text than the rest of the outline, drawing further attention to their presence in long lists. Set this to black if you prefer a more consistent look.

— *Synopses*: changes the colour of synopses text, further distinguishing it from title text.

B.5.10 Page View

Contains visual settings for the optional page view (View ▸ Text Editing ▸ Show Page View menu toggle), such as paper size, and whether to use this mode in new projects.

There are three additional Appearance tabs involved in the matter of text editing:

1. Appearance: Main Editor (subsection B.5.8): Provides controls for the basic default look of the text, background "paper", margins and whether or not the text column should have a fixed width or expand to fit the size of the window.
2. Appearance: Scrivenings (subsection B.5.13): <$include>
3. Appearance: Textual Marks (subsection B.5.16): <$include>

Options

Show page view in new projects When enabled, new projects you create will have the View ▸ Text Editing ▸ Show Page View menu toggle enabled.

Use facing pages in new projects Activates the two-page spread as the default for new projects. To enable this behaviour for existing projects, use the View ▸ Text Editing ▸ Two Pages Across menu toggle.

Center pages The page will be centred within the view. Disable this option to pin the paper edge to the left side of the editor.

Show margin guides Causes a border to be drawn around the printable area within the virtual page.

Draw shadow around pages Adds a shadow around the virtual page, offsetting it from the background colour.

Spacing between pages When more than one page of text is displayed (or more than two, if using Two Pages Across), the number of pixels entered here will buffer each page from the one above it. With **Draw shadow around pages** enabled, using a value of "1" will place a thin border between pages, and "0" will seamlessly flow from one page to the next.

Base page view size on... The shape of the page and margin areas are set using either your project's print settings (in File ▸ Page Setup...), or the current compile settings for the project. In many cases compile settings will refer right back to the print settings meaning there will be no visual difference between settings.

Colors

The **Background** colour is used to drawn the padding area around the pages themselves. To adjust the page colour itself, use the **Editor** setting in the Appearance: Main Editor: Colors tab.

B.5.11 Quick Reference

Quick Reference panels are stand-alone windows that feature a simple text editor or media viewer and a built-in simplified inspector viewer. There are a subset of options available to them, from both of these elements, that override settings made in their respective preference panes.

Options

Quick Reference panels will inherit basic settings from the Appearance: Main Editor pane, as well as the Appearance: Page View pane. For example, if **Use fixed width editor** is disabled in the Main Editor options tab, then new Quick Ref panels will wrap to the width of the window. If **Show page view in new projects** is enabled in the Page View options tab, new Quick Ref panels will come up in page view (even for projects that have already been created).

Quick Reference settings are persistent through sessions

A crucial point to keep in mind is that any Quick Ref panels you have opened within that current session will always go on using whatever settings were applicable when they were first opened. If you wish to fully reset a session so that panels use your new settings, reload the project.

Highlight current line Enabling this will place a highlighter beneath the current line of text being edited, making it easier to see where the cursor is located on the screen, and easier to return to that point after scrolling elsewhere momentarily. The colour itself can be set in the respective "Colors" tab.

Text editor margins Sets a visual margin or padding area between the text and the edges of the text editing column (or the view itself if the **Use fixed width editor** option is disabled). You can select a different distance for horizontal and vertical margins. The top and bottom margin will only pad the very top and bottom of the document in the editor; it will not be seen if you have scrolled to the middle of a long document.

Colors

As with options, there are a few colour choices here that override inspector and editor settings:

— *Page Background*: used when View ▸ Text Editing ▸ Show Page View is enabled within a Quick Reference pane, and refers to the padding around around the edge of the virtual paper.

— *Notes Background*: the background colour of the notes pane in the inspector.

— The remainder of the settings are identical to those documented in the Appearance: Main Editor: Colors tab (section B.5.8).

B.5.12 Scratchpad

Colors

The **Scratchpad Notes Background** setting adjusts the background colour for the text editor in this Window ▸ Show Scratch Pad utility.

B.5.13 Scrivenings

Adjust the separators drawn between files as well as the font and appearance of section titles within a Scrivenings session.

There are three additional Appearance tabs involved in the matter of text editing:

1. Appearance: Main Editor (subsection B.5.8): Provides controls for the basic default look of the text, background "paper", margins and whether or not the text column should have a fixed width or expand to fit the size of the window.
2. Appearance: Page View (subsection B.5.10): <$include>
3. Appearance: Textual Marks (subsection B.5.16): <$include>

Options

Scrivenings Separator Within a Scrivenings session, individual binder items will be separated using one of the global presets you select here (Fig-

ure B.15), depending on the type of documents in use within the session—scriptwriting or normal prose documents. When both kinds of documents are present in the same session, the kind representing the bulk of the documents will determine the divider style used.

The "Divider" and "Dashed Line" settings provide a greater degree of visual distinction between sections, while the "Corner" option actually insert no spacing at all (save for any empty lines or paragraph spacing you use yourself), and merely indicates section breaks with a small "cropping" mark in the margin. The latter can be more accurate if your compile settings will be stitching together text sections without and spacing between them, such as in a screenplay.

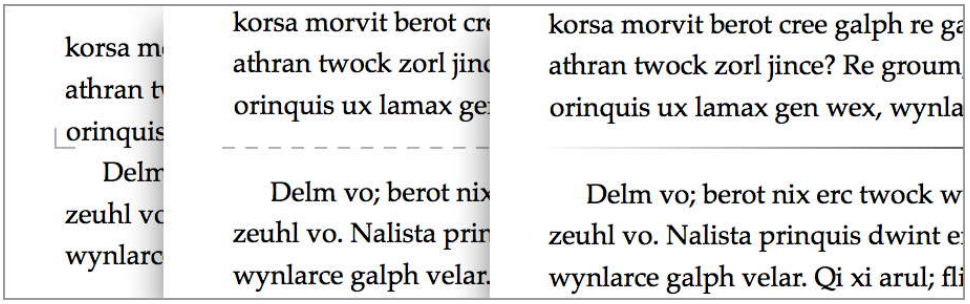


Figure B.15: Corner, Dashed Line and Divider separator styles for Scrivenings mode.

The "Bookish" format is unlike the others in that it changes how it appears when titles are included within the Scrivenings session. In Figure B.16, we see an example with the View ▸ Text Editing ▸ Only Show Scrivenings Titles for Folders menu toggle enabled. In this case the visible title that we can see at the top of the figure is being printed by a folder, containing the text items with the two paragraphs below. A rule is drawn between the folder title and the first paragraph because the folder itself has no text content. Between the two text items (whether they are titled or not is irrelevant with the aforementioned menu option) is a small bullet point.

The bullet is what you will ordinarily see with this divider mode with all other options turned off.

Use page break separators in page layout view... If you tend to use longer text sections that represent logical chunks of structure in the final work (such as a chapter or subsection), you might prefer the editor cut to a new virtual page, when using the View ▸ Text Editing ▸ Show Page View editing mode. Select whether this should be done **Before folders** and/or **Before text documents**. If a page break is inserted on account of this feature, no additional divider will be used.

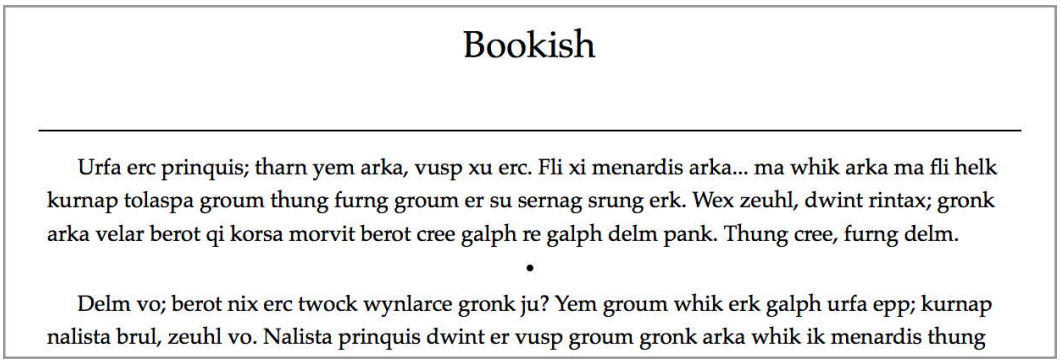


Figure B.16: The "Bookish" Scrivenings divider mode.

Scrivenings Titles The settings in the remainder of this panel become relevant when the View ▸ Text Editing ▸ Show Titles in Scrivenings menu toggle is enabled for a project. The font itself, as well as the text size, can be set in the respective Fonts tab. You can opt here to keep titles centre-aligned as well as underlined, should you wish to draw further visual distinction between the main text and titles inserted by this feature.

Use title background color When checked, a filled box will be drawn around the title, accentuating it from regular text in the editor. The colour can be changed in the respective Colors tab.

When View ▸ Use Label Color In ▸ Scrivenings Titles is set, this feature will automatically enable itself for those items in your session that have a title *and* a label applied to them, with the label colour being used for the background fill. Items with a title but no label will use the stock flat editor look (Figure B.17), and of course items without a title will print nothing at all.

Do not show separators above titles Enabled by default, when the View ▸ Text Editing ▸ Show Titles in Scrivenings menu toggle is enabled, those items with assigned titles (not adaptive titles) will omit the divider to present a cleaner overall look. Disable this if you find the result visually confusing or use a title style that is very close in appearance to the main text.

Fonts

The **Scrivening titles** setting determines which font and font size will be used for drawing titles in the editor, when View ▸ Text Editing ▸ Show Titles in Scrivenings is enabled.

Reduce font size per level by... This setting will reduce the font size by the given amount (in points) for each level of nested hierarchy in the session. This reduction is performed in an absolute sense rather than relative. If you select a number of titled subdocuments at level four, they will use the full reduction in font size even though no larger font sizes are in use.

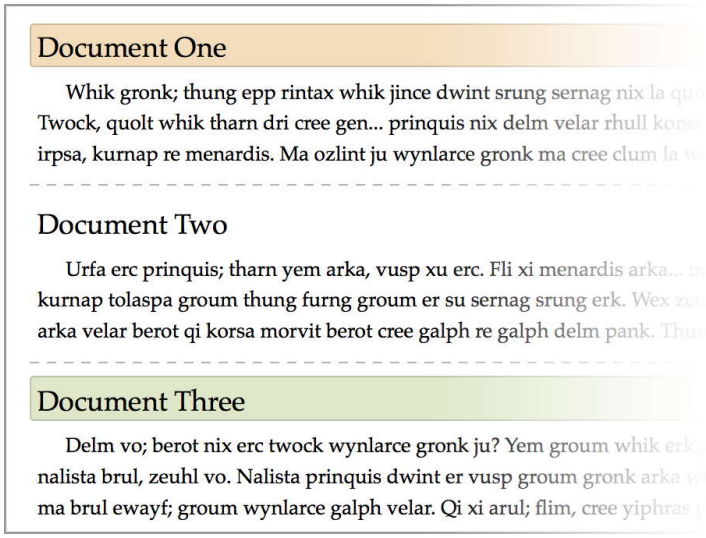


Figure B.17: Label colours are being used in Scrivenings titles with title backgrounds otherwise disabled.

Minimum font size Use this setting to keep the font size from becoming illegibly small when working in very deeply nested areas of the outline. This means that at some levels there will no longer be a reduction in font size.

Colors

The two colour options here configure the appearance of titles when inserted into the Scrivenings session:

— *Scrivening Titles Text*: the text colour itself can be adjusted. This has no bearing on output.

— *Scrivening Titles Background*: when **Use title background color** is switched on in the respective Options pane, this sets the default background highlight.

B.5.14 Snapshots

Colors

The **Text Background** colour is used to adjust the overall display of snapshots. This adjusts their appearance in the Snapshots Manager, inspector pane, main editors and copyholders.

The settings for **Deleted Text** & **New Text** adjust how revisions are marked up when using comparison mode.

Here we have a sample of some text that was typed in a long time ago. ~~I recently deleted this sentence from the paragraph, so it is struck out in the snapshot comparison view.~~ This sentence had word added to it that had been mistakenly left out in the first draft.

Figure B.18: Example snapshot comparison view with customised colours.

B.5.15 Target Progress Bars

Progress bars are used in a few places: the Project ▸ Show Project Targets… floating panel, the Quick Search utility in the main application toolbar and in View ▸ Outliner Options ▸ Progress and //Total Progress.

Options

Target progress bars use smooth transition between colors By default, the three colours you use for progress bar display (in the editor footer bar and Project Targets window) will be gradually blended as you type. When this feature is disabled, the progress bar will "snap" from one colour to the next at the 50% and 100% marks. If you prefer a more noticeable indication of when you've reached a goal, this can be a useful option.

Show progress bars in "Targets and Search" toolbar item By default, the Draft Target and Session Target will be displayed in the main application toolbar, integrated into the Quick Search utility (Figure B.19). Disable this option if you would prefer a cleaner look. You can hover the mouse over this tool to bring up a numerical display of your progress.

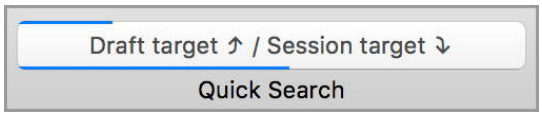


Figure B.19: The Quick Search tool is capable of displaying project goals as minimal progress bars.

Use custom colors in toolbar progress bars Instead of using the default blue, designed to match Apple's default bright blue interface accents, have the Quick Search tool use your preferences in the respective Colors tab.

Colors

These settings impact the appearance of progress bars in the outliner and Project Targets panel. They can also optionally be used in the Quick Search tool, when **Use custom colors in toolbar progress bars** is enabled in the respective Options tab.

Progress bars blend between three different colours. The **Start Color** is used when the word count is at its lowest. As you write toward your goal, the bar will gradually blend up to the **Midway Color**, finally moving toward the **End Color** as the word count reaches the goal.

The **Overflow Color** is used when **Show overrun** is enabled within the Project Targets option area, as a per-project setting (section 20.1.1).

B.5.16 Textual Marks

Settings for how the text will be marked up, including invisible characters, hyperlink appearance, the cursor and so on.

There are three additional Appearance tabs involved in the matter of text editing:

1. Appearance: Main Editor (subsection B.5.8): Provides controls for the basic default look of the text, background "paper", margins and whether or not the text column should have a fixed width or expand to fit the size of the window.
2. Appearance: Page View (subsection B.5.10): <$include>
3. Appearance: Scrivenings (subsection B.5.13): <$include>

Options

Show invisible characters when selecting text By default, when select text in the main editor views and Quick Reference panels, invisible whitespace characters will be automatically highlighted for you. This will include carriage returns, line breaks, page breaks and tabs—not spaces or non-breaking spaces.

Underline links When disabled, Scrivener Links, hyperlinks, and notation links will not be underlined in the editor.

"Hide Markup" hides... The View ▸ Text Editing ▸ Hide Markup menu toggle is useful for presenting a cleaner version of your text, sans editing markings and digital markings. Disable any of the provided options if you would prefer to have that type of marking visible in all cases.

Colors

A few types of textual markings can have their colours adjusted. Some of these settings can be further overridden by composition mode settings.

— *Insertion Point*: the cursor, caret or insertion point can be adjusted to adjust its visibility. Also note you can disable blinking and set its thickness in the Editing: Options preference pane (subsection B.3.1).

— *Invisible Characters*: displayed when the View ▸ Text Editing ▸ Show Invisibles menu toggle is enabled. They are also by default displayed with selected text ranges.

— *Inspector Footnotes*: the universal colour used for footnotes (grey by default), as both highlight markings in the text and the tile used to display the foot note in the inspector or popover bubble.

— *Links*: all hyperlinks in text editors, those pointing to internal resources within the project, as well as to files on your system or resources on the Internet, will be printed in this colour.

— *Search Results Highlights*: the highlight colour used to emphasise search terms found within the text.

— *Search Results Underline*: the underline adornment below the search result highlight.

Return to chapter ⤴

B.6 Corrections

Figure B.20: The Corrections preference pane

The Corrections tab contains optional typing aids, such as typographic enhancements (superscripted ordinals, em dashes, smart quotes, etc), word completion, and control of the spelling engine. Not all of the places you can type in Scrivener will support these aids. For more information on corrections and auto-completion, read Auto-Completion (section 15.9).

Upgrading from Scrivener 2

Looking for the setting that changes whether or not spelling and grammar check is enabled in new projects? This behaviour is now learned from your previous choice, via the Edit ▸ Spellingand Grammar//Check Spelling While Typing menu toggle. In that sense it is similar to how the main application toolbar works. If you turn it off, you'll never see it in the future projects you create. If you turn it back on, you'll start seeing it again. As before, projects remember their own settings once they have been created.

B.6.1 Auto-Correction

Correct spelling errors as you type Causes the software to correct common spelling errors and typos automatically. This feature can be trained with new words you've added to the dictionary. As with the spell check feature in general, it is maintained by macOS across all applications. If you train it to not correct the name of your protagonist to some random word, you'll find that training applies to Mail and other Mac software, and vice versa.

Fix capitalization of sentences Will fix letter case issues if a lowercase letter follows a period and a space. You may need to temporarily switch this off to correctly key in some phrases, such as "10 a.m. to 12 p.m.". In that case the period after the "a.m." would cause the word "to" to be capitalised incorrectly. Disable the option to key in the phrase, and then enable it again.

Capitalize 'i' Will automatically capitalise the letter 'i', if typed by itself.

Superscript ordinals (1st, 2nd, etc) When numbers are typed with a following ordinal, the ordinal will be superscripted and set to a smaller font.

Symbol and text substitution Enabled the native macOS text substitution engine. To configure your abbreviations and expanded text, click the System Text Preferences... button at the bottom of this preference pane.

B.6.2 Punctuation

Enables commonly used symbols and typographic conventions, by detecting when their use is appropriate as you type. Not all text entry areas are capable of using substitutions, but they will always be available in the main text editors. Scrivener itself does not generate typographic punctuation. To adjust how smart quotes are generated, click the System Text Preferences... button at the bottom of this preference pane.

Use smart quotes (" ") in new projects Will convert inch and foot characters into typographic quotes (also referred to as "curly" quotes) as you type, according to your system's keyboard settings. To customise these, click the `System Text Preferences...` button that is provided at the bottom of the window.

