

# User Manual

*Scrivener 1.5 for Microsoft Windows*



Literature & Latte

August, 2013

Revision: 1.5.9-01c

© 2010–2013, Literature & Latte LTD. All rights reserved.

---

# Contents

<b>Contents</b>	<b>ii</b>	<b>II Preparation</b>	<b>40</b>
<b>I Introduction</b>	<b>1</b>	<b>7 Project Management</b>	<b>43</b>
1 Philosophy	2	7.1 Project Size Limitations . . .	44
2 About This Manual	5	7.2 Creating a New Project . . . .	44
2.1 Terms and Conventions . . .	6	7.3 Saving and Making Copies . .	45
2.2 Finding Things . . . . .	7	7.4 Opening Existing Projects . .	46
3 What's New	8	7.5 Moving and Deleting Projects	46
4 Installation	11	7.6 Splitting Projects Up . . . . .	47
4.1 Registering . . . . .	12	7.7 Templates . . . . .	47
4.2 Application Updates . . . . .	13	7.8 Backing Up Your Work . . .	51
4.3 Installing Scrivener on Linux	14	7.9 Tips for Working Cross-Platform . . . . .	54
4.4 Staying Informed . . . . .	14	<b>8 Setting Up the Binder</b>	<b>56</b>
4.5 Installing Scrivener on a Portable Flash Drive . . . . .	14	8.1 How the Binder is Organised	56
5 Interface	16	8.2 Binder Icons . . . . .	61
5.1 Interface Overview . . . . .	16	8.3 Binder Controls . . . . .	61
5.2 View Modes . . . . .	23	8.4 Using Collections . . . . .	63
5.3 Full Screen Editing Mode . .	26	<b>9 Navigating and Organising</b>	<b>69</b>
5.4 Selections . . . . .	26	9.1 General Navigation . . . . .	69
5.5 Interface Language . . . . .	27	9.2 Go To Menu . . . . .	70
6 Quick Tour	29	9.3 Controlling Binder Integration . . . . .	71
6.1 Project Format . . . . .	29	9.4 Linking Documents Together	73
6.2 The Main Window . . . . .	30	9.5 Layouts Panel . . . . .	74
6.3 Outlining Your Draft . . . . .	31	<b>10 Setting Up Meta-Data</b>	<b>76</b>
6.4 Scrivenings Mode . . . . .	35	10.1 Meta-Data Types . . . . .	76
6.5 More About Corkboards . . .	36	10.2 Project Keywords . . . . .	80
6.6 Compiling the Draft . . . . .	37	<b>11 Gathering Material</b>	<b>82</b>
		11.1 File Import . . . . .	82
		11.2 Linking to Files on Your Computer . . . . .	85
		11.3 Scratch Pad Panel . . . . .	86
		11.4 Text Appending Tools . . . .	87
		<b>12 Project Planning</b>	<b>88</b>
		12.1 The Corkboard . . . . .	88
		12.2 The Outliner . . . . .	93