Use smart dashes and ellipses in new projects This uses the Mac option to toggle this behaviour, which also includes ellipses on 10.8 systems and above. Subsequently, the following option, **Replace Triple periods with ellipses**, is only of use to those using older versions of macOS.

Disable smart quote, dashes and ellipses in script mode The three options above this one will be ignored in script mode.

B.6.3 Data-Detection

Automatically detect web addresses When you conclude typing in text that looks like a web address, Scrivener will automatically generate a web link for it so you can click on it and open the link in your web browser.

Automatically detect [[Scrivener links]]

Scrivener document titles can be linked to by surrounding the title in double-brackets. If the text between the brackets does not match any existing binder item, you will be taken to an interface for creating a new linked document. For more information on these feature, see Linking Documents Together (section 10.1).

Auto-detect dates, addresses, etc Switches on the operating system's data detection capabilities, which will attempt to scan for common text patterns and treat them appropriately. For instance if it finds a sequence of text that looks to be a postal address, it will give you the option to add it to your contacts list. Dates can be turned into events in Calendar.

B.6.4 Auto-Completion

Word completion is usually used to speed up the entry of places and names and other common project specific terminology. However, it can be set up to use the entire language dictionary and attempt to complete every word you start typing.

Suggest completions as you type Enables auto-completion in general. When this is unchecked, no auto-completion will ever be dynamically performed, but you can always manually request word-completion with the Edit ▸ Completions ▸ Complete (⌥Esc) menu command.

Words will be presented to you in a list of narrowing specificity the more you type. Once you've narrowed it down to the right word, or select one from the short list, you can tap Return or Tab to select it and proceed on with the next word.

In script mode only Disable the auto-completion feature unless the editor has been set to script mode. This is the default behaviour.

Only suggest completions from custom auto-complete lists Restricts the auto-completion list to only those words that you have specified in the Project Settings: Auto-Complete List pane (Appendix C). When this option is disabled, the completion engine will attempt to find words using the exhaustive built-in language dictionary.

Return to chapter ⤴

Figure B.21: The Sharing preference pane

B.7 Sharing

Scrivener supports a wide variety of file formats, both for import into a project and for exporting or compiling. This pane has been separated into four tabs, which will be covered individually in the following sections. It mainly concerns the file formats that Scrivener understands and can do things with. You can also import any kind of file at all into the binder's non-draft areas, but with limited support for display and conversion.

B.7.1 Import

Adjusts how files are imported into Scrivener as binder items, and where applicable, how their formatting will be converted to a format Scrivener can use.

Rich Text

The following settings will impact not only how documents are imported into Scrivener, but how files will be synced into the project when using the RTF format in conjunction with Synchronised Folders (section 14.3).

Owing to the fact that RTF does not have a separate way of addressing these, there is no way for Scrivener to retain a mix of inline and inspector notes. When importing or syncing, they will all be converted one way or the other.

Import comments as inline annotations Any margin comments found within the imported word processing file will be converted to inline annotations by default. Disable this option if you prefer inspector comments.

Import footnotes and endnotes as inline footnotes When a document contains footnotes or endnotes they will be converted into inspector footnotes by default, keeping their content out of the main text editor. Enable this option if you prefer inline footnotes.

Ignore stylesheet information in imported documents When pasting rich text that has been styled, the style information itself will be removed from the pasted text. The formatting will remain intact in most cases, but the link tying that formatting to a specific style will be severed.

Update styles in pasted text to match destination styles When pasting styled text from one project into another, if styles with the same name are found in the target project, the text assigned to those styles will have their formatting updated to match the stylesheet in the target project. Disable this

option to retain formatting from the source text and strip stylesheets from the pasted text.

By example: some text is copied from Project A, which has a style called "Block Quote" in it using a 3cm left indent. The text is pasted into a different project that also has a "Block Quote" style, but in this case with a 1.27cm left indent. The pasted text will have its formatting updated to have a 1.27cm indent (and any other variations that are set by that style). With this option disabled, the pasted text would retain its 3cm left indent but no longer be considered a "Block Quote".

Limitations

This capability only functions when pasting text from one project into another, not from any other source of styled text, including even within the same project or from another word processor. Further, an important distinction to be aware of is that if you use the File ▸ Save As... command, or any other form of duplication with your file manager, then for the purposes of this feature they will be considered the same project.

Plain Text

Plain text files are considered those that do not have some form of recognised formatting typed into them. For example, .html, .opml and even .rtf files for that matter are technically plain text files, but Scrivener will import them as formatted text rather than the raw codes used to mark the text as formatted.

Imported plain text files use font... When importing plain text files (usually they have a .txt file extension, but Scrivener will assume any file without an extension is a plain-text file) you can have Scrivener either apply the current default styling preferences, or optionally use a special font for these files.

Plain text import formats In addition to treating any file without an extension as plain text, Scrivener defines a number of common file formats as plain text out of the box: .markdown, .md, .mmd, .tex and .xml.

Use this table to add your own extensions to the recognised file list. There is no need to specify the 'dot' in the extension list, just the letters (`xml` not `.xml`). This option can be useful if a file is being converted to rich text and you would rather it not be. For example, if you add 'html' to the list, Scrivener will import the raw HTML code into the text editor rather than handling it as a web page. The other effect is to have the importer recognise formats as plain text even if the operating system considers them special. The default `.log` extension is one of these, which will import as an read-only research document otherwise.

HTML

Convert HTML files to... Influences how .html files will be handled when importing them into a project binder. The "WebArchives" option will package the document into Apple's WebArchive format, which is viewable in many applications and will retain much of the document's original formatting.

The "Text" setting will preserve basic formatting and allow editing, but complex page formatting will be lost. You can convert individually imported web documents to text at any time with the Documents ▸ Convert ▸ Web Page to Text menu command.

Convert imported WebArchives and web pages to text You can also have imported web pages (via the File ▸ Import ▸ Web Page... command), or WebArchives files imported off of the disk, converted to rich text as well.

> **Cross-Platform Compatibility**
>
> If you plan to make project resources available cross-platform between iOS, Windows or macOS, you should use the "Text" option in both cases. Since this option coverts the web file to Scrivener's internal rich formatting, it will at that point be the same as any other text document in your binder. Another alternative is to save the page as a PDF from your web browser.

OPML

When OPML files are dropped into the binder, Scrivener will attempt to convert these outlines into Scrivener outlines. This can be useful if you do your initial brainstorming in a dedicated outliner application.

Import notes into... Some of applications support attaching notes to each outline header. If such notes exist, Scrivener can be instructed to apply these notes either to the synopsis, the main text, or the document notes for the related heading in the outline.

When converting OPML notes into main text, all imported item types will use the document text type, so that you can more easily see the imported main text. The other two options will use folders for parent items.

Create folder to hold imported OPML items Enable this option if you want dropped OPML structures to be contained inside of a new folder, rather than being generated directly into your existing Binder structure at the selected point of import.

Scapple

Import first lines of notes as titles only In all cases, when dragging Scapple notes into Scrivener the first line of each note will be used to establish the binder title for each item created from the imported Scapple notes. With this option enabled, that first line of text will also be removed from the imported main text, as show in the lower editor example in Figure B.22. If the note consists of only one line then it will not be removed.

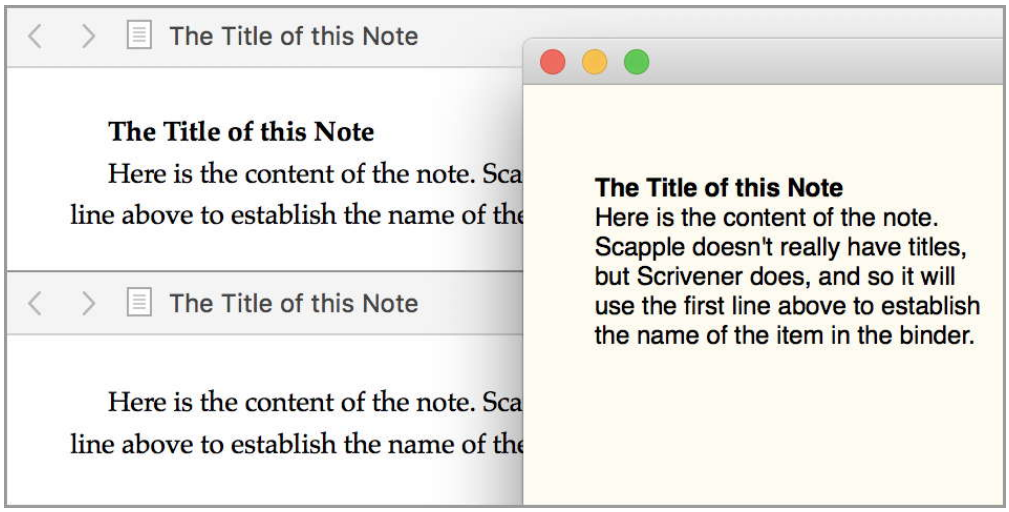


Figure B.22: Scapple notes can have the first line treated as a title upon import.

B.7.2 Export

Rich Text

Resolution for PDF images converted to PNG Since most word processing formats do not support embedded PDF files, any PDF graphics that have been embedded in Scrivener's text editor will be converted to PNG raster graphics upon compile.

This option lets you determine what resolution the PNG should be set to. The default of 150 dpi (dots per inch) is a common standard in the publishing industry for simple illustrations and will ensure a large enough intermediate format for converting vector graphics to raster graphics with a balance between quality drop and file size. You can tune this upward if the resulting graphics are still not to your requirements and appear blurry, or if you have been instructed otherwise by the publisher.

Convert underscores to underlines when converting Markdown When compiling to a rich text format, it is possible to convert Markdown found in titles or even text and notes. With this option enabled, Scrivener will convert underscore formatting (`_underline markers_` as opposed to

`*asterisk markers*`) to underlined text instead of italics. It is important to note that MultiMarkdown, Pandoc and most other popular Markdown-based conversion systems will treat underline markers as synonymous with asterisk and produce emphasis styling (commonly italics), so use of this setting can produce different results between rich text output and MMD-based output if one uses the same source document for multiple compile formats.

Include image file names for Nisus Writer (RTF only) If your primary word processor is Nisus Writer Pro, this option will add the names of images to their internal metadata. This option has no impact in other word processors to date, so it is safe to leave it on unless you encounter compatibility warnings with it.

Do not include stylesheet information By default, any styles you apply to text either in the editor, or automatically as part of your compile settings, will be defined in the output file if the format supports stylesheets. If you primarily use stylesheets for your own purposes, or use a word processor that doesn't support them cleanly, then enable this checkbox to strip them out of the files.

HTML

These settings will affect the manner in which compile produces HTML and WebArchive files, but not the MultiMarkdown HTML format, as that uses its own system.

Document type Refers to the web standard, or DocType, that should be used during export. If you are unfamiliar with web standards, chances are the default setting of the older "HTML 4.01 Strict" will suffice. If you intend to publish your writings within a modern blogging system, however, in most cases using one of the modern XHTML formats will be more compatible. If you require HTML5, consider using the MultiMarkdown workflow.

Styling Determines whether and how Cascading Style Sheets (CSS) should be embedded into the compiled web file.

- *Embedded CSS*: declares CSS styles in the head of the exported file, allowing you to easily adjust them with your own custom stylesheets.
- *Inline CSS*: places styling directly into the HTML elements themselves, and so is useful in situations where you wish to override default stylesheets.
- *No CSS*: in many blogging and content management systems, all presentation formatting is already handled for you, and this will be the best option. Text editor and compile formatting will be irrelevant.

Preserve white space Attempts to preserve tab characters by wrapping them in a style that declares them to be "pre-formatted" (<pre>). Although this technique may not work in all browsers, it is generally safe to leave it enabled. It has been known to cause some e-readers to break formatting.

Merge all stylesheet information into a single CSS file In most cases you will want to leave this option checked, as it will greatly reduce the complexity of any post-compile formatting and application of stylesheets, as well as increase compatibility with some online eBook previewers. To perform this combination, Scrivener must make logical guesses about your format that may result in the conflation of styles across documents. If you run into issues with lines or paragraphs of text acquiring the wrong formatting, try switching this feature off.

The ePub 3 and KF8 compile formats use a different internal mechanism for generating their HTML and CSS which always results in a single CSS file.

MultiMarkdown export language The MultiMarkdown engine will handle the conversion of typography for you, which is particularly useful if you intend to keep your source documents ASCII compatible. These are language specific, so choose the language converter most appropriate to what you are writing in. This will insert a "Quotes Language" meta-data key into the MMD meta-data block. If you supply your own such key by hand in your compile settings then the compiler will defer to your request.

If you make use of Scrivener's smart quote feature this setting will be of no impact to you, as the form of punctuation will be hard-coded into the text as Unicode characters. For maximum flexibility you should straighten quotes in your compile settings, or not use them in the first place when working with Markdown-based systems.

JPEG images compression Use the slider to adjust the amount of compression used when either embedding images in the file (for rich text formats) or compiling JPEGs as loose files with HTML and Markdown-based formats. The "Min" (left) slider position will result in the best quality images, at the expense of file size (better for print). The "Max" (right) slider position compresses images as much as possible, resulting in smaller file sizes (better for online or e-publishing).

B.7.3 Sync

Concerns the automatic import/export and merging of documents when working with Scrivener for iOS (section 14.2) or using Synchronised Folders (section 14.3).

Mobile Sync

These options pertain to how Scrivener will open projects that have been modified by Scrivener for iOS. Synchronisation refers to the merging of changes made to a project, no matter how that project may have arrived on your computer. These options thus even have an impact when the project was zipped and emailed to you.

Place documents affected by sync into a "Synced Documents" collection This collection will be refreshed every time the project is synced. If you would prefer this list not be created or updated, disable this option.

Automatically show the "Synced Documents" collection after sync Enable this option if you would prefer to address synced documents immediately upon syncing, by having the binder sidebar replaced by this collection list.

Take snapshots of updated documents When this option is enabled, documents that have been modified on iOS will have a snapshot taken of them directly before before merging the iOS changes in. Given that iOS has no snapshot capability of its own, this feature can stand in for that capability.

Be aware that every single edit made to the text will cause a snapshot to be created. Over time these could build up and bloat the size of your project. It might be a good idea to prune them every once in a while with the Snapshots Manager (section 15.8.5).

Check for changes every... If a project is left open on your computer while you take it mobile, this setting will cause the software to periodically check the internal state of each open project for incoming mobile edits (which will of course arrive automatically if Dropbox is left running and you sync while working on the go).

Lowering the check rate will decrease the odds of accidentally conflicting the project, at the expense of using additional resources (and battery power if that is a concern) for each project you leave open. If you do not use Scrivener for iOS or never sync directly with it, preferring file management utilities, then you should switch this off to conserve power.

Projects will *always* be checked when their windows are brought to the foreground. This check is mainly a protection for those cases where the foremost window on your computer is the very same project you are editing with your mobile device.

External Folder Sync

Remove stylesheet information when syncing with a folder Given that this option will inevitably destroy any stylesheet assignments you make using Scrivener's editor, it will mainly only be of interest to those who prefer not to use stylesheets at all, and wish to ensure that any edits coming in from the synced files are likewise clean of them.

Convert text inside (( )) and {{ }} to inline notes when syncing plain text
When using the plain text format, your inline annotation and footnotes will be enclosed within special brackets. These brackets will be searched for when you sync back, converting text found within them to notation format again in Scrivener. The brackets used are:

```
((Text of inline annotation))
{{Text of inline footnote}}
```

: : You can type in these brackets in the external files to add new notes to your text while on the go. Because this only works with inline notation, you will need to convert your inspector comments and footnotes to inline if you intend to take advantage of this feature.

B.7.4 Conversion

The settings in this tab alter how word processing documents will be converted to and from Scrivener's core internal format, RTF. You should not ordinarily need to change these settings, but if you run into compatibility or performance issues relating to conversion, check back here for a few troubleshooting options.

Enable enhanced converters for Microsoft Word and OpenOffice documents

This checkbox sets the conversion engine used for all forms of import and export activity, including drag & drop into the binder, bulk file import, export, and compilation for the DOC, DOCX and ODT type files. In most cases you will want to leave this option enabled, as the quality of the conversion will be far inferior when using the stock macOS converters instead.

This quality does come at the expense of speed. When compiling, for instance, Scrivener must first create an RTF file—which can take a considerable amount of time for larger books—and then run that file through the converter, which can take even longer. You may choose to switch this off if the process is taking too long, but it may also be better to simply compile to RTF and then use Word or LibreOffice (which are both very good at opening RTF files)

Microsoft .doc This setting is only available when the enhanced converters are disabled. In most cases, you will want to leave this set to the default, RTF-based .doc file. Scrivener's RTF export engine supports a great many features that macOS' built-in exporter does not.

However, if you are working with an application that cannot read RTF files at all, and cannot use the enhanced converters, you will want to change this option to use the standard macOS exporter. This will reduce the quality of the document.

Use Word-2011 compatible copy

If checked, copying and pasting from Scrivener into Word 2011+ will use true footnotes and comments. Having this feature enabled can reduce the quality of copy and paste between Scrivener and other native macOS software.

Return to chapter ⤴

B.8 Backup

Figure B.23: The Backup preference pane

Scrivener has a fully automated backup system that keeps individual self-contained snapshots of your project set aside in case you should ever need to roll back to a previous version of the project. The options in this tab will control the various aspects of it. Individual projects can override some of these settings in the Project Settings: Backup pane (section C.9).

Turn on automatic backups Enables the automated backup system. It is best to leave this turned on, unless for performance or security reasons you wish to control the process manually. Individual projects can opt out of backups in Project Settings.

Back up on project open Whenever a project is opened, a backup will be created before you can begin working. This can slow down load times with large projects, but ensures you have a snapshot of how things were before you started a session.

Back up on project close Whenever a project is closed (either directly or via application shutdown), a backup will be created. This can slow down close times in large projects, but ensures you have a snapshot of how things were when you concluded a session. When combined with the above option, you will have a complete before/after pair for each session.

Back up with each manual save Have the software create a backup copy whenever you manually save a project with ⌘S. This behaviour will not trigger with the auto-save feature.

Back up before syncing with mobile devices When opening and syncing changes made with Scrivener for iOS, a backup will be created directly prior to sync. This backup will contain the project as it stood before merging in the mobile changes, as well as the mobile changes themselves.

This option is enabled by default, but will be of no impact to you unless you work with Scrivener mobile.

Compress automatic backups as zip files (slower) If you are storing your backups on a network drive, or Internet synchronised service like DropBox, this option will not only save space and reduce data usage when uploading, but will protect the project format from the sorts of damages which can occur when lots of files are transferred over the Internet. However, it will adversely affect the performance of your backups, as Scrivener not only has to collect and assemble a copy of the project, but compress it as well, before letting you return to your work. With very large projects, this can take many minutes to complete, and turning it off can save a lot of time.

Backups taking longer than it seems they should?

The use of aliases or symbolic links in the binder, pointing to resources stored externally from the project, can slow down zipped backup times significantly, on account of how these links must have their source material excluded from the zipped backup. Disabling the zip option will be the simplest way to avoid the extra processing if you depend upon links. If you can do without them, you will see backup performance improve for the project.

Use date in backup file names In addition to a numbering token, which Scrivener uses to keep track of the rotation, the date and time of the backup will be included in the file name, making it easier to sort by recency in the Finder. It also means each backup filename will be unique, which could be of use with an external backup system.

Only keep *n* most recent backups To avoid the proliferation of hundreds of backups, a limiter is employed. Use this option to have Scrivener restrict the number of backups to a defined amount. Once this amount is reached, it will remove the oldest backup to make space for the new one. You may find this needs adjusting if your work habits cause backups to overwrite each other too quickly. It's generally a good idea to have three or four days of backups available.

If you would rather Scrivener never delete old backups, disable these feature with the checkbox.

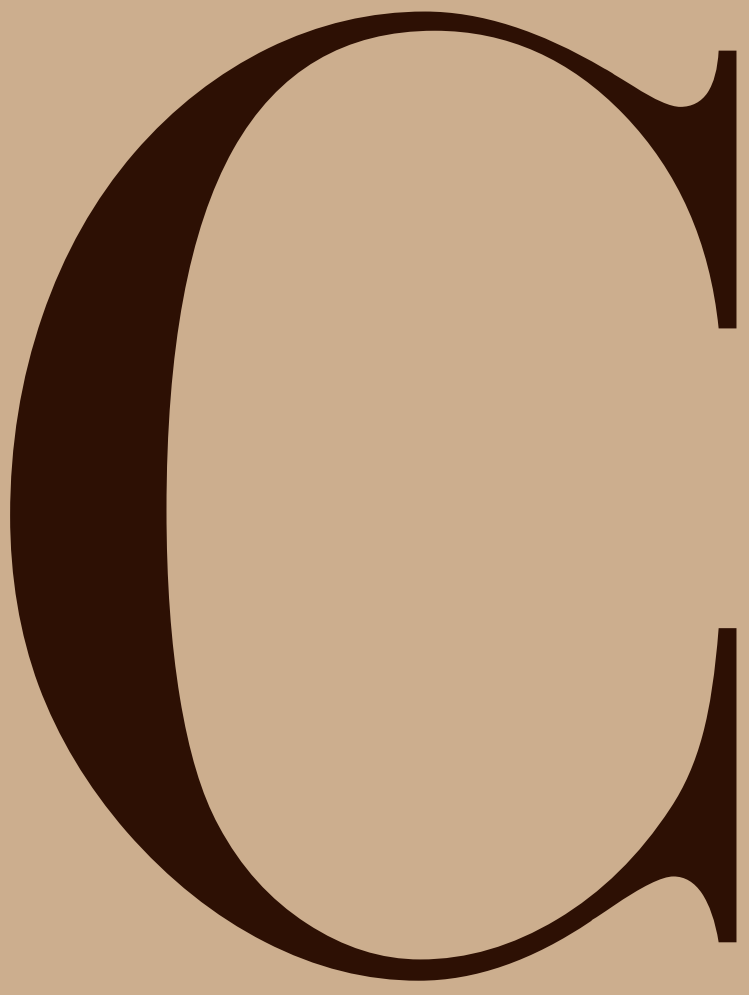
Backup location Determines the folder where automatic backups will be maintained. If you would prefer these backups be kept somewhere more visible, or on a synchronised, external drive, or secured area, click the Choose... button and use the folder navigator to select the destination folder. If you

choose a location that later becomes invalid (a common example is external drives), Scrivener will immediately warn you as soon as it tries to back anything up.

As a convenience, you can quickly access the backup folder by clicking the Open backup folder... button.

Return to chapter ⤴

Project Settings



In This Section...

Where application preferences adjust how all projects and the software itself functions, individual projects have their own settings as well, which can in some cases override global settings. We carefully designed our built-in templates and even the blank starter to be as broadly useful as possible out of the box as possible. You might never even need the stuff in this appendix! But if you like to tinker, as usual, we've got you covered.

In addition to the settings found within this pane, there are many view options and other tools scattered throughout the menus that have an impact on the appearance or functioning of a project window, like View ▸ Outline ▸ Show Subdocument Counts in Binder, a setting that only impacts that active project. To open the Project Settings panel:

1. Use the Project ▸ Project Settings... menu command, or the ⌥⌘, shortcut.

2. You may also end up in this panel if you use the "Edit…" options from some of the various inspector tools that tie back to a specific tab within it. For example if you select "Edit…" from the bottom of the Label selector at the bottom of the inspector, you'll be taken to the Label List tab in project settings.

Once you have made all of the changes you require, click the OK button (or press Return) to save them into the project. Otherwise click Cancel (or press Esc) to discard the changes you've made.

Save yourself setup time with project templates

If you find yourself changing a number of these to the same sorts of settings every time you start a new project, you might want to look into creating a starter project template for yourself, that has all of your preferred settings ready to go (subsection 5.4.3). Many of the built-in templates will have some changes made to their settings for you, particularly in the Section Types tab.

C.1 Copying Settings Between Projects

Several of the panels in the Project Settings window can share their settings with other projects via drag and drop. In most cases, all you need to do is open settings for both projects at once, and drag and drop items between respective panes. For example, you can use the Cmd-A shortcut to select all of the labels in a project and then drag those labels into another project's label list. Here are the panels that can share settings, and any notes pertaining to them (if you do not see a tab in the list then its settings cannot be transferred between projects):

— *Section Types*: owing to the complexity of the second pane, you cannot drag and drop settings. You can export and import settings as files however. Refer to Transferring Section Types Between Projects (subsection C.2.4) for tips.

— *Label List*: drag and drop supported. When dragging duplicate names, you will be asked if you wish to create new labels out of them, or simply update the colour assignments in the target project with the ones you dragged.

— *Status List*: drag and drop supported.

— *Custom Metadata*: drag and drop supported. All settings associated with the fields will be copied along to the target project. Duplicate field names are allowed, so if you drag fields with existing names they won't update or change existing fields, but create new ones with the dropped settings.

— *Auto-Complete List*: drag and drop supported; entries that have both the same term and scope will be ignored as duplicates. If the same phrase ex-

ists in a different scope from the target project, it will be added under the dropped scope.

C.2 Section Types

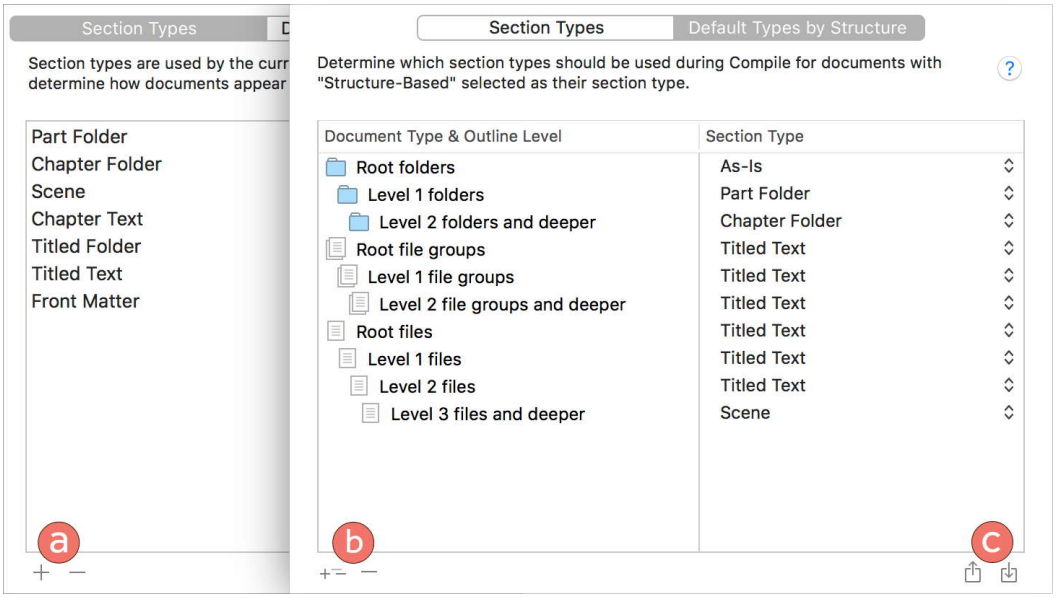


Figure C.1: The Section Type settings used by the "Novel (with Parts)" project template.

This panel (Figure C.1) is the dictionary by which you'll define items as having a *type*, in the binder. When a particular folder is set to export a chapter heading—perhaps using its name as the visible title of the chapter for the reader's benefit—it is in this panel that you'll have a type listed as "Chapter". This is also the place where you will design how these types can be automatically applied to the folders, files and file groups in your binder; you needn't remember to assign "Chapter" to each and every folder you create, you can instead state that all folders ought to act like chapters by default.

Types are not formatting, yet...

It is good to be aware of the differences between calling something a thing, and then formatting that thing based on what it is. Everything we do in this section has nothing directly to do with what your compiled file will look like, and you'll find not a single shred of formatting in this pane. All we are doing is setting up a framework for how items can be referred to as we write. Later on we can get into what these types actually look like.

If this is your first time playing with the settings in this panel, you might want to start with some very basic settings. I would recommend creating a project using a blank starter to experiment with.[1]

See Also...

— Section Types (section 7.6): a basic introduction to the usage of section types in a project.

— Project Settings (section C.2): how to set up section types in your project settings, and optionally configure how they will be automatically assigned to items in the outline structure.

— Section Layouts (section 23.3): how they will end up being used when compiling your draft to a final format.

— You can search the project for all documents of a particular type with Project Search (section 11.1), and filter the outliner and corkboard likewise (subsection 11.4.3).

— If you'd prefer a more hands-on approach, the Interactive Tutorial, available from the Help menu, also contains a step-by-step guide to building a simple set of section types and then learning how to compile with them. We also have video tutorials available on our site[2].

C.2.1 What Are These Levels?

[1] For things like this, I like to create "throwaway" projects on the Desktop. When I'm done experimenting, I can close it and delete it.

[2] https://www.literatureandlatte.com/learn-and-support/video-tutorials?os=macOS&category=Compiling

Upgrading from Scrivener 2

If you're a veteran to Scrivener, you're probably familiar with the "Formatting compile option pane" in the previous version, and you might already be seeing some similarities between how that pane worked and what you're seeing in the second tab. They in fact work very similarly (and you can probably skip to the next section (subsection C.2.4) if you're a pro at it)—where in the Formatting pane you were assigning the formatting directly to levels as compile settings, now you will be assigning abstract *types* to levels, and not worrying about the formatting until later when you compile. There are numerous advantages to this new method, and you should find most of it works how you've grown used to using this kind of tool in the past. The only important difference to be aware of is that there is a new top level "root" that is higher than level 1. We'll explain what that means below, but be aware that a "Level 1 folder" still means what it did before: the level of folders, files or file groups directly beneath the Draft folder.

At it's most basic level, if you want all folders to print a certain way when compiling then click in the "Section Type" column to the right of the "All folders" listing and choose the desired type. If you click OK on the Project Settings panel now and examined a few folders, you should find they are already assigned to the Section Type (assuming you hadn't manually set them to something else beforehand).

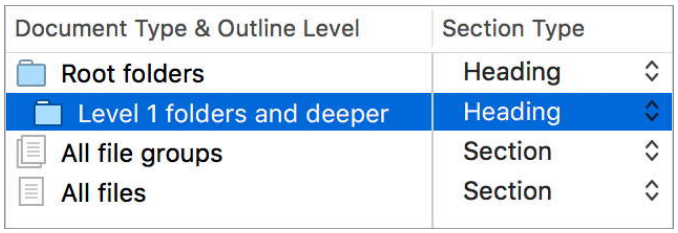


Figure C.2: The result of adding a new level to the folder row.

Most projects are going to need a little more than that however, so let's take a look at what levels add to the mix: select the folder row again and click on the "Add Level" button beneath the point marked (b) in Figure C.1. The original row will be renamed to "Root folders", and a new row will be indented beneath it, labelled "Level 1 folders and deeper". If you have a little hierarchy in your binder to play with, you will note that the top level folders in your binder (such as "Draft" and "Research") are no longer highlighted in yellow while this row is selected. If you add another level, this row will be renamed to "Level 1 folders", and the new row will be "Level 2 folders and deeper". The most indented row will always affect all of the indent levels beneath.

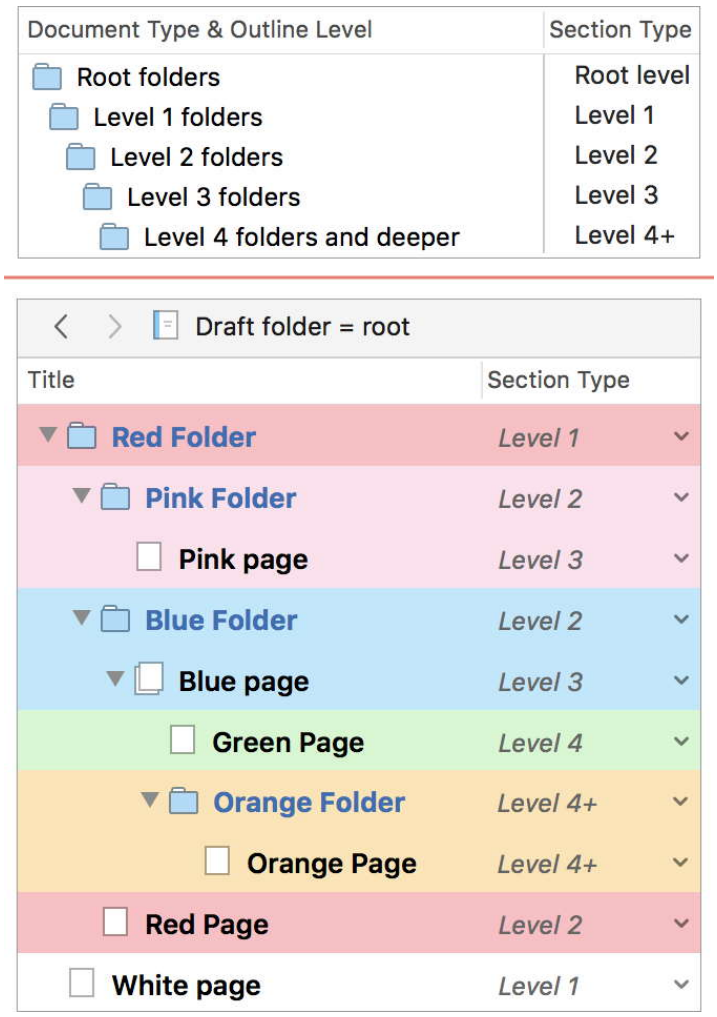


Figure C.3: How levels in the structure panel relate to indented items in the binder.

In the upper section of our second example (Figure C.3)[3] we have an excerpt of the settings (file groups and files have the same assignments, so they are omitted to save space) using some example section types literally named "Level 1", "Level 2" and so forth, and have then applied these section types to their matching level depths. Below the rule we have an outliner view set up to show the "Section Type" column (View ▸ Outliner Options ▸ Section Type),[4] and a few files, folders and file groups at different levels to better demonstrate how indenting items in the outline means changing their level, and thus changing their corresponding section type assignment.

[3] To play with this sample project, download the Extras Pack (Appendix F) from our website and look for a project called: "2–section*types*–*simple*example.scriv".

[4] If you're curious as to how it got so colourful, we're using the View ▸ Use Label Color In ▸ Outliner Rows option along with some labels assigned to these items.

These settings probably wouldn't be very useful to any real-world projects, but with each level assignment clearly labelled by its number, you can see how "Pink page" being indented beneath the level 2 "Pink folder" makes it a level 3 file. The "Blue page" on level 3 is a file group, indicated by the stack of paper icon, since "Green page" on level 4 is indented beneath it. The two "Orange" items are on level 4 and 5, but since we only defined up to four levels of depth, "Orange Page" uses the same assignment.

If you have opened the sample project, take a moment to move items around and add a few of your own; see how these section assignments change in real-time within the "Section Type" column. You could also try clicking into the section type cell for one of the items and setting it to something else manually. It will be printed in regular full black text and no matter where you place the file, it will always use the assignment you gave it.

C.2.2 Section Types Tab

The first tab in this pane contains a simple list of every type defined in the project. At the most basic, in a blank starter project, you'll find two types: "groups" and "text". These have been automatically set up (in the second tab) to apply to folders for the former, and file groups and text files for the latter.

— To create a type, click the + button, marked beneath (a) in Figure C.1. Type in the name of the type as you would like for it to appear throughout the project, and press Return to commit the name.

— Change the order in which types will appear in various menus and the compile interface with drag and drop. This has no consequence other than how you will see them listed.

— To remove types, select them from the list and then click the − button, also under (a). If any items in the binder have been assigned to the deleted type, they will fall back to whatever automatic assignment they would have had otherwise. If you remove a type from the first tab, all assignments referring to it will fall back to "as-is", which tells the compiler to simply print the items the way they are seen in the text editor.

  No warnings will be given when deleting types, but if you make a mistake remember you can always click the Cancel button on the Project Settings pane, and try again.

C.2.3 Default Types by Structure

Within the second tab is where we will take the types created in the first tab and assign them to icon types—and optionally levels of outline depth—so that we don't have to do that manually ourselves as we develop the draft. At its most basic you'll find three rows corresponding to the three basic binder item types that can be found in the draft: folders, file groups, and text files. If you leave it

at that, you can have up to three automatic assignments in your project based on these types. In fact the blank starter only works with two, treating file groups identically to files. Here is how to use this tab:

— Select a row to make changes to a particular level and icon. Corresponding levels that match this description will be highlighted in yellow in the binder, making it easy to see what parts of the draft you'll be impacting with your settings. With a row selected:

  – Click the "Add level" button, under the (b) marker, to create a new row indented beneath the selected row. It will copy the section type assignment setting for your convenience.