12.3	Project Notes . . . . .	96	18.3	Document Support Panes . .	148
<b>13</b>	<b>Cloud Integration and Sharing</b>	<b>98</b>	18.4	Locking the Inspector . . . . .	151
13.1	Scrivener Everywhere . . . . .	98	<b>19</b>	<b>Scriptwriting</b>	<b>152</b>
<b>III</b>	<b>Writing</b>	<b>101</b>	19.1	Formatting a Script in Scrivener . . . . .	152
<b>14</b>	<b>The Editor</b>	<b>104</b>	19.2	Printing or Exporting a Script	154
14.1	Header Bar . . . . .	104	19.3	Working with Final Draft 8 .	154
14.2	Footer Bar . . . . .	107	19.4	Creating Your Own Script Formats . . . . .	156
14.3	Viewing Media in the Editor	108	19.5	Using Script Formatting for Other Purposes . . . . .	161
14.4	Splitting the Editor . . . . .	110	<b>20</b>	<b>Writing Tools</b>	<b>163</b>
<b>15</b>	<b>Writing and Editing</b>	<b>112</b>	20.1	Searching and Replacing . . .	163
15.1	Rich Text Editing Philosophy	112	20.2	Goals and Statistics Track- ing Tools . . . . .	169
15.2	Editing Basics . . . . .	113	20.3	The Name Generator . . . . .	173
15.3	Editing with Scrivener . . . . .	114	20.4	Bibliography Management . .	174
15.4	Formatting Tools . . . . .	120	20.5	Using Equations with MathType . . . . .	175
15.5	Inline Images . . . . .	125	20.6	Using Equations with MathML . . . . .	175
15.6	Using Snapshots . . . . .	127	<b>21</b>	<b>Using MultiMarkdown</b>	<b>177</b>
15.7	Auto-Completion . . . . .	130	21.1	What is Markdown? . . . . .	177
15.8	Editing Multiple Documents	130	21.2	What is MultiMarkdown? . .	178
<b>16</b>	<b>Full Screen</b>	<b>133</b>	21.3	Getting Started with MMD .	179
16.1	Using Background Image and Textures . . . . .	134	21.4	MMD and Scrivener . . . . .	180
16.2	The Control Strip . . . . .	134	21.5	Compiling . . . . .	185
16.3	Floating Inspector HUD . . .	135	21.6	MMD Meta-Data . . . . .	189
16.4	Customising Appearance . .	135	21.7	Further Information . . . . .	190
<b>17</b>	<b>Annotations and Footnotes</b>	<b>136</b>	<b>IV</b>	<b>Final Phases</b>	<b>191</b>
17.1	Inline Notation . . . . .	138	<b>22</b>	<b>Compiling the Draft</b>	<b>194</b>
17.2	Linked Notation . . . . .	140	22.1	Built-in Presets . . . . .	194
17.3	General Usage Tips for No- tation . . . . .	144	22.2	Available Compile Formats .	196
17.4	Text Colour and Highlights .	144	22.3	Compiling and Saving Set- tings . . . . .	196
17.5	Text Markings and Multi- Markdown . . . . .	145	22.4	Expanded Compile Interface	196
<b>18</b>	<b>Inspector</b>	<b>147</b>			
18.1	Synopsis Card . . . . .	147			
18.2	General Meta-Data View . . .	148			

22.5	Contents . . . . .	199	A.2	Edit Menu . . . . .	242
22.6	Separators . . . . .	201	A.3	View Menu . . . . .	246
22.7	Cover . . . . .	203	A.4	Project Menu . . . . .	254
22.8	Formatting . . . . .	205	A.5	Documents Menu . . . . .	255
22.9	Script Settings . . . . .	212	A.6	Format Menu . . . . .	259
22.10	Transformations . . . . .	212	A.7	Tools Menu . . . . .	263
22.11	Replacements . . . . .	213	A.8	Help Menu . . . . .	264
22.12	Footnotes/Comments . . . . .	216	<b>B Options</b>		<b>266</b>
22.13	Page Settings . . . . .	220	B.1	General . . . . .	267
22.14	Meta-Data . . . . .	222	B.2	Appearance . . . . .	269
<b>23</b>	<b>Exporting</b>	<b>226</b>	B.3	Corkboard . . . . .	273
23.1	Drag and Drop Between Projects . . . . .	226	B.4	Navigation . . . . .	274
23.2	Exporting Binder Files . . . . .	226	B.5	Editor . . . . .	275
23.3	Exporting to an Outliner with OPML . . . . .	227	B.6	Corrections . . . . .	277
23.4	Exporting Meta-Data to a Spreadsheet . . . . .	229	B.7	Keyboard . . . . .	279
			B.8	Import and Export . . . . .	280
			B.9	Backup . . . . .	281
<b>24</b>	<b>Printing</b>	<b>231</b>	<b>C Customising Toolbars</b>		<b>284</b>
24.1	Document Printing . . . . .	231	<b>D Placeholder Tags</b>		<b>285</b>
24.2	Printing Index Cards . . . . .	232	<b>E Tips and Tricks and Trouble</b>		<b>287</b>
24.3	Print Preview . . . . .	233	E.1	Getting Support . . . . .	287
24.4	Printing the Draft . . . . .	234	<b>F Project Bundle Format</b>		<b>288</b>
<b>V</b>	<b>Appendices</b>	<b>236</b>	F.1	The Files Folder . . . . .	289
<b>A</b>	<b>Menus and Keyboard Shortcuts</b>	<b>237</b>	F.2	The Settings Folder . . . . .	290
A.1	File Menu . . . . .	237	F.3	The Snapshots Folder . . . . .	290
			<b>G Legal</b>		<b>291</b>