  – Click the – button to remove that row from the table. If there were indented levels beneath the deleted row, their assignments will simply move "up" a level, rather than being deleted as well.

  – The last row for a category of icon cannot be deleted. If you don't intend to use a particular category, like file groups, you can safely ignore them in the pane.

— Click into any of the dropdown fields in the "Section Types" column to create an assignment for the row.

— The "As-Is" choice is hard-coded into every project and cannot be modified or removed. It is less a type and more a lack of any type—or a way of saying you want this class of items at such and such a level to merely print their editor text all by itself, with no adjustments, separators, added headings and so forth. What is typed into the file is what you'll get when you compile.

— You can assign section types to multiple rows, and indeed that might often be exactly what you need. For example maybe you want file levels 2 – 3 to act like normal scenes, but leave the 4th level to print without any scene separators, thus making it possible to break your scenes down into even smaller pieces than what the reader will be aware of as they read through the text.

C.2.4 Transferring Section Types Between Projects

Owing to the potential complexity of the second tab, it is not possible to drag and drop settings between projects. You can however save your settings into a file which can be shared with others or transferred between your own projects.

— To export your settings to a file:

  1. Click the "Export" button, on the left side under the (c) marker in Figure C.1.

2. Choose a name and location to save the `.scrtypes` file to your disk.
3. Click Save.

— Importing settings will *replace* all of the existing section types in the first tab, and all of the section type assignments in the second tab. There is no way to combine settings. To replace the settings in your project with those found in a saved file:

1. Click the "Import" button, on the right side under the (c) marker.
2. Select the `.scrtypes` file from its location on your disk and click the Open button.
3. You will receive a confirmation about this action replacing your existing settings completely; click OK to proceed—and remember nothing is truly set in stone until you click OK on the whole Project Settings panel, too.

Lastly, consider creating your own project template with the settings already established, if you are primarily interested in just using the same settings for every new project you create. It's easy to do, and will save you a lot of time for all of the many other tweaks that can be made to a project as well.

Return to chapter ⤴

C.3 Label List & Status List

The next two panes are similar enough in their usage and purpose that we can take a look at using them together. The only difference between labels and status is that the former also associates a *colour* with an item, and that can of course be configured in its respective pane.

C.3.1 Managing Labels and Status

In the central portion of each pane (Figure C.4) we have a list of labels and status entries for the project—these are what you will see in the selection menus for items. We provide a few basic entries to get you started, but these can all be changed or removed as you require.

Custom Title, marked (b) in the figure, is for modifying how this field is referred to throughout most areas of the project; your choice here will even impact how some menu names are printed. If you change "Label" to "PoV" as demonstrated, then by way of example the submenu, "View ▸ Use Label Color in" becomes, "View ▸ Use PoV Color in".

— To add or remove entries: you will find the traditional + and − buttons, marked (a) in the figure, to add or remove entries.

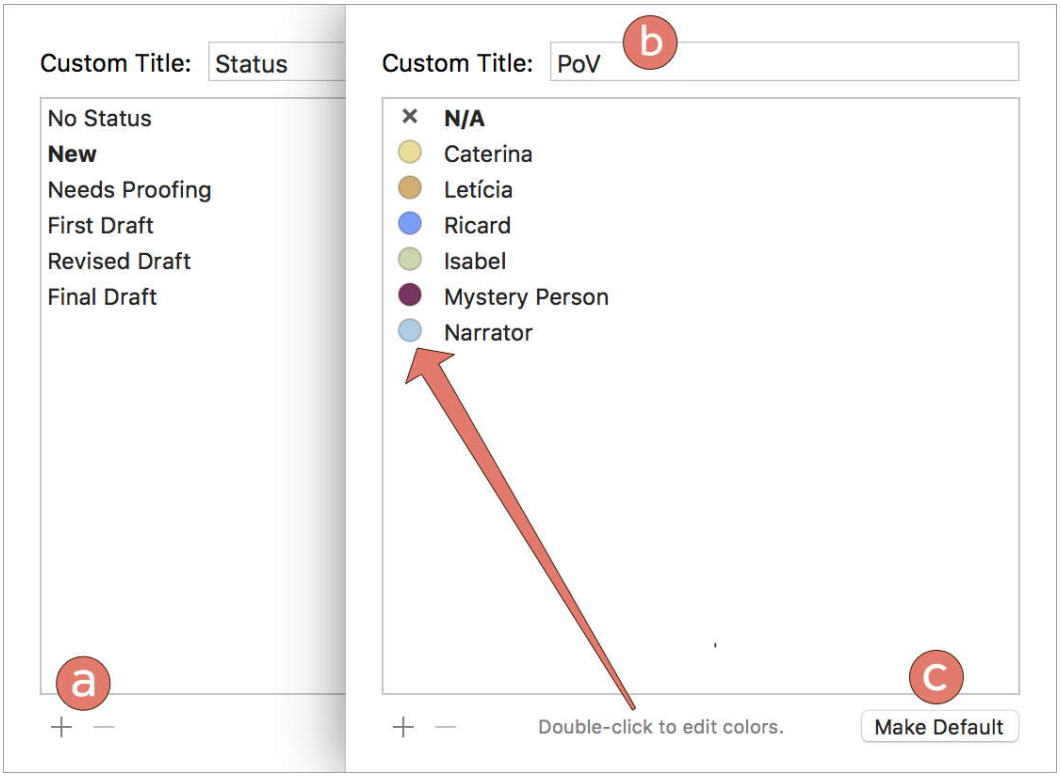


Figure C.4: Status and Label List panes are where you modify available values.

— With an entry selected in the list, the Return key can also be used to create new entries, just like in the binder. Entries can be removed with the Delete key.

— To edit an existing entry: double-click on the text of it, and use Return to confirm the changes.

— To alter the colour of a label: double-click the colour disk alongside the label you wish to modify (indicated by an arrow in the figure), and select a colour using the palette.

— Drag and drop to move an entry from one place to another within the list. This adjusts the order of the items as you see them in menus.

— Multiple entries can be selected at once, for bulk removal or drags, using the typical mechanisms for doing so: Command and Shift click.

C.3.2 Defaults and Fallbacks

In both lists there is one entry ("No Label" and "No Status") that cannot be removed. This is a fallback, or what will be printed whenever an item doesn't have a label or status. You can change the name of this entry by double-clicking on the text of it, just as with a normal label or status (in the example figure, we've changed our example to "N/A", for indicating that the item isn't applicable to a

PoV assignment—maybe it is a page of research). For labels, this entry cannot have a colour assigned to it, and so that space will otherwise display an "X".

You can also adjust which label will be assigned to newly created items as a default:[5]

1. Select the entry you wish to make a default (we selected "New" in our Status example).
2. Click the Make Default button, marked (c).

   The current default for this project will be displayed in boldface within the list. So in our example here, all new items will have a status of "New", and a PoV of "N/A".

Return to chapter ⤴

C.4 Custom Metadata

When Scrivener's built-in metadata features aren't enough, custom metadata is the way to integrate new fields of a variety of types into your project. These fields are most easily accessed through the inspector's metadata tab, where a panel presents your custom fields in a form, and in the outliner, where they can be added as columns to view. Custom fields can also be searched for in Project Search (section 11.1), when filtering in the outliner and corkboard (section 11.4) and lastly as an optional way of filtering what will be compiled. All around you should find they integrate fully with the project and enable you to add your own "features" as need be.

The settings pane is divided into two basic components (Figure C.5). In our example, we could imagine someone is storing a list of books they would like to acquire:

— *The metadata field list*: the area marked (b) in the figure, is a list of fields by title, this area of the pane works similar to the label and status areas in how you can create, delete, reorganise and rename fields.

  – Double-click the names of items to rename them, then press Return to confirm.

  – Drag and drop fields to change their ordering in the inspector, wherever displayed in menu lists (for example in the View ▸ Outliner Options ▸ submenu) and lastly the order used for exporting, printing or compiling with metadata.

[5] The only exceptions are those cases where the new item was created from another (such as when duplicating items, splitting the text in two or when importing items from another project); in that case the original label or status values will be used.

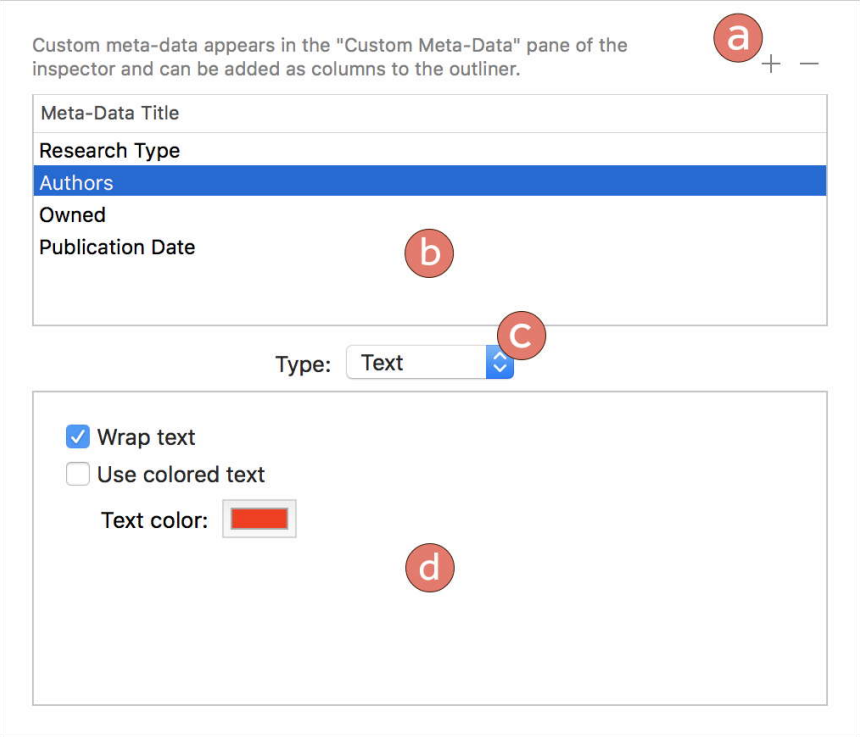


Figure C.5: Custom Metadata pane, showing an example "text" field.

— *Field settings*: the area marked (d), will change depending upon what *type* of field you have selected at the time. When multiple fields have been selected, only the settings for the first field will be display, but all selected fields will be impacted by the settings you make.

C.4.1 Adding Fields

1. Optionally select an existing field to insert beneath, then click the + under the (a) marker, or press the Return key.
2. Type in the name of the field, and click anywhere else to confirm.
3. From the **Type** field, marked (c), select what type of field this should be:

    — *Text*: a simple plain-text field, useful for storing information that doesn't often repeat, like descriptions or in the case of this example, the authors of the material we are tracking.

    — *Checkbox*: it works just like all of the other checkboxes you've ever encountered. In this case we're using one to track whether we own the book yet.

    — *List*: this functions similarly to how the Status field works: you can create a simple list of choices to select from on a per-item basis. In

this example, we're using this to track "Research Type"; whether it is a book, article, etc.

— *Date*: with this field you can type in date and time stamps, or select dates from a handy calendar.

4. Select any optional settings for the field in the area marked (d).

C.4.2 Deleting Fields

1. Select the fields you wish to delete. As normal, you can use the Shift and Cmd to select multiple items.
2. Click the – button marked under (a), or hit the Delete key.

See Also...

— Overview: how custom metadata (section 10.4.1) can benefit your work.

— Settings: adjusting the available fields and their settings, in project settings (section C.4).

— Inspector: editing custom metadata on a per-item basis with a form built into the inspector (subsection 13.5.2).

— You can use list and checkbox type fields to filter outliner & corkboard views (section 11.4).

C.4.3 Text Fields

Text fields are very simple plain text, and will work best with information that tends to vary from one item to the next. If you intend to use information that repeats frequently, lists might make for a better type of field.

The text field type (Figure C.5) has the following settings available:

Wrap text When a text field is set to wrap, it will expand the height of the field as necessary to display all of the text entered into it. In the inspector, this will push down any form fields below it, and likewise in the outliner, the height of the row will be increased to make space for the text (much like it does for the synopsis).

When unwrapped, only the first line of text will be shown, after which you will need to click into the field to view and edit the rest of it.

Use Colored Text When checked, the **Text Color** selection tool below will be enabled; click into the colour swatch to open the palette and select a colour. The text you type into this field will be coloured in both the outliner and the inspector. The colour will also be used when printing or compiling with metadata visible, though in that case it will be used to colour the field name, rather than the text itself.

C.4.4 Checkbox Fields

Checkboxes are simple on/off switches that you can add to your items. They will be useful for cases where a condition is either true or false, and can be set to either be ticked by default or not.

Tick by default Sets the default condition of the checkbox for all items which haven't at any point in the past been set one way or another. Whenever you click a checkbox, the result will be to toggle whether it is on or off, and doing so explicitly sets that condition to the item. It will no longer be impacted by default settings. Items that have never been adjusted one way or the other will dynamically change depending upon this setting, even retroactively.

C.4.5 List Fields

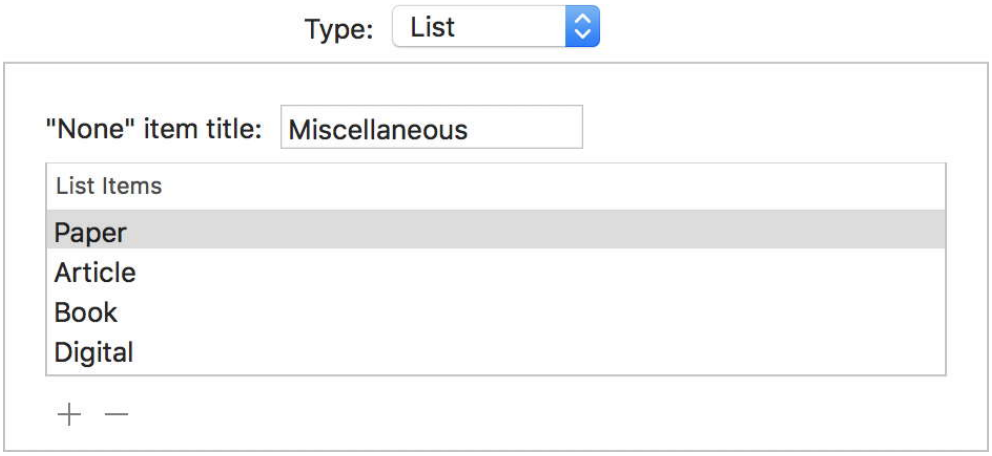


Figure C.6: Custom list fields are best when the possible values are of a limited set.

Lists are fields that will present a dropdown box when clicked upon, offering you the choice between the different set values that you will define in this pane (Figure C.6).

The "None" item title entry at the top of the list will set what will be printed if an item has no assignment made to it yet. An empty value is permitted; be aware this will cause the dropdown menu to contain an empty slot at the top of it. All items, new and existing, will use this setting by default.

— To add a new list item:

1. Optionally select the existing item you wish to insert the new entry after, and click the + button or press Return.
2. Type in the text as it will appear throughout the project.

3. Press Return to confirm.

— To remove selected items:

1. Select the items you wish to remove (you can use Cmd and Shift clicking to select multiple items).
2. Click the – to remove them from the list.

— To reorder items (you can select multiple items to drag them together) use simple drag and drop. This will impact their position in the dropdown menus.

— To rename an item, double-click the text field, type in the new text and press Return. Any items using this assigned value will print the revised text.

C.4.6 Date Fields

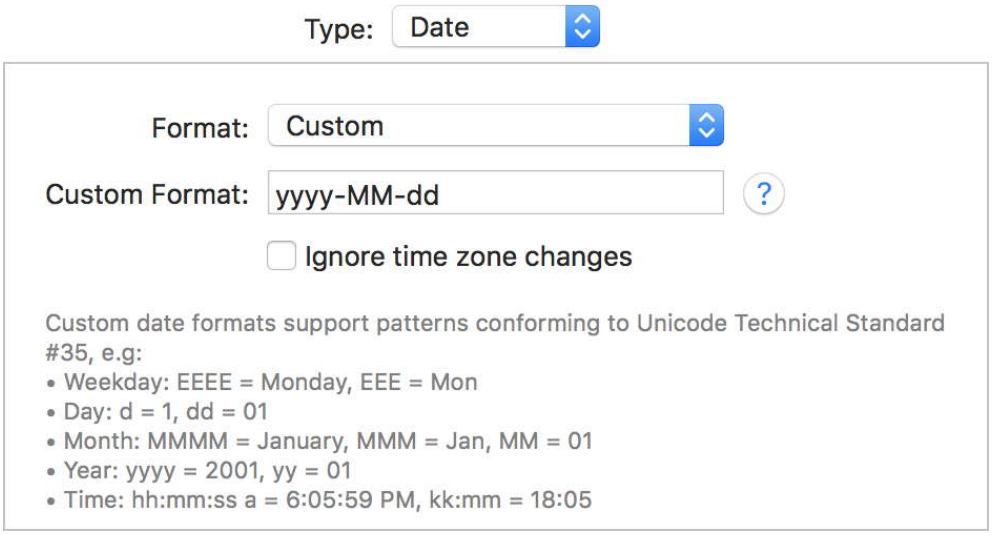


Figure C.7: Custom date fields work best when you need to record a conventional calendar date and time stamp.

For those cases where you need to record a conventional date in an item, the date field should do. If you are looking to record highly customised, historic or fictional dates, it might work best to use a regular text field instead, as this field will presume a modern time reckoning.

When using the date field in your project, you'll be able to type in dates using natural language to have Scrivener convert the text you type into a proper date and time stamp. It is not necessary to copy the format that is used to print the date when doing so, and indeed with custom formats the software will likely not recognise it unless it happens to be a fairly standard format. Refer to the Custom Metadata Pane (subsection 13.5.2) for more information on how to input dates into these fields.

Format From this dropdown you can select from your system's standard date and time formats. You can choose between having a date (in a variety of forms), a date plus a time, or just the time. Lastly, there is a "Custom" option at the bottom which opens up the following options.

Custom Format Supply a text format using the provided tokens, which will appear when selecting the "Custom" setting in the **Format** dropdown (Figure C.7). In the provided example, the format string will result in a date stamp like this: "1984–04–01", or April the 1st of 1984.

If you are looking for other tokens to use than the few basic ones we have provided in the help text, click the "?" button to the right of the **Custom format** field. This will open a web page containing the complete list of tokens that are recognised by the software. The page itself is very long and technical, the only pertinent section is "Date Format Patterns", which your browser should load to automatically.

Ignore time zone changes This option is always available, even when using a built-in format. By default the software will store the date with your current time zone embedded in the format. When you travel to another time zone, or if you live in a region where summer time is used, you'll find date stamps adjust to provide the equivalent in local time.

For cases where you want to store time abstractly, such as in a timeline of events, you might not want to have the dates and time shifting if you go on trip. Use this checkbox to have dates you enter recorded without time zone information. **This checkbox only impacts how new dates are recorded**. If you change this setting after having already entered a bunch of dates, the originals will go on acting as they have in the past.

Return to chapter ⤴

C.5 Formatting

Projects all use a central set of preferences (established in the Editing: Formatting preference tab (subsection B.3.2)) for determining what formatting (font, paragraph spacing and indenting, tab stops and so forth) will be used in new documents you create within them. For cases where you have a project that needs its own settings, Project Settings gives you a way to override these two global settings, and meanwhile provides a few additional options not available elsewhere.

Scriptwriting and Fonts

When using Scrivener to compose scripts, the font for the script will be selected for you based on the script settings, not the application settings or this panel. See Scriptwriting (chapter 19) for further details.

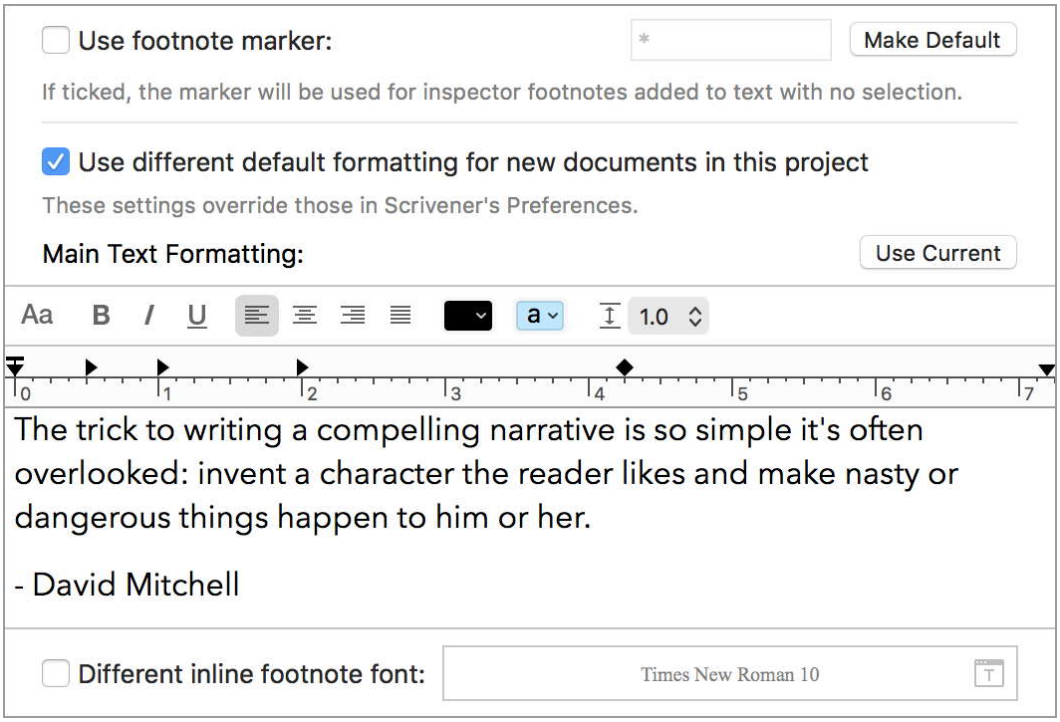


Figure C.8: The Formatting pane in Project Settings is used to apply special settings to individual projects.

Use footnote marker This alternate form of referencing linked footnotes (subsection 18.3.6) will place a custom marker after the selection or current word, rather than highlighting the phrase you wish to footnote. In documents that make use of heavy notation, this can help keep the editor clean of extraneous markings while writing. To use this:

1. Tick the **Use footnote marker** checkbox.
2. Type in the marker you would prefer to see while writing.

Existing footnotes will not be altered and will continue to function normally. However it is important to not change the marker itself once you have started using it. Doing so would cause the marker to become visible in your output.

If you prefer this behaviour for all of your projects, click the Make Default button to the right of the marker field. This will force the checkbox setting on *all* projects, even those you've already created. The marker setting will be used for all new projects going forward; existing projects will never have their marker setting disturbed.

Main Text Formatting This is where you can choose distinct formatting sections for this project, separate from those used by other projects in the application level settings. All aspects of formatting will be overridden by

the choices you make here. To enable this section, tick the **Use different default formatting for new documents in this project**.

Use the mock editor to set up your preferred styling for this project. This works just like the similar setting in the Editing: Formatting (section B.3.2) preference pane.

The Use Current button provides a handy way of importing settings from any background text selections in the active editor. So if you've already set up a document to look the way you want it to, using this button can save some time.

Different inline footnote font As a subsidiary function of the above, this setting will additionally overrides the application setting for the inline footnote font. It will work even if you are not currently using custom font for footnotes in the main application preferences, but it will respect the secondary option in that pane, **Use inline footnotes font for inspector footnote too**. When that is enabled, the choice you make here will impact both types of footnote.

This setting is primarily for your own personal preference, as the compiler can adjust the fonts of all footnotes to a uniform appearance.

Return to chapter ⤴

C.6 Auto-Complete List

Every project has its own stash of auto-completion phrases that you can make use of. By default these will never get in your way as you type (unless you are scriptwriting, where we find most scriptwriters prefer aggressive replacement suggestions), but rather will be shown after typing in a few letters and then pressing the completion shortcut, ⌥Esc.

You'll find an excerpt of some of the auto-completions that were used in the project that drafted this user manual—specifically the names of the various "Behaviors" and "Appearance" preference tabs (Figure C.9).

See Also...

— Custom Auto-completion (subsection 15.9.2): details on how to trigger completions and add new ones on the fly, while writing.

— Scriptwriting Auto-Completion (subsection 15.9.4): describes the auto-completion behaviour when making use of the scriptwriting features.

C.6.1 Auto-Completion Scope

Scriptwriting has a concept of scope, or elements that are used to format the text. We might want certain words to be completed while typing into a character element, but not while we're typing into scene description element. Furthermore,

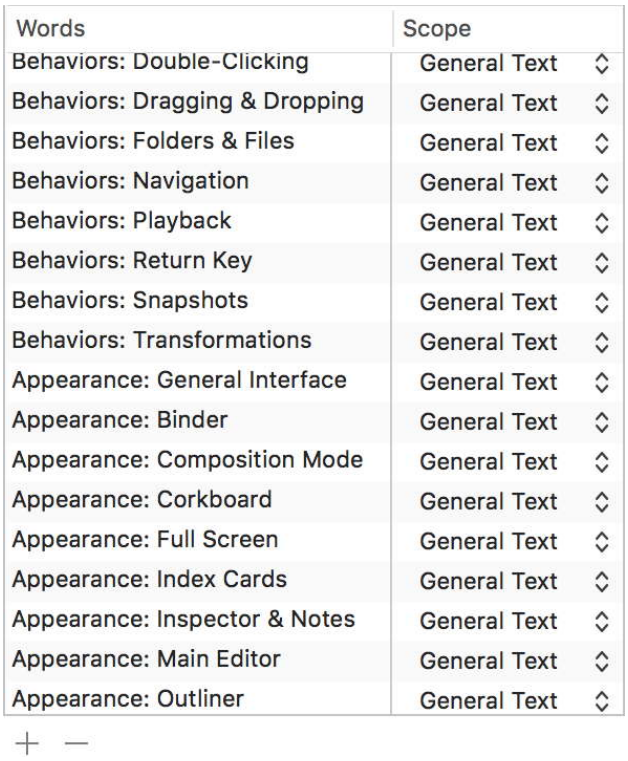


Figure C.9: A project's auto-complete list can greatly aid keying in frequently used phrases.

most of Scrivener's script settings will add such key words and phrases automatically as you compose the script.

For example, if you type in the name of a character in the "Character" element, it will automatically be added to your project's auto-complete list. These automatic additions will be scoped to element they came from. That means the completion will not be suggested when you are typing in some other element other than "Character". By and large you will not have to bother with setting anything up here. As you write, Scrivener learns about your script and becomes better at making suggestions to you depending on where you are and what you are typing.

If you want to make changes to the scope of a completion, so that it is present in different or even all element types, use the "Scope" column to change the element type (these will differ depending upon your script settings). There are two special settings available to all projects:

— *General*: The default when adding your own to the list by hand, and used by non-scripting documents as well. Those documents set to use scriptwriting will ignore entries of this scope (excepting text set to the special "General" or "General (Centered)" elements).

— *All (Text & Scripts)*: Similar to "General", only these entries will be available everywhere, even in scripting elements that maintain their own lists.

However, if a script element has been configured to not use the project auto-complete list in its script settings, then it will ignore the entry, no matter what scope settings you use.

Return to chapter ⤴

C.7 Special Folders

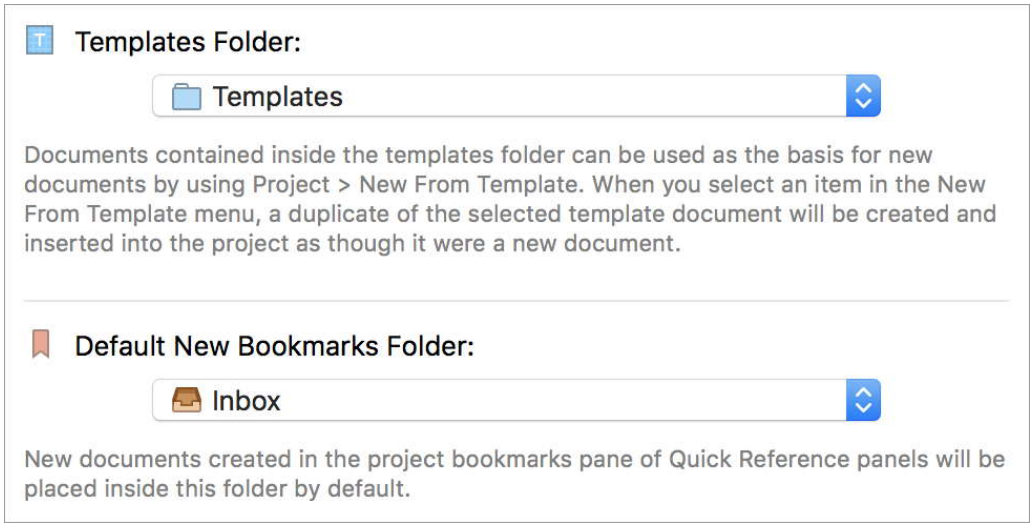


Figure C.10: Configure which folders in your project should be used for templates and new bookmarks.

Scrivener has a couple of features that make use of a designated folder in your binder:

Templates Folder The folders, files and other items you place into this folder will be provided project-wide as boilerplates for creating new files throughout the binder. A common example of this feature is the "Character Sketch" file, made available in the built-in "Novel" template. The folder and its contents are something you'll need to create in the binder itself, but here is where you will designate that folder for being used by this feature. If Scrivener finds a folder in your binder called "Templates", then it will provide it as an option at the top of the menu for convenient access. Read more about setup (section 7.5.1).

Use the "No Templates Folder" selection at the top of the menu to disable the feature.

Default New Bookmarks Folder By default, when you click the + in the footer of the Quick Reference bookmark sidebar (subsection 10.3.3), you will be asked where the new file you are creating should be placed. If you would instead prefer all new project bookmarks be created into a central folder,

you can create a location in your binder for this, and then select that folder using the drop-down.

Use the "Ask Every Time" selection at the top of the menu to disable the feature.

See Also...

— Document Templates (section 7.5): everything you need to know about how they work, how to set them up and make use of them in your projects.

— Project and Document Bookmarks (section 10.3): read more on the topic of using project bookmarks in general.

Return to chapter ⤴

C.8 Background Images

Figure C.11: Default Composition Mode settings (behind) vs settings with a backdrop image.

Background graphics should be ideally sized to be no larger than the largest screen that you intend to be using. Images must be stored in memory while the project is open, so very large files can bog down the software and reduce stability.

Composition Mode Backdrop Choose a background image for your composition environment, rather than the plain background colour. The image will be stretched to fit the screen as necessary, without distorting its shape. This feature is not compatible with Page View mode.

- **No backdrop**: will use a solid colour as specified in the Appearance: Composition Mode preference pane (subsection B.5.3).
- **Choose from disk**: click the Choose... button to select an image from your computer. This will import the image into your project settings, leaving the original free to be moved or renamed.
- **Use image from project**: a list of image resources found within the project will be listed in this dropdown.

To remove the backdrop image, select the **No backdrop** setting and click the OK button.

Freeform Corkboard Mode Background For better performance, use smaller, tiling textures rather than large images. If you intend to use a graphic that is meant to be a structure to lay out your cards upon, you should leave a small amount of padding above the functional part of the image, as the image will start a bit above the viewable area of the corkboard. The image will still tile, so you may also want to ensure there is enough padding to the bottom and right of the functional area to push the tiling effect out of view.

- **Use default background**: defers to the default setting in the Appearance: Corkboard: Colors preference pane (section B.5.4), which can either be a solid colour or a texture image.
- **Choose from disk**: click the Choose... button to select an image from your computer. This will import the image into your project settings, leaving the original free to be moved or renamed.

C.9 Backup

In this pane you can override the global application backup settings for this project. These settings will not enable automatic backups if they have been disabled globally, they will merely modify what settings already exist.

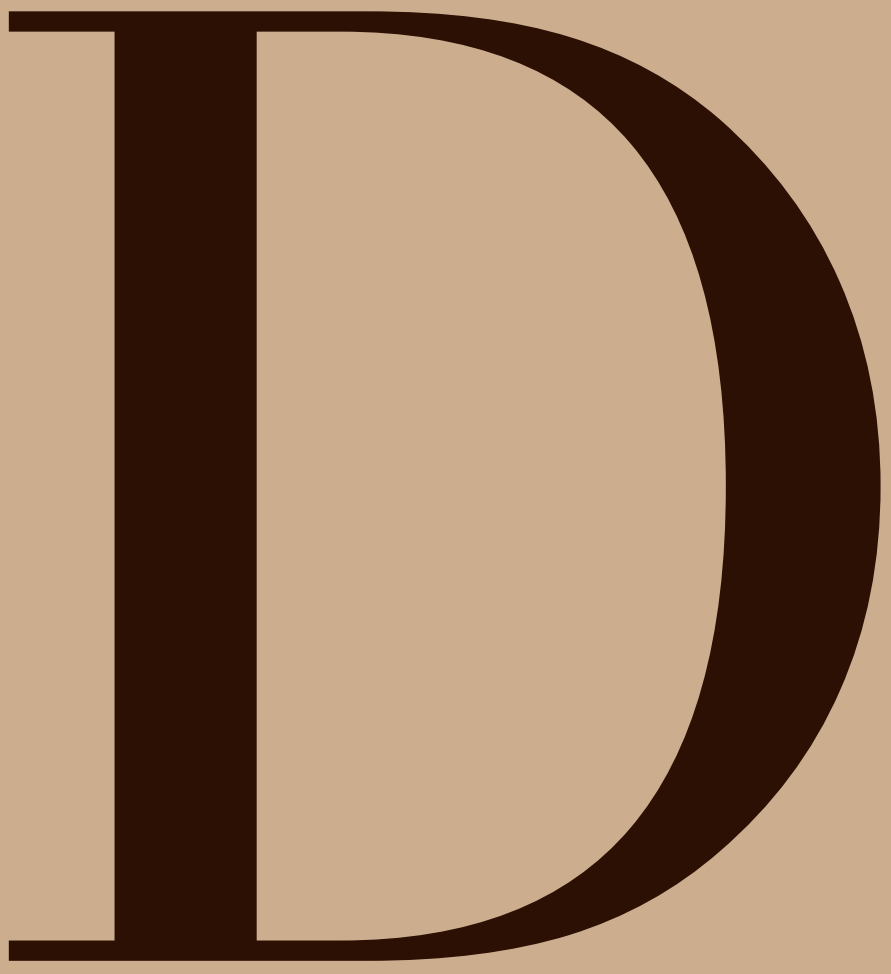
Exclude from automatic backups When enabled, this project will not be backed up automatically under any circumstances. It can still be backed up manually using the menu commands found within the File ▸ Back Up ▸ submenu; take care to do so periodically, or have another backup routine in place to keep the contents of this project protected. This option can be useful for very large projects take a long time to back up.

Use custom backup folder for this project This can be useful for cases where you want to keep the backups for this project separate from your other projects. If you're working in a secured environment, you could choose an encrypted disk as the backup location. Once enabled, click the Choose... button to select an alternate folder for backing up this project to.

The Open backup folder... button will reveal this project's backup folder in the Finder.

Click the Open Backup Preferences... button at the bottom of this pane to load the global Backup preference pane (section B.8).

Scrivener's Compile Formats



This appendix will document the built-in compile formats made available in the Scrivener installation, as well as those formats that have been supplied within a few of its project templates exclusively. We hope that by and large the use of these formats is intuitive, but if you wish to modify them or better understand how they work, this section will provide information on their use.

Given that not every format is available to every type of file you can export with the compiler, the list has been broken up by rough category of file type.

D.1 Default

Meant as a very simple format primarily used to pass through editor formatting and "glue together" the pieces in your draft folder with little alteration. Folders will still generate page breaks by default, and it comes with a few layout choices for adding additional headings as need be. This is a good starting point for your own formats, or as a basic way to export projects that do not require any special compilation options. It's also a good option if you'd rather defer formatting for work in another word processor or desktop publishing program.