**Part I**  
**Introduction**

# Philosophy

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

Most word processors and text editors aimed at writers assume the creative process will take place in linear form; that is, they assume that the writer knows how his or her work will begin and will start at the beginning and continue through until reaching the end, and for those that do work that way, they assume that a linear form is a useful format for a text that spans hundreds of pages. Planning and restructuring is therefore forced into a separate workflow—the writer must either plan before beginning and keep track of hundreds of different files using the computer or face the laborious task of cutting and pasting numerous chunks of text to restructure a long piece of work at the end. For shorter pieces of writing, this is not a massive problem, but for longer texts—such as novels or academic theses—the writer can often find him- or herself battling against the tools of their trade. What a word processor does get right is in not presuming anything of your working methods. It is, at the core, ruthlessly simple.

Scrivener is part of a new breed of writing software which focusses more on providing a solid open-ended writing *tool* (as the word processor does) rather than a writing *process*, which presumes a system upon you. As with a word processor, there are no chapters or sections or scenes; how you use the software determines these things, not some feature baked into the program. Scrivener’s design is fairly unique in that it easily provides a platform for writers of many philosophies and disciplines. It’s a tool that works equally well for a doctoral dissertation, as it does for writing a short story or a screenplay.

Often you will hear of “pantsers” vs. “plotsers” within fiction writing circles. Those that write by the seat of their pants and those that plot everything out before they start.

You will not find that this program urges you into one way of working or another. Rather you should find that its features and design provide a set of basic abilities which can amplify your method and make the mechanics of writing easier for you. If you start with a big pile of files that need to be sewn together, you'll probably find an emergent outline to be useful. If you make the outline first, you'll probably find the integrated outline-is-your-book approach to be useful. No longer will you need to maintain a separate outline document that needs to be kept up to date with the text of the book. The two will be one and the same. No more will you need elaborate spreadsheets to track crucial details. All of that can be woven right into the mechanical structure that is your book in this software. These principles extend to all forms of writing, from the novel to the legal brief to the documentary shot list.

Scrivener was developed because no other writing environment quite provided all of the features required for the sort of writing that demands easy access to research, a quick overview of numerous documents and the easy restructuring of swathes of text.

The main concepts of Scrivener can be summarised as follows:

- The software should allow the writer to use many of the formatting features familiar to word processors—different fonts, italics, paragraph styles and so on. It is up to the writer to avoid distractions, not the software.
- What looks good in print is not always the best way of viewing something on the screen: the software should allow the user to completely reformat the text on export or for printing without affecting the original (thus making it easy to export a manuscript in a format suitable for an editor and then export it just as easily in a format more suited to publishing on the internet).
- There should be a synopsis linked with each document that can be viewed with other synopses to get an overview of the project as a whole.
- Linked to the above, it should be easy to view the project as an outline and use drag and drop to restructure the project. Moving between the outline and the text itself should be fluid.
- It should be possible to view individual “documents” either as discrete chunks of text or combined with other (arbitrary) documents. Thus, for instance, a chapter could be viewed as a whole or each scene could be viewed individually.
- The user should be able to view more than one document at the same time—for instance, the end of one chapter alongside the beginning of the next, a character sketch alongside a scene in which that character appears, or a research document alongside the writing it is being used to support.

- The software should be capable of storing and displaying common research documents, because writers do not just use text—they also refer to images, web pages, recorded interviews and other media.