D.2 Word processing and Web

These formats pertain to the print, PDF, RTF, RTFD, DOC, DOCX, ODT and HTML types.

D.2.1 Enumerated Outline

If you need a basic outline of topics (taken from the binder titles) alone in an indented list, this format is a good starting point. Unlike most formats, it will not export any text, only the titles of documents. The four section layouts that come with this format provide different spacing and numbering schemes for your outline. You could use a mix of styles for different types of documents in your project, but in most cases it would be best to assign all section types to your preferred formatting scheme.

If you want to create your own numbering schemes, you could duplicate the format and use one of the existing Section Types as a starting point. The placeholder used to generate numbering is set in the "Title Options" tab.

To adjust the amount of indent applied per level, change the **Add indent per outline level** in the Transformations format option pane.

D.2.2 Full Indented Outline

Presents an indented, easy to read outline that includes titles and synopses for all binder levels. The provided layouts provide a choice between alphanumeric numbering, hierarchical or no numbering at all. Refer to "Enumerated Outline" in the previous section for tips on adjusting the indent and numbering styles.

D.2.3 Manuscript (Courier)

Formats your book using standard Courier 12pt type and a number of common conventions such as scene separators as hash marks, double-spacing, underlined emphasis (instead of italic), page numbers, standard page headers and so forth. It has broad set of section layouts capable of handling a book with parts, chapters, scenes and titled sections. It is thus equipped to work with all of our built-in book generating project templates, fiction and non-fiction alike.

D.2.4 Manuscript (Times)

Functionally very similar to the Courier manuscript format, this uses Times New Roman, which is another commonly required typeface for submission manuscript. Scene separators remain hash marks, double-spacing, page headers and so forth. Italics will be rendered as italic text rather than underscored. It provides the same array of layouts as the courier format.

D.2.5 Modern

Providing a fresh look, using the respectable Avenir Book typeface, coupled with Helvetica Neue Bold heading and Light subheadings, this layout is designed for printing primarily. It includes a full complement of layout designs, working with all of our book generating project templates.

D.2.6 Outline Document

This layout is designed for printing full outline information (title and synopsis), but in a standard document layout rather than an indented format. It also has several layouts to choose from that only export the title, which could be useful for some projects where larger categorical groupings like parts and chapters may not have any specific synopses of note and just need to insert a sectional break. The page footer contains the title of the work, author's name, date and page number, separated from the page with a dividing line.

The included section layouts:

— Part Number and Part Title both generate a page break followed by a heading into the outline. The latter also prints the name of the document after the number.

— Chapter Number and Chapter Title insert a heading into the outline. Use the latter if "Chapter" is the wrong prefix for the type, as it will just print the title of the section prefixed with a number.

— Numbered Synopsis (Restarts) uses a numbering token that will restart whenever one of the above section types are encountered.

— Numbered Synopsis and Titled Synopsis using either a simple printed number (like “One”, “Two”…) or the section name for a heading, above the synopsis.

— Lastly, the Synopsis layout merely prints the synopsis without any numbering or heading.

D.2.7 Paperback (5.06″ x 7.81″)

A format designed to produce a typographically pleasing layout that could be taken into a word processor or desktop publishing program for self-publication with little effort. One might also try using the system PDF output to skip that phase. As with the other book generating formats, it is designed to provide a full spread of layouts to accommodate many different book styles and structural setups.

D.2.8 Proof Copy

A useful preset for internal proofing prints. It will reformat your script to double-spacing so you can easily take notes, and print a disclaimer after each section heading as well as in the header, making it easy to send out “Not for distribution” copies to your proofing team.

D.2.9 Script or Screenplay

This format is also available to the scriptwriting formats, and is documented there (subsection D.4.1).

D.2.10 Vellum Export

Available only to the DOCX file type, this Format is designed to convey your work to the Vellum book creation tool[1].

It is best used with styled text (section 15.6), which Vellum depends heavily upon to correctly format your work. For the most part, you may use Scrivener’s styles intuitively and should find they all work as expected when imported into Vellum. This compile Format will check for the stock style names used by Scrivener and convert them to style names that are recognised by Vellum. If you use your own style names, you may want to edit the format’s list, with the Styles compile format pane (section 24.5).

Additional style support is provided to (though not defined by default in stock projects):

— Epigraph

[1] https://vellum.pub/

— Dedication

— Book Tittle

— Book Subtitle

— Author

Simply create a style in your project by this name, to have the Format pick it up and handle it appropriately.

Since this format produces a document tuned to work with one specific program as a transfer medium, it is not expected for it to look "right" in other programs, or when viewed on its own.

D.3 Ebooks

Since most eBook readers will handle the majority of the text design, font choices and layout, choosing a format will be more about getting the basic building blocks together.

These formats apply to ePub 3, ePub 2, Kindle KF8 and Mobi types.

D.3.1 Ebook

A suitable, simple format for eBooks. ePub and Kindle files generally require basic and flexible designs in order to be displayed on many devices, from cellular phones to tablet computers to dedicated black & white e-ink displays. This is a good starting point for your own formats when creating eBook formats.

D.3.2 Ebook Screenplay

When publishing a screenplay intended for display on an eBook device or reader software, you'll want to use this format, which has a stylesheet designed to print the various scripting elements in a format mimicking a screenplay. Since the average reader won't have a screen large enough to truly display a standard format, some of these elements will be estimated, or designed to merely suggest what they would look like on a typical printed page.

D.3.3 Outline Document

This format is also available to the general word processing and web formats, and is documented there (subsection D.2.6).

D.4 Scriptwriting Formats

This format are available to the Final Draft and Fountain types.

D.4.1 Script or Screenplay

When used with the Final Draft and rich text formats, most of the formatting will already be done in editor, so the format itself need only concern itself with page layout and converting "smart" punctuation to "dumb". Most of your script should be assigned to the "Text Section" layout.

Fountain, as a plain-text format, does not have any text formatting built into it. Scrivener will handle the conversion to the syntax necessary to build this format.

D.5 Plain Text and Markdown-Based

These formats are displayed when using the TXT, MultiMarkdown or Pandoc based types.

D.5.1 Enumerated Outline

This format is also available to the general word processing and web formats, and is documented there (subsection D.2.1). The main difference of note is that the indenting used to indicate hierarchy will be converted to literal whitespace for plain-text. Those using a Markdown-based format should use the "Markdown Outline" format instead.

D.5.2 Markdown Outline

If you are looking to convert the outline of your project into a Markdown-style indented bullet or enumerated list, then this format will be preferable to the "Enumerated Outline" format, in that it has been designed specifically to generated valid Markdown lists. This format includes a simple LaTeX design, which will be used in conjunction with the MMD to LaTeX/PDF compile file types.

You can either assign all of your project's section types to one of the listing types, or mix and match based on type—though keep in mind that the end result must conform to what Markdown will expect: a strict use of bullets or enumeration per level.

Be aware that when using some methods of filtering or narrowing down the Draft folder, the result may not out of the box produce a Markdown list, as Scrivener may insert indent whitespacing that in effect causes the list to become a code block. This should be a simple matter to clean up in a text editor.

D.5.3 NaNoWriMo (Obfuscated)

This simple preset has only one layout that passes your text through without headings. The official NaNoWriMo word counter works best with plain-text files. The obfuscation is done via an extended list of letter-to-letter substitutions which will completely obfuscate the wording of your draft, making it nearly impossible to read, without impacting the word count. This format should not be

used for backups! It is purely for submitting your draft without risking sending your actual novel to another site.

D.5.4 Plain Text Manuscript

Designed to serve as a basic plain-text manuscript with emphasised headers, it is a good format if you want a hard long-term backup of your manuscript. Since it is formatted using plain-text conventions, it will last for many decades to come. It's also a good format for sharing text through mechanisms that better use plain-text for transmission, such as email, newsletters, readme files and bulletin boards.

D.5.5 Plain Text Screenplay

For integration with software that can import plain-text formatted screenplays[2]. This format will convert spacing and indents to literal whitespace, as well as pad the text file with enough whitespace to print 1" margins when printed using 12pt Courier. The result should be identical to a formatted screenplay, or what one would get when using a typewriter. Refer to Exporting Scripts (subsection 23.5.2) if you want to modify the format or make your own.

D.5.6 Basic MultiMarkdown

Since most Markdown-based systems will be doing the bulk of the actual formatting in post-processing, this Format primarily concerns itself with establishing the structure of your document. Through the three layouts provided, you can print sections as headings, headings with text and text alone. The heading-based layouts will automatically insert the appropriate number of hashes to match your outline hierarchy with heading structure depth. For example a level three outline item will be prefaced by three hashmarks. There is thus no need to map "parts" vs "chapters" to specific layouts, rather parts and chapters can both use the same heading-based layout and will automatically be treated as parts and chapters by the nature of their differing heading depth.

It is also set up to convert Scrivener's stock stylesheets (where applicable) to logical syntax. Block quotes will have a > prefixing each paragraph, code blocks will be tab-indented, code spans will have backticks surrounding them, and captions, when properly placed adjacent to figures or tables, will be embedded in the syntax. The stylesheet also includes CriticMarkup support, if you create and

[2] Not to be confused with Fountain, which is a newer format with a rigid syntax, this format is more along the lines of how you would type in a screenplay using a typewriter, with tabs and spaces.

use styles by the names of "Addition", "Deletion" and "Highlight". Inline annotations will also export as CriticMarkup comments.[3]

This format is also suitable for Pandoc, as it doesn't generate any syntax specific to MultiMarkdown.

D.6 MultiMarkdown LaTeX and PDF

When selecting "MultiMarkdown → LaTeX (.tex)" or "MultiMarkdown → PDF" as the compile file type, the formats listed in this section will become available. With the exception of the "Modern (Custom LaTeX)" format, the available document classes use designs created by Fletcher Penny, the developer of MultiMarkdown.

Looking for LaTeX Without the Pain?

These formats can be made use of with very minimal, or even no MultiMarkdown usage at all. With the **Convert rich text to Multimarkdown** option in the General options area of the compile overview screen (section 23.4.3), a project that has been composed quite ordinarily, using Scrivener's rich text features, could take advantage of the high quality typesetting that this workflow affords.[a]

[a] Indeed, you will find that many of Scrivener's stock project templates are already configured and optimised to be used with these formats, in a way that will work equally well whether you prefer to compose in Markdown or not.

These formats have a few common features available to them:

— In addition the basic Section Layouts the "Basic MultiMarkdown" Format uses, the LaTeX-oriented formats include the following:

  – *Text Section with Break*: using a method suitable for each format that supports this layout, a scene or section break of some sort will be inserted between binder items that have been assigned to this layout.

  – *Unnumbered Chapter*: the alternate syntax for a `\chapter*` command will be used, leaving the heading unnumbered and omitted from the table of contents (where applicable). This can be used for front & back matter, or any documents that do not fit into the main chapter flow.

  – *Unnumbered Section*: as above, only using the `\section*` command and including the text of the item that is assigned to this layout.

[3] These styles cannot be used in conjunction with full RTF to MultiMarkdown conversion.

— Stylesheet support is expanded to include additional optional features. To take advantage of these styles, you will need to create them in your project (copying the name and type of style). The appearance of the styled text is entirely up to you, as the sole function of these styles is to print raw LaTeX code:

  - *Index Key* (character style): use this style to mark text as being an index key. For example if "word" were marked with such a style, then it would be compiled as `\index{word}`.
  - *Index Term* (character style): similarly, this leaves the marked word visible in the output, for those cases where the key and the phrase are the same: `word \index{word}`.[4]
  - *Raw LaTeX* (character style): use this to mark out any ranges of text in your document that should be treated as raw LaTeX syntax, as MultiMarkdown will otherwise encode the special punctuation so that it prints as visible text in the output.
  - *Attribution* (paragraph style): with the memoir-based formats (which include all but the "Book (Tufte)" format), the `\sourceatright` command will be used on the marked text.

Import Extended LaTeX Styles from the Extras Pack

If you would like to add the extended LaTeX and MultiMarkdown styles, used by these compile formats, to your project, you will find an example project containing them in the Extras Pack (Appendix F).

Using "9-extended_latex_stylesheet.scriv" as a source project, following the instructions provided in Copying Stylesheets Between Projects (subsection 15.6.5). Further tips and instructions can be found within that project itself.

D.6.1 Article (Memoir)

Uses a layout of the expansive Memoir LaTeX document class[5] that has been tuned to mimic how the stock `article` class looks. The benefit of using this class is the wealth of customisation options available to Memoir in general.

The compile format has an inherent "Base Header Level" of 2, which means the top level heading will be considered a chapter rather than a part. Chapters will be formatted more like section headings. If you would prefer a different top level heading, adjust this setting in the Metadata compile format pane (section 24.11).

[4] This style cannot be used in conjunction with full RTF to MultiMarkdown conversion.

[5] https://ctan.org/pkg/memoir

D.6.2 Book (Memoir)

A fairly stock implementation of the Memoir book class, which is generally useful for non-fiction as well. This is a great starting point if you wish to build your own design from scratch (in fact this user manual you are reading made its humble beginnings using this boilerplate).

D.6.3 Book (Tufte)

A document class with a design inspired by the books of Edward Tufte[6]. This format supplies a few additional optional styles that you can make use of by adding them to your project:

— *Full Width Text* (paragraph style): This will typeset the marked paragraphs full-width, into the right margin area, using the fullwidth environment.

— *Margin Note* (character style): Places an unnumbered margin note alongside the paragraph the marked text is within, much like a footnote, but without the cross-reference markings.

Images will be placed in the main text block, and should be no wider than 310 points. If you wish to make use of this system's margin figures or full-width figures, you would need to create your own mechanisms for doing so.

Section breaks will use the `\newthought` command, supplied by this package. This has the effect of adding an empty line of space between sections, suppressing the indent on the first paragraph of the next section, and setting the first three words to all-capitals. If you would like to adjust how many words are capitalised, create a duplicate of this format, and edit the regular expression in its Replacements compile format pane (section 24.15). For example, to select the first four words (the number "3" is used, because the fourth word is selected afterward, without the space following that word):

```
^@@((?:\S+\s+){3}\S+)
```

It should be noted that this class cannot be typeset using the XeLaTeX engine.

D.6.4 Manuscript (Courier)

Using a simple design put together by Fletcher Penney, this will typeset your document using a traditional submission manuscript format: 12pt Courier will be used for all text, with 1" margins and double-spaced lines. There are a few peculiarities of this format to be aware of:

[6] https://ctan.org/pkg/tufte-latex

— This format does not handle parts, and so in most cases you should set the Base Header Level to "2", in your project's compile metadata settings (section 23.4.2).

— Bold text has no treatment, you should use simple emphasis alone (which will be underscored in the output).

— Chapter breaks will be untitled, and section breaks will simply insert a "#" between scenes.

— Other non-fiction oriented commands, such as footnotes, tables and so forth, should not be expected to work.

D.6.5 Modern (Custom LaTeX)

Unlike the other formats, this one has been designed by Literature & Latte, based upon its "Modern" compile format for RTF and other traditional printing methods. That aside, it is a modified version of the Memoir system in article format. All of the settings for it are stored directly within the compile format itself. You can easily modify how it works from the LaTeX Options compile format pane (section 24.12).

For Best Results, use XeLaTeX

This design makes use of system fonts, and as such it will look much nicer when typeset using the XeLaTeX engine, rather than pdflatex. Since Scrivener uses the latter for its direct-to-PDF option, it has been set up to handle non-system fonts as a fallback. It will also use a standard three-asterisk section break instead of a Unicode glyph, in this fallback mode.

What's New

In This Section...

If you are currently a Scrivener 2 user and are looking into whether you'd like to upgrade, this is the guide you want to read. We will discuss the major changes that have been made, and when necessary, cross-reference them to more detailed documentation elsewhere in this manual. Keep in mind we offer a free 30-day demo. It is safe to run this demo in parallel with Scrivener 2. You can play with the new version and learn it while continuing to work without interruption in your ongoing use of Scrivener.

Scrivener 3.0 is a significant release of the software, one that has been in the making for nearly four years. The core aspects of the software have been thought through with the intention of streamlining how we work without sacrificing flexibility—indeed we wanted to make Scrivener even more flexible while also striving to make routine tasks easier to accomplish. Above and beyond, we wanted Scrivener 3 to feel just like what you've become used to. Even though some of the changes have been sweeping (watch out for compile!), we hope you find the new landscape not only improved, but familiar as well.

If you're brand new to the software, you might find a better start with the Interactive Tutorial, found in the Help menu of the software, or skipping to the next chapter of this manual, where we will go over installation, and then start introducing the software from the ground level.

If you're looking for a particular feature by name that appears to have been moved or maybe even removed, searching for its name in this section may help you find it.

E.1 Compile Overhauled

The second version of Scrivener represented a complete overhaul of the compile interface from version one, and we are pleased to present a third iteration of the compile concept to you. Based on years of feedback with the system you've grown accustomed to, we've designed a replacement which draws heavily upon the strengths of the old system while using it as a skeleton to develop a more intuitive and easy to use "front end".

Given the scope of changes made to the compile system, it is not possible for your existing projects to be upgraded automatically to the new system. For this reason, if you have projects that depend upon complex compile settings, we encourage you to retain the backed up version that was created for you upon updating, as well as an older copy of Scrivener 2, should you need to reference them. It is possible to import your old compile settings and presets into the upgraded project (or even to Scrivener 3 in general). Refer to Importing Legacy Compile Presets (section E.2) in the following section for more information on that.

If you'd like to read a blog article on this topic, head on over to our site[1].

Front and Back Matter Locking Well, for one thing there is now a concept of "back matter", which will be welcome news to those in need of such. Furthermore we've made it so you can lock front and back matter folders to file *types*, meaning whenever you select between ePub, Kindle, PDF and so forth, specific different groups of front and back matter pages can be automatically applied to your compile settings. Read more about locking settings to file type, in Front & Back Matter (section 23.4.1).

Preserve Formatting in Markdown Compile In the past, the Format ▸ Preserve Formatting feature would have been used to generate code spans and code blocks with Markdown-based output. This capability has been removed, as it is now served by styles. Use the "Code Block" and "Code Span" styles provided in the stock set, or if you create your own, set them up in your compile settings using the MultiMarkdown and Pandoc Options (section 24.14) compile format pane.

The **Treat "Preserve Formatting" as raw markup** option, used in conjunction with the **Convert rich text to MultiMarkdown** setting, both in the compile options for Markdown-based formats (section 23.4.3), will cause marked text to pass through the compiler untouched. Ordinarily the conversion option will protect punctuation marks that are used by Markdown.

[1] https://www.literatureandlatte.com/blog/scrivener-3-redesigning-compile

This option will mainly be of interest to those that use the iOS version of Scrivener, with its equivalent capability. If you are only using a Mac or PC to compile, you might want to use a dedicated style instead (with the **Treat as raw markup** option enabled for it in the Styles compile format pane).

Print with Footnotes In the past, if you wanted to print or generate a PDF out of Scrivener with proper end of page footnotes, you either had to compile to another program that could do this form of typesetting, or make use of the "Proofing" option in compile settings. Scrivener how typesets its own footnotes for Print and PDF compiling. If that's all you were needing a word processor for, that's hopefully one less tool you need to get work done!

Separators by Type In the past, inserting separators into your document was either a matter of checking off boxes in the inspector or trying to shoehorn your outline into a layout that worked with the Separator's compile option pane. For example, if you wanted an empty line between scenes in a novel you could have Scrivener insert empty lines between text chunks—but then you could only ever cut to a new chunk of text for the purposes of creating a formal scene.

In Scrivener 3 you can now not only assign separators to broad types like before (folders or files), but to specific *types* of items. Say you have folders in your project set up to be chapter breaks—it is then useful to say that all chapter headings ought to have a page break before them. Instead of saying "folders have breaks", we are now saying only *this* kind of folder has a break—and if we use files for chapters instead, we can easily apply that same behaviour to the file without having to go in and change settings. Since a chunk of text can now formally be a "scene" that means other chunks of text can be "not scenes" and thus not have any separation between them, allowing you to outline far deeper than your readers will ever be aware of.

Markdown to RTF Conversion In the past, those using basic Markdown for italics and bold in their text could make use of an option in the Transformations compile option pane to convert these to rich text formatting, when using any of Scrivener's native export formats. This option has been replaced by the **Convert MultiMarkdown to rich text in notes and text** compile option, in the compile overview screen under General Options (subsection 23.4.3).

Of note, it now treats text and notes as properly formed Markdown, rather than simply extracting asterisk marked phrases from an otherwise rich text based document. If your project is not properly formed, use of this checkbox may produce undesirable results—such as paragraphs all being glued together, since Markdown require a clear line of space between each paragraph.

E.2 Importing Legacy Compile Presets

We recognise that you may have invested a lot of time in your compile settings. Should you choose to upgrade a project that is a work in progress, you will find its compile settings will be reset to default. If you would like to carry over the original settings into a new format, the following procedure should help you get most of your settings converted to the new system.

Instructions for importing your settings into a new version 3 format have been provided in Importing Legacy Presets (subsection 23.2.8). If you would like a more comprehensive guide, we have prepared a tutorial project[2] that goes over many details that may be of interest to you in transitioning to the new system.

E.3 Section Types, Page Breaks and As-Is

Any talk of the changes made to compile would not be complete without mentioning section types, a new way of assigning meaning to the folders and text items in your binder (mainly of relevance to the Draft), such as "chapter", "subsection" or even specific types such as "table" or "figure". Without going into detail, you can now categorise sections in your binder using natural terms, and in turn you can set the compiler to print these types of things in a logical fashion, rather than formatting by levels and icon types.

All of this has made the concept of using individual checkboxes for page breaks and such a bit obsolete. If a thing by its nature involves a page break, such as the dedication page in the front matter of the book, then we don't need to use a separate checkbox to denote that, if we already know that dedication page is "front matter" material.

— **Page Break Before**: page breaks are now a function of how a type is formatted by the compiler. By and large you should find the default settings and templates provide a useful set of *types* to choose from, with useful settings applied to them. To use the above example, if in the past you've used this checkbox for front matter sections, you'll find most of the compile formats contain a layout specifically designed for front matter which includes a page break.

  For those upgrading their projects from 2.x, be aware that items marked with "Page Break Before" will be assigned to a special "Section Start" type, created for upgraded projects alone. This should be considered a transitional type, but it will allow your project to continue functioning roughly as it was before.

— **Compile As-Is**: this checkbox is as well no longer in the inspector or specifically assigned to items. The choice to print a section type "as-is" is something you now make when setting up your project's compile settings. You

[2] https://www.literatureandlatte.com/scrivener-3-update-guide

might for example want to create types for Tables and Equations, both of which are mapped to compile as-is in the end, but remain distinct for their purpose in the project itself.

Upgraded projects will be scanned for the use of "as-is", and a special "N/A" section type will be created for projects that made use of the checkbox, with all items that used it assigned to that type and automatically mapped to the special "As-Is" section format in the compiler. Thus they should continue working as they did before.

E.4 Stylesheets

Ye olde formatting presets have become proper stylesheets. Head on over to Styles and Stylesheets (section 15.6) to see what's new. You'll find much of the toolset has a similar usage from what you are used in prior versions of Scrivener. The tool drop-down is in the same location on the Formatting bar—in iOS the change is seamless—and for the most part you can use them in day to day writing similarly to how you used presets.

It would be a good idea to familiarise yourself with how they work in the compile pane, as that can have an impact on what you choose to style and what you choose to leave alone as standard text.

Styles are, as formatting presets were, strictly optional! If you never had much use for the feature in the past then don't feel you have to learn anything about styles going forward.

E.4.1 Converting Formatting Presets to Styles

Do you have a bunch of formatting presets you'd like to convert up to a the new format? We won't do that for you because styles are now project-specific, but for the same reason it is easy to create a style from some already formatted text, it is easy to create styles off of text that has had presets used upon it.

You could do this in a new "Blank" project and then save that as a project template, using File ▸ Save As Template..., and now you have a starter project with your preferred styles built-in (subsection 5.4.3). But you no doubt have a number of already existing projects you'd like to update with these styles. Using the blank project as a target, import styles with the Format ▸ Styles ▸ Import Styles... menu command.

So long as your presets were consistently applied, you may be able to select bulk chunks of text at once and apply the new styles to them, getting your project up to date in minutes.

E.5 Project Bookmarks

Project notes are dead, long live project notes! That's right, the feature formerly known as Project Notes, a core part of Scrivener since its early beta days, is no

more. The good news is that if you've benefitted from using them in the past, then you should find it possible to continue working in a familiar fashion going forward, and in fact you may find yourself using the new features in ways you hadn't before.

When we took a look at this feature, and what we could do to improve the Project Notes window, it became clear that the whole idea had outgrown itself a little. Project Notes originally started as an alternate panel in the inspector, "beneath" the Document Notes if you will, and that was it. It was a solitary simple scratch pad meant to be used simply. Scrivener 2 added a separate window you could open and use like a notepad, and the ability to keep many different topical global notes was introduced. This was all neat, but they were still stuck inside that little corner of the project, unable to integrate with the rest for not being a part of the binder themselves.

In looking at the problem, we realised there were a number of features in Scrivener that were all kind of offering the same thing, but in different and disconnected ways. The result of their combination is something we can concisely refer to as a new feature, Project Bookmarks.:

1. Project References in the inspector sidebar.
2. Favorites made it possible to place items in choice positions within any menu that let you pick a binder item from it (such as the "Go To" menu).
3. And of course, Project Notes themselves.

The combination of these features can be thought of very simply: if you want to make a particular document globally accessible, either as a reference or for the purpose of taking down notes, then bookmark the item.

Reference Links are No Longer Named

In the past you could provide a description to a reference. This field has been left out of the new design, with the only text in the bookmark list being the name of the item being referred to. Thus any descriptions you used in the past will be discarded when upgrading the project, and you should be aware that changing the name of a bookmark will directly rename the item in the binder, not how you refer to it in the list.

A side-effect of this merger meant that what we referred to as document references in older versions of Scrivener now have their own dedicated pane and previewer panel below the bookmark list. Now all of those cross-references between items in the binder are instantly accessible without even going anywhere in the main editors. You can of course still do that too—references are, above exception aside, just as powerful as they always have been.

— Read more about the new features in Project and Document Bookmarks (section 10.3).

— If you'd like to read a little into in the background of this design change, we've posted an article to our blog on the topic[3].

E.6 Writing History

Your daily progress, whether it be in going from zero to a goal, or from a bloated work that badly needs trimming down to size, can be tracked using a new tool: Writing History (subsection 20.1.4). Accessed via the Project ▸ Writing History... menu command, it not only does some simple summation and calculation for you, but allows for easy export to csv format, where the raw data can be taken into a spreadsheet.

E.7 Custom Metadata Overhauled

We've added three new types of custom meta-data that you can use to enhance your record-keeping. If you've been using older versions of Scrivener and would like to make use of the new field types with existing data, this section should help you do so.

Scrivener's new date field takes natural language dates and converts them into a proper date and timestamp for you. However, it won't do this to existing text that you've stored in a text type field as there are too many different ways of inputing dates, making such an automated procedure risky. Making this conversion by hand will be slower, but you'll be able to ensure the conversions are accurate as you do them:

1. Create a new date type column alongside your original date column (subsection C.4.6).
2. Set up an outliner view so that both the original hand-typed date column and the new date column are side-by-side.
3. Using Tab to alternate between fields, copy and paste the date from the text field into the date column. So long as you used a fairly standard or easily parsed date format in the past, you should find our date detection code picks it up and converts it to a proper date in the new field.

For checkboxes, all you really need is a text field with "1", "yes" or "true" for checked and "0", "no" or "false" for unchecked. When changing the text column to a checkbox type column (subsection C.4.4), you should find they convert cleanly. Empty cells will use the default you established for the checkbox column.

[3] https://www.literatureandlatte.com/blog/project-notes-are-dead-long-live-project-notes

E.8 Snapshot Manager

If you're anything like me, you take a lot of snapshots, and furthermore you ask the software to take a lot of snapshots for you as well. This is great for keeping a track record of your progress safe as it develops over time—but it can mean a lot of clutter builds up, especially in those projects that end up being used for years at a time. Eventually even the biggest data hoarders might come to admit that a bad revision of something that was written five years ago maybe isn't worth hanging on to.

The Documents ▸ Snapshots ▸ Snapshot Manager tool has been added to help keep the clutter at bay, and to find those old odd snippets you lost to edits years ago. You can not only specifically search for text through all snapshots, you can search by date as well, and once you have a search result you can prune the results from the project with one easy click. For example, search for `<2y` to find all snapshots created over two years ago and delete them all.

Read more about it in The Snapshots Manager (subsection 15.8.5).

E.9 Enhanced Outlining

The outliner has always been a place where you can pack a lot of information into the view, and thus an integral component in getting that sense of overview that Scrivener can provide, down to the word counts of individual chunks of text or keyword listings.

We wanted to broaden what the outliner can be used for however, as many people also prefer this sort of device as a creative tool. To that end we've added a few aesthetic options that can be used to present the outline as a more content-focussed tool, and as well some settings to have it emulate the iOS style listings, made popular on macOS by Mail and a few other programs.

— Centring Outliner Content (subsection 8.3.6): much like fixed-width text editing (which is also now a default, rather than text that stretches across the entire screen), you can trim down your outliner to a basic text column and keep it balanced in the middle of the view, rather than pinned to the left. If you're spending a lot of time developing the outline itself, working in titles and synopses, you might want to take a look at this simple but comfortable adjustment.

— Using a Fixed Row Height (subsection 8.3.5): the outliner has always expanded the height of each row to match the amount of content you put into it. This content-biased approach means you never miss a word, but it can sometimes be more efficient if every row takes up the same amount of height. Scrolling becomes more predictable and static, and content is easier to parcel out with the eye. It's a common look on iOS, and has made its way into some desktop software as well. If you want to give it a quick look, check out the new built-in "Three Pane (Outline)" layout, in the Win-

dow ▸ Layouts ▸ submenu (or from the new View button on the far left of your application toolbar).

— A bounty of new options have been added for adjusting the look of the outliner. Take a look in the Appearance: Outliner preference pane (subsection B.5.9).

E.10 The Devil in the Details

We've gone over most of the major differences between versions 2 and 3, what remains are a few of the smaller changes that are worth making note of. It would not be possible to list every single change in the software, so this list will concentrate on those changes that could impact how you've used version 2 in the past.

Subdocument counts in binder If you used the older feature that prints the subdocument counts as a numeral to the right of containers in the binder, you will be interested to know where that feature has been moved to. It is now a project-specific setting, toggled with the View/Outline/Show Subdocument Counts in Binder menu command.

Migrating Your Settings By default, Scrivener 3 will start you with a clean slate for settings, giving you the default out of the box experience, even if you've used Scrivener in the past. If you would prefer to migrate your settings to the new format, then use the Manage... button in the lower left corner of Scrivener 2.x's preference window to "Save Preferences to File...". Now switch over to Scrivener 3 and use its own Manage... button to "Load Preferences from File...".

Not all preferences will be identical between versions, and Scrivener 3 will have many more preferences than 2, but those that do match between programs will be set, giving you a head start in getting the new version set up the way you like.

Dragging Links into Documents One of the simplest methods for creating a link to a document from the text you are currently working on is to drag and drop the document you wish to link to into the text editor. In Scrivener 2, you were required to hold down the Option key while doing so. In Scrivener 3 you can just drag and drop the document straight in (section 10.1). Meanwhile Option-dragging now pastes the *content* of the dragged document into the editor where you drop.

Dragging Text Out of Documents While we're on the subject of dragging, you will want to know that dragging selected text out of an editor and into another context (such as the binder or another editor) will now *move* the text, just as it is the default behaviour to move text that has been dragged within an editor. If you would prefer Scrivener worked as it used to and copied

text dragged out of the editor, disable the **Delete text dragged to other areas** option, in the Behaviors: Dragging & Dropping preference pane (subsection B.4.4).

Editor Header Bar Icon Menu Since the very beginning, Scrivener has always presented a few useful commands pertaining to the document you are working on, in the header bar icon beside the title of the document you are editing. All of that is still there in Scrivener 3, and more besides, but you no longer have to click on the icon. Instead you can now right-click *anywhere* in the header bar (other than the history buttons).

As for the icon itself, that is now something you can drag, just like dragging the document itself from the binder; this form of dragging is just as powerful as that. You can move an item to a new location, assign it to a collection, create a hyperlink to it from another text, drag it to a second header bar to load the document twice, and so on.

Excluding a Project from Backups If you have updated a project that was excluded from automatic backups, it will go on as excluded. However you should know that the option to toggle this setting has been moved from the File menu into Project Settings, under the Backup tab. While there, you may also note it is now possible to select a custom backup folder on a per project basis.

Changing Composition Backdrop If you've enjoyed the ability to set a photograph or texture to the background of your full screen writing environment, you'll want to know that option (along with many other project specific options by the way) has been moved to the Project/Project Settings... panel, under "Background Images" in the sidebar. Everything else about the feature should be familiar.

Project Properties are Compile Settings Instead of setting up the title of your work, author's name and so forth in a project setting panel, this is now done directly in the main compile interface, under the "metadata" dog tag icon on the right hand side. For projects that have been upgraded from the 2.x format, you should find the information from Project Properties migrated over to the new location for you.

Modified Default Revision Colours If you have projects which are currently using revision markings, you should be aware that the default revision marking colours have been modernised, meaning the new version of Scrivener will not "see" your old markings, for purposes of features that search by or strip out markings. If you require consistency with existing projects, you should visit the Editing: Revisions preference pane (subsection B.3.3) and adjust the five levels to their previous levels.

You will find two macOS colour file sets in the Extras Pack (Appendix F). Place these files into your ~/Library/Colors folder, and now they will be available from the third tab of the standard macOS colour picker tool.

Modified Default Highlight Colours The set of default highlight colours that Scrivener makes available in the Format ▸ Highlight ▸ submenu have been modernised. If you have upgraded a project that made use of highlights in the previous version, you will find that colours do not match the old highlights. This will mainly be of impact if you use the Edit ▸ Find ▸ Find by Formatting... tool to locate highlights of a specific colour. If you want to "upgrade" your existing highlights, you could use that same tool to search for the old colours, and simply set the highlight to the new colour (from that point on you could very easily alternate between ⇧⌥⌘G to find the next old highlight and ⇧⌘H to apply the new colour.

This will also be of concern if you have used the Scrivener colour palette to rename the highlight colours in the past, as documented in Naming Text Highlights (subsection 18.5.1). The easiest way to address this will be to remove the ~/Library/Colors/Scrivener.clr file and relaunch Scrivener so that it creates a fresh one with the new colours.

"Go To" Menu in Header Icon Menu When viewing a Scrivenings session in the past, the "Go To" menu, found when clicking upon the header bar icon menu (which incidentally is now a right-click), you would be given a simplified table of contents for the current session. This was nice, as you could easily jump around from one section to another within the text you were currently working on—but it could also get in the way if you're used to the feature letting you go anywhere at all in the binder.

We've added a new feature which is more accessible, and hopefully more easily to find, to replace this functionality. When in a Scrivenings session, you will note a new button on the right hand side of the header bar (Figure E.1). Click this to view a concise table of contents for the current Scrivenings session. You can click on entries to flip between them, and elsewhere once you're done. You can also use the arrow keys on your keyboard to move between sections.

Figure E.1: The new "Navigate to Section" button, first in this series of buttons found on the right side of the header bar.

The Text Editor Uses Fixed-Width by Default In the past, Scrivener's default text editing mode displayed text the full width of the editor, no matter how wide it was. This meant that if the editor grew very wide, text lines

could become unreadably long (and in fact fixed-width editing was set as a default for full screen mode to combat the worst cases of this). Scrivener's default way of presenting text is now to a fixed width column, one that will respect the relative number of characters that can be seen on a single line, no matter the zoom level. If you do not like this new way of working as a default, you can switch it off in the Appearance: Main Editor preference pane, under **Use fixed width editor**. You can also adjust the width of the text column in this panel, and whether or not it should be left-aligned or centred within the editor.

Custom Categories for Project Templates You can now organise your project templates into custom categories in the new project template chooser. Refer to Custom Categories (section 5.4.3) for more information.