The overriding philosophy behind 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 summed up as follows:

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 particular 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 is thus a nonlinear writing tool that provides the writer with all of the features found in any text editor along with the functionality for “growing” your work organically within the program itself in a way which will become more useful the more you write, naturally and effortlessly.



# About This Manual

This manual has been written using Scrivener and is available in three different formats. If you purchased a boxed copy of Scrivener, you should have a printed version of it available for desktop reference. It can also be accessed as a PDF from the Help menu, under User Manual. You should also find two copies which have been specially formatted to be printed on either US Letter or A4 standard printer paper on our [website](#)<sup>1</sup>. Lastly, the Scrivener project used to author the manual can be downloaded from the Support section of the Literature & Latte website as a demonstration of writing in Scrivener. It has been written using the MultiMarkdown formatting syntax and so demonstrates that system as well.

**Annotating the PDF:** If you wish to make notes and annotate the PDF using software such as Reader or Skim, it is recommended you download a separate copy, or locate the PDF in the Scrivener installation and drag a copy out for your own personal use. 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.

The manual itself has been split into four primary sections and an appendix. The first section introduces Scrivener's terminology and interface and gives a quick tour for those who want to start writing as soon as possible. It is highly recommended that you also go through the separately available Interactive Tutorial (which can be accessed via the Help menu in Scrivener) to get the most out of Scrivener and eliminate confusion over some of its unique features, but the Quick Tour ([chapter 6](#)) should have you writing using Scrivener's basic main features within half an hour.

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, the next three sections take a pragmatic look at the primary stages of any major project,

---

<sup>1</sup><http://www.literatureandlatte.com/support.php>

Preparation ([Part II](#)), Writing ([Part III](#)), and Final Phases ([Part IV](#)), and detail Scrivener’s tools for aiding you through the drafting process and ultimately exporting and printing your work. In each section the features most useful to you during those phases will be explained in depth. If you want to find a specific feature, consult the appendices at the end of the manual for a cross-referenced list to every menu item and window.

## 2.1 Terms and Conventions

Whenever the documentation refers to an action that must be taken by you, the visible name for that action will be formatted **like so**. 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, sub-menu, and items. Example: To highlight a portion of selected text in yellow, invoke the **Format ▶ Highlight ▶ Yellow** command.

Keyboard shortcuts will use the following terms:

- **Alt-**: The Alternate key is generally labelled the Alt key on keyboards. Some laptops only have one Alt key on the left side, and most international keyboards have an “Alt Gr” key on the right side of the space bar. Use the left key for shortcuts.
- **Ctrl-**: Control is usually the leftmost modifier key in the spacebar row, abbreviated to “Ctrl”, with a copy on the right side of the spacebar as well.
- **Shift-**: The Shift keys are rarely used by themselves in shortcuts but are often used in combination with other modifier keys.
- **Meta-**: Otherwise known as the “Windows key”, this is the modifier key which has the Microsoft Windows logo printed on it.

When a keyboard shortcut requires a combination of modifier keys, they will be printed together. Example: **Ctrl-Shift-X**, which matches **File ▶ Export ▶ Files...**, means you should hold down both of these modifier keys and then tap the **E** key on your keyboard and let go of all three. The hyphen is used to separate the modifier keys from the target key and is not meant to be included in the shortcut itself. Sometimes shortcuts follow a sequence (or chord) of keys that must be pressed after the main shortcut has been pressed. An example is the spelling panel, **Ctrl-G, Ctrl-Q**. The comma separates the sequence. Type in the **Ctrl-B** first by itself, let go of both keys, and then tap **Ctrl-Q** immediately after.

Interface elements will be capitalised as proper nouns to help differentiate them from standard nouns when appropriate. A Collection is an interface element, while the word

“collection” can be used to indicate a casual grouping of items and not necessarily a formal Collection.

Some of the names for various elements within Scrivener are customisable on a per project basis, and how you name these may impact the interface. A good example are labels, a type of meta-data you can use to assign a colour to a piece of your draft. This can be renamed by you, and the interface will change to reflect your preference. In all cases, this documentation will refer to these malleable elements by their default names.

Screenshots and terminology will be based upon the Windows 7 Aero theme. Ordinarily this will not represent a discrepancy between the descriptions and what you see on your computer, but if you are using a Windows Classic theme or an older version of the operating system, you may see references to disclosure arrows in tree and outline views, where you see plus and minus signs.

In cases where file paths are printed, the standard Windows convention of printing paths will be used. An example might look like:

```
C:\Program Files\Scrivener\Scrivener.exe
```

## 2.2 Finding Things

Since this PDF has been birthed within the age of digital documentation, and has yet to make its way to print form, 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—in the PDF.

Alternatively, the appendices have been designed to use 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 and Keyboard Shortcuts ([Appendix A](#)). 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, a detailed table of contents has been provided at the beginning of the document.

# What's New

## 3.0.1 New in 1.5

### Data Import & Export

New import & export features have been added to allow for better integration with spreadsheets, mindmapping and outliners, using common standards for doing so. For spreadsheets, you can now export any Outliner view to a delineated (you can choose between comma, tab, or semi-colon separators). This could be a full output of every possible column, or just the columns you have visible in the Outliner. For applications based on structural hierarchies, like dedicated outliners, you can now import and export with OPML and MM (Freemind) files.

### Windows 8 Compliant

Scrivener is now compliant with Windows version 8 for the desktop. As with most standard software, it will run in desktop mode, rather than tile mode. At this time we have no plans to make a tile version, or a touch version for mobile devices running Windows 8.

### Marginalia

Now you get a choice between using inline annotations and footnotes or placing your notes into a new panel located in the Inspector. This not only lets you place long notes out of the way of the primary draft, but also works as a bookmarking system. Clicking on the notes in the Inspector column will automatically scroll your editor to its linked position in the document. Your old projects will continue using the inline system as they always have, but you can easily convert to the new system with a single menu command, or vice-versa. For more information, read Annotations and Footnotes ([chapter 17](#)).

These changes also allow for more extensive formatting controls of footnotes, annotations and comments, when compiling ([section 22.12](#)).

### Support for New Languages

Scrivener now has interface (menu and dialogue boxes) support for English, Chinese, German, and Italian. If you would prefer to use the software using one of these languages, read more about setting your preferred language ([section 5.5](#)).

## Continued Performance Improvements

Further efforts have been made to increase the performance of Scrivener on all recommended hardware. You should see swifter reaction to the formation of large Scrivenings sessions, and overall improved project load and save times.

## New in 1.2

### Performance Improvements

A great deal of effort has been put into making Scrivener faster all round. Performance has been completely reviewed and optimised in this release. We'll let you be the judge as to whether we have achieved that. We will be able to optimise editor performance considerably further once we implement core layout changes later this year.

### MultiMarkdown 3

MultiMarkdown integration support has been added, using MMD version 3.6. You can now compile directly to Flat ODT, OPML, HTML, plain MMD and L<sup>A</sup>T<sub>E</sub>X. The latter option has a new compile option pane which makes switching document classes as easy as a single drop-down menu.

### MathType and MathML Integration

Those who need to work with equations will be pleased to find that Scrivener 1.2 now offers integration with MathType, the premier equation editing and typesetting software by [Design Science](#)<sup>1</sup>. Adding equations is now as simple as using **Edit ▶ Insert ▶ MathType Equation**. Once added to your draft, you can easily revise the equation as necessary. MathML is available for systems that do not have MathType, and can also be used to edit MathType equations by hand.

### Final Draft 8 Support

Final Draft 8 FDX files can now be imported into the binder. The resulting imported file will be converted to Scrivener's scripting format. Simply drag and drop an FDX file into the binder to begin working on it in Scrivener, or use the **File ▶ Import ▶ Files...** menu command. When it comes time to go back to Final Draft, you can now compile scripts to the FDX format, for direct transfer to Final Draft 8 and higher.

### Improved Scratchpad

The Scratch Pad feature, which allows you to easily take notes even while working in other applications, has been improved. You can now keep multiple notepads in the

---

<sup>1</sup><http://www.dessci.com/en/products/mathtype/>

scratchpad window as well as easily grab screenshots and save them as pictures in the notes. Read more in Scratch Pad Panel ([section 11.3](#)).