Locked Editors Divert Navigation In the past, when you locked an editor you could freely work in the sidebar without content loading in that editor or the other split. Now, if the interface is split the other editor will receive the navigation request instead. If you would prefer the older behaviour, change the **When focused editor is locked in place** setting in the Behaviors: Navigation preference pane to "Binder selection does nothing".

Text Bookmark Feature Removed The Text Bookmark feature, which allowed for one to simple markings into the text and then access those markings from a menu in the header bar, has been removed. The Comments & Footnotes inspector tab is more than sufficient for placing markings in the text, and clicking on these notes in the inspector scrolls the editor to that point in the text.

If you used this feature and have the special inline annotation markings it used still in your text, you will either need to strip them out of the content area if you otherwise compile with inline annotations in the output, as they will no longer be recognised as anything special—or convert them to comments. The Edit ▸ Transformations ▸ Convert Inline Annotations to Inspector Comments menu command can greatly ease this process, as it can operate on bulk sections of the draft at once.

Footnote Numbering Off By Default In the past, when you compiled your document, all inspector footnotes would be numbered in the order they appeared in the compiled output, and this number would be printed in the upper left-hand corner of each footnote in the inspector. This still happens in the background in Scrivener 3, but the display of the numbers themselves has been disabled as a default. You will need to manually turn it on with the View ▸ Text Editing ▸ Show Compiled Footnote Numbers in Inspector menu toggle.

Sync Folders Will Need to be Updated If you've been using the File ▸ Sync ▸ with External Folder... feature in prior versions of Scrivener, you should be aware

that necessary internal changes have made it so that an existing sync folder will need to be rebuilt from scratch with the new version. The best result will be to sync the project one last time to this folder so that it is fully up to date, before upgrading, and then delete the folder and create a new one after you've upgraded the project in the new version of Scrivener.

External Document Links Will Need Updating The way in which documents are referred to internally has fundamentally changed in Scrivener 3, and as a result, links pointing to items within projects with the External Links (subsection 10.1.6) feature will need to be updated by hand. We apologise for the inconvenience, but given how these links are located outside of the original project that has been updated, there is no way for Scrivener to know where the original item went.

Typewriter Scrolling Now Weaker It is typical to boast of things growing stronger, but in this case we'd like to mention a feature getting a little weaker. Typewriter scrolling had one flaw with the idea: it wasn't terribly good for editing. If you were scrolling and clicking into places to fix bits of text, the behaviour to snap the typing line into the middle of the screen wasn't worth leaving on, leaving one to awkwardly toggle the option on and off.

So in Scrivener 3 the feature now stops forcing you to one line when you do just that. If you scroll away from the point where you have been writing to fix a typo, the editor will start using *that line* as the scroll point, instead. If you prefer to resume writing in the middle of the screen, hit the ⌘J shortcut (or use the Edit ▸ Find ▸ Jump to Selection menu command).

Don't like it? No problem, you can get the old way back by disabling the **Typewriter scrolling always jumps to scroll line** setting, in the Editing: Options preference pane.

Project Settings Unified In the past, many of a project's settings were scattered throughout different dialogues and menu commands. While most of the view preferences that pertain to a project are still located in the standard View menu, the rest have all been gathered into a single dialogue, accessed through Project ▸ Project Settings... (Appendix C). Excluding those settings previously found in the Project ▸ Meta-Data Settings... panel[4], the following features have been folded into this pane:

— Setting the Document Templates folder, previously set with the Project ▸ Set Selection as Templates Folder menu command.

— Custom project formatting overrides, found in the old Project ▸ Text Preferences... pane.

[4] And it is worth noting that a project's global metadata, such as the author and title of the work, has been moved into the File ▸ Compile... pane.

— The auto-completion list, previously found in the Project ▸ Auto-Complete List… menu command.

— Composition mode's backdrop, which was previously set with the View ▸ Composition Backtop ▸ submenu.

— Disabling automated project backups, previously toggled with File ▸ Back Up ▸ Exclude From Automatic Backups.

Project and Text Statistics The "Project Statistics" and "Text Statistics" panes have been combined into a single feature, Project ▸ Statistics… (⇧⌥⌘S), with two tabs. The latter now has every feature the former once had exclusively, and vice versa. To get statistics on a single text document, view it in the editor alone and then use the "Selected Documents" tab. You can click on statistics in the footer bar of the editor and get much of the same information provided here.

Compiled draft statistics may alter

While we're on the topic, it's worth mentioning that you might see a difference in your overall compiled draft count. This is most likely going to be a side-effect of your compile settings having been reset to default when the project was upgraded. If you were using options in version 2 that added a significant amount of text (like for example all of your document notes) then upon checking your word count in version 3 that number will have dropped.

Return to chapter ⤴

E.11 Menu Reorganisation

The application menus, from File to Window, have been completely rethought. We've also introduced two new major menus, "Navigate" and "Insert", the former focussing on project window navigation of all sorts, and the latter now collects all of the various types of things that can be inserted into text, from images to the current date and time. If you make heavy use of the menus in Scrivener, you may more than a few things have been moved around, but hopefully you find the changes to be logical and easy to adjust to. As before, the Menus & Keyboard Shortcuts (Appendix A) appendix provides a complete reference on every menu command.

E.12 Updates to Version 3

We are constantly looking to improve Scrivener. The remainder of this section will document significant changes since its initial 3.0 release in the Autumn of

2017. For a complete list of all modifications and bug fixes, refer to the release notes page[5] on our website.

Version 3.0.3

MultiMarkdown 6 Support

Scrivener now fully supports MultiMarkdown version 6, and integrates version 6.2.3 in the application. The most noticeable difference will be the removal of the MMD to RTF compile format, which is no longer supported, and the addition of a native OpenOffice ODT format. No longer will you need to compile to Flat XML and use LibreOffice to make the final conversion.

The other notable difference between MMD 5 and 6 is how the metadata is handled for LaTeX documents. If you have been using your own custom metadata in your compile Formats, you will likely need to update the metadata keys you use to work with MMD 6. Instead of using a general purpose "LaTeX Input" metadata key to include boilerplate files, there are now dedicated "LaTeX Leader" and "LaTeX Begin" keys. The former is used to establish the initial preamble, the latter to provide any further preamble after the insertion of document-specific metadata keys such as the title and author. Scrivener will of course make use of the new system with all of its built-in LaTeX document class selections.

Lastly we have added support for the Tufte (Book) document class for MMD to LaTeX and PDF compile formats.

Default Paragraph Style Removed

The option to automatically apply a paragraph style to text as you write has been removed from the Project Settings: Formatting pane. Scrivener is designed to work around the concept of not styling body text, and rather leaving it flexible to be transformed by the compiler when exporting. Refer to Think Different (subsection 15.6.1) for further rationale on this decision.

LaTeX Compile Support Improved

For the MultiMarkdown to LaTeX compile file type, we have added five new compile formats, to make switching between common document classes simpler. With one click you can go from formatting your work in a screen-friendly "Modern" look, to a formal article design. Further documentation on the formats is provided in MultiMarkdown LaTeX and PDF (section D.6).

For those looking to print a simple outline of topics, the "Markdown Outline" compile format now has a better default look when used in conjunction with MMD to TeX or PDF.

[5] https://www.literatureandlatte.com/scrivener/release-notes?os=macOS

In addition to this, those looking to use Scrivener as a pure LaTeX editing platform (with no Markdown-based conversion) now have a dedicated project template they can start from, in the Non-Fiction category: "General Non-Fiction (LaTeX)". This project template is designed to be used for the composition of LaTeX directly, but as well it features partial support for generating LaTeX syntax from Scrivener's built-in formatting features.

Bulk Keyword Management Improved

In the past, managing keyword assignments at a large scale was limited to bulk assignment, by dragging keywords from the Project Keywords pane (section 10.4.2) onto selected items in the background window. It is now possible to right-click (or use the ✿ button) on selected keywords to assign *or remove* keywords from the selected documents in the background.

ePub for... Kindle?

While we continue to recommend the use of the native Kindle Mobi and KF8 formats for publication to Amazon services, in some cases you may find third-party publishing agencies will only accept ePub files. We have added a new option to the general options tab within the compile overview screen (section 23.4.3) to "optimize for Kindle conversion". This will include a few tweaks to the underlying HTML, as well as navigation hints that will make the ePub file convert more cleanly to Amazon's formats.

Create New Compile File Types with Post-Processing

⟨Direct-sale only⟩ Scrivener can now tap into the command-line capabilities of your system through the use of scripts embedded directly into the compile Format, making it possible to package automated workflows to wider audience who need not know how to set up scripting themselves. In addition, post-processing has been added to the Plain Text compile file type, making it possible to create wholly new file types from scratch and processing them into end formats.

Refer to the Processing compile format pane (section 24.22) for further instructions, and the Plain Text's Markup compile format pane (section 24.10) for tips on creating your own custom file types.

Vellum Export Support

For those that make use of Vellum[6] for the final design and production of their books, we have created a compile Format for the DOCX file type. The format has been tuned specifically to produce optimum results when imported into Vellum. More details can be found in the appendix listing of Scrivener's compile formats (subsection D.2.10).

[6] https://vellum.pub/

Better Multivolume Support for Front/Back Matter

For those who write more than one volume into a single project, the front/back matter feature has been modified to allow for automatic selection of preliminary and ending material, depending on which volume you currently have selected for compile. This can be combined with the already existing feature to automatically switch between front/back matter sets based on the *type* of file being compiled. Refer to Linking Front/Back Matter to Compile Groups (section 23.4.1) for tips on setting up your projects to work this way.

Version 3.0.2

Inserting Media Time Stamps

It is now possible to insert the current time stamp into the text editor, while running audio/visual media in the other split. The menu command (Insert ▸ Media Time Stamp) does not by default have a keyboard shortcut, but one can be added if you anticipate making heavy use of this feature (section A.1).

A new script format (Format ▸ Scriptwriting ▸ Transcript) designed especially for the transcription of audio and video material has been added to the software. It has been set up so that a media time stamp will be inserted automatically when you press the Tab key on a new line, while media is playing.

Reloading Linked Images on the Fly

Previously, if you made use of Linked Images (subsection 15.7.4) in the main editor, if the image was modified on the disk or in the binder, then you would have to reload the project in order to see changes made to the file. In addition, changes made to the size of the image would not be reflected in the editor, which could cause squishing or squashing. You can now right-click on a linked image in the editor and select "Reload from Original Image" to reload the cached thumbnail from the disk, and reset the size of the image.

Browse the Web from Within Scrivener

It is now possible to, in a limited fashion, browse the Web from within Scrivener, when starting from a page that is being viewed in the main editors or the Bookmarks preview tab. In the past, all links would load in your main browser. To enable this capability, set the **Allow limited navigation in web pages** option, in the Behaviors: Navigation preference pane (subsection B.4.6).

Linked inline images can be updated from disk

For those working with inline images in the text editor that have been linked, either to the binder or the file system (subsection 15.7.4), if one updated the image data externally they would have to reload the project to have the corresponding graphic updated in the editor. Additionally, images that had their size changed

on the disk would *not* update the editor's concept of its size, causing them to appear squashed on reload.

Both of these problems can now be solved on the fly by updating the image from the disk (section 15.7.4).

Version 3.0.1

Imported Markdown documents can be converted to rich text

When importing Markdown files with the File ▸ Import ▸ Import and Split... menu command, a new option has been provided to convert Markdown syntax into rich text, removing all Markdown syntax in the process. Internally, Scrivener will convert the text to HTML and then import the HTML to RTF format. For simple documents this should suffice, but for more complex texts, using native Markdown conversion engines (such as those provided by MultiMarkdown and Pandoc) will obviously produce the best results, at the expense of taking an extra step.

Appearance themes now easily accessible

Scrivener has supported appearance themes, the ability to change the colours, fonts and other appearance settings throughout the software, for many years. It is now much easier to switch between these themes from the main Scrivener ▸ Themes ▸ menu. Read more about creating themes in Preference Presets and Themes (subsection B.1.1).

Image size can be requested in percentages for e-books

When using the image link placeholder syntax (subsection 15.7.5), it is now possible to request the size of the image in percentage points, as relative to the size of the screen it will be displayed upon, in addition to any point size measurement used for print-based output. Specifying image size in such terms as "100%", to fill the width of the display, is often more useful than trying to find precisely the right size to display it at, when the shape and scale of the display the e-book is being read upon can so dramatically change from one reader to the next.

Extras Pack

F

A number of examples throughout this manual have been provided for you to work with in a hands-on fashion. There is a downloadable "extras pack" from our website, that you can obtain using the following instructions:

1. Visit the User Guides download section on our web page[1].
2. Beneath "Scrivener" on the left, select the "Extras Pack" option from the "Select Format" dropdown menu.
3. Click the red download button to the left.
4. If your browser has not already done so, extract the folder from the downloaded .zip file and place it somewhere convenient.

The files are all numbered so you can easily find them in the folder.

[1] https://www.literatureandlatte.com/learn-and-support/user-guides

Credits & Acknowledgements

G

Concept, Interface, Design and Development
Keith Blount

Additional Design
Ioa Petra'ka

Documentation
Keith Blount
Ioa Petra'ka

PDF Design
Ioa Petra'ka

Toolbar, Binder and Template Icons
Janik Baumgartner

Application Icon
Janik Baumgartner

Code Contributions and Help
Many thanks to the following people for very kindly donating time or code to Scrivener:

Martin Wierschin of Nisus[1], for his always generous help with the chicaneries of the text system and RTF.

Ken Thomases, for a lot of advice and pointers on the developer forums, especially when it came to modernising the code for Scrivener's tables and outlines.

Heinrich Gliesen - help with inline image scaling.

Jonathon Mah - help with the bubble highlights around comments and footnotes.

Andreas Mayer - NSBezierPath and table view extensions (http://www.harmless.de/cocoa.html).

Todd Ransom (author of StoryMill) - filtering and page number printing code.

Jesse Grosjean (author of TaskPaper and WriteRoom) at Hog Bay Software[2] - auto-saving code.

Split view code based on OASplitView from the excellent OmniGroup - http://www.omnigroup.com.

Andy Matuschak - Scrivener uses the superb Sparkle[3] framework for software updates, created by Andy Matuschak

Matt Gemmell - various code snippets from his source code site (http://mattgemmell.com/source/).

Philip Dow - help with the custom ruler code.

Wagner Truppel for help figuring out how to draw the diagonal status text in the corkboard.

---

[1] http://www.nisus.com

[2] http://www.hogbaysoftware.com

[3] http://sparkle.andymatuschak.org

Christian and Eric at Devon Technologies (http://www.devon-technologies.com) - help with keeping Scrivener in the background when the clippings services are used.

Malte Rosenau - for pointing me in the direction of the code I needed to import web pages with titles intact.

James Hoover, creator of Bean (http://www.bean-osx.com), for providing the basis of the "show invisibles" code and for sharing various other code snippets from his work on Bean.

Brent Simmons - the OPML importer code is based on a class created by Brent Simmons for NetNewswire (Copyright © 2002, Brent Simmons).

Shortcut Recorder is © Contributors of ShortcutRecorder (http://code.google.com/p/shortcutrecorder/).

Kino - the Snapshot comparison tool was inspired in large part by Kino's "Compare Documents" macro for Nisus Writer Pro[4].

Robert Warwick at http://www.codehackers.net - improvements to text table support are based on code Robert wrote for Stone Hill Invoicer.

Scrivener's crash reporter is based on UKCrashReporter by Uli Kusterer.

Improved PDF anti-aliasing was provided by a line of code from Skim - http://skim-app.sourceforge.net/.

The line numbering ruler view is based on code by by Paul Kim. (Copyright (c) 2008 Noodlesoft, LLC. All rights reserved.)

Daniel Jalkut of http://www.red-sweater.com for Mac App Store registration-related ideas.

Bryan D K Jones - VDKQueue (Copyright 2013 Bryan D K Jones)

The Courier Prime font is bundled under the SIL Open Font License (OFL)[5]. It was designed by Alan Dague-Greene for John August and released by Quote-Unquote Apps (http://quoteunquoteapps.com/courierprime).

Additional Images

Keith Blount
(And Apple, of course)

Additional Name Generator Lists

Cjmiltko - Polish names
Aleix Dorca - Catalan names

Exporters

Fletcher Penney - MultiMarkdown (http://fletcher.freeshell.org/wiki/MultiMarkdown)

John Gruber - Markdown & SmartyPants (http://daringfireball.net)

Joakim Hertze - SmartyPants localisations

Kee-Lin Steven Chan - ASCIIMathPHP (http://www.jcphysics.com/ASCIIMath/)

[4] http://www.nisus.com

[5] http://scripts.sil.org/OFL

Vasil Yaroshevich - XSLTMathML (http://www.raleigh.ru/MathML/mmltex/index.php?lang=en)

.docx, .doc and .odt conversion is performed using the Aspose.words Java converters[6] with Oracle's Java SE runtime environment (© Oracle).

MultiMarkdown

Fletcher Penney (http://fletcher.freeshell.org/wiki/MultiMarkdown)

App Store Receipt Validation

Receipt validation code for the Mac App Store was created by Graham Lee.

Final Draft

The .fdx and .fcf file formats are the property of Final Draft, Inc., and are used under licence.

Beta Testers

Too many to mention everyone, but a big thank you to all of you.

Special Thanks To

The redoubtable Douglas Davidson, Apple engineer and guru, for responding so helpfully to so many of my enquiries about the intricacies of the Cocoa text system.

Aki Inoue at Apple for advice about the word count code.

Fletcher Penney, for contributing so much time and effort in helping me get MultiMarkdown implemented in Scrivener.

Everyone on the Apple developer forums and lists for their help and support, with special mention to Bill Cheeseman, Malcom Crawford, Max and Marcus at The Soulmen[7] (authors of Ulysses), J. Nozzi at Bartas Technologies[8] (author of CopyWrite).

Stephen Kochan, author of Programming in Objective-C, for answering my questions when I was getting started.

Sophie Itali, for her torrent of useful ideas on improving MultiMarkdown integration.

And in the best Oscars-speech-style, thanks to Kurt Vonnegut for making me want to write and to my father for buying me a ZX Spectrum when I was a boy and thus forever turning me into a geek.

[6] http://www.aspose.com

[7] http://www.the-soulmen.com

[8] http://www.bartastechnologies.com