# Installation

If you purchased the boxed version of Scrivener, insert the CD-ROM into your computer and wait for it the installation screen to appear. If you have disabled auto-load, you will need to run the installation program using Explorer.

For online downloadable purchases, grab the trial version from the Literature & Latte web site. The trial version can be unlocked at any time with the registration serial number that was sent to you when you purchased the application. When the executable finishes downloading, double-click on it in Explorer (if it hasn't opened for you automatically).

For both methods, once the installer starts, Windows will present a warning dialogue, asking if you wish to run the installer program. Verify that your program has been digitally signed by "Literature & Latte LTD". If it does not state that it is verified, you could be attempting to install an unauthorised copy from a third-party. Click the **No** button, and try downloading again from the official Scrivener homepage, if you continue to experience problems, contact us via the forums or through customer support. If it is verified, then click the **Yes** button to proceed.

The installer will start, and you should see the Scrivener icon and Literature & Latte logo. Click **Next** to continue and read the licence agreement. You will need to check "I accept the agreement" before the installer will let you proceed.

The next screen will ask where you wish to install Scrivener. At this point, the installer will verify if you have administrative rights on your computer. If you do, the default install location will be where all of your other program install, in the Program Files folder<sup>1</sup>. If you do not have administrative rights, you can still install Scrivener, and an alternate location where you have write permissions will be selected. By default this will be your Documents and Settings folder.

In all cases it is okay to change the defaults, though do read the special note about installing Scrivener directly on a portable USB flash drive, below. Once you are satisfied with the install location, click **Next** twice to confirm.

---

<sup>1</sup>On 64-bit Windows systems, this will be the x86 folder. On 64-bit Windows systems, this will be the x86 folder.

Once the progress bar fills, the **Next** button will again activate and you'll be given the option to load the introductory text, as well as immediately launch Scrivener and start working.

An application icon will be placed on your desktop for you, as with all shortcuts, it is safe to remove this if you'd prefer to use the task bar or Start Menu to launch Scrivener.

## 4.1 Registering

### 4.1.1 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, 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 22](#)) and Exporting ([chapter 23](#)), for details on getting your data out of the program.

### 4.1.2 Purchasing Scrivener

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. Clicking on the **Buy** button will take you to the Literature & Latte online store, where you can purchase a licence for Scrivener. You can also visit the online store at any time whilst using Scrivener in the **Help ► Purchase Scrivener...** menu.

When you buy a licence for Scrivener online, you will be e-mailed your unique serial number, which will have the following format<sup>2</sup>:

```
SCRWIN001-XXXX-XXXX-XXXX-XXXX-XXXX
```

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 if you ever have to register Scrivener again in the future. Remember that you may need to re-register at a later date, too—for instance when installing Scrivener on a new computer, or re-installing after wiping your hard drive for whatever reason—so it is very important to store this information safely.

If you have lost your serial number, a button will be provided in the registration window that will take you to our vendor's self-service support site.

---

<sup>2</sup>The number in the first block may be something other than '1', this is okay.



### 4.1.3 Registering Scrivener

After purchasing a licence and receiving your unique serial number, you can register Scrivener by clicking on the [Enter Licence...](#) button in the “Scrivener Trial Version” box that appears whenever you launch the program. Alternatively, you can choose [Registration...](#) from the Help menu. Either way, you will be presented with the registration dialogue box.

Enter the exact name under which you registered Scrivener (referred to as the “Serial Number Name” in the e-mail you received with your registration details) in the “Serial Number Name” box and your unique serial number in the “Serial Number” box and then click on the [Register](#) button. You must make sure that both the user name and serial number are exactly as they appear in the registration e-mail you received, or registration will fail. For this reason, it is probably best to copy and paste both the user name and serial number directly from your e-mail, though make sure you don’t accidentally select any extra padding space around the text.

Once Scrivener has been registered, there is no longer a time limit on its use and you will no longer see the nag box at startup—the application will launch immediately without any messages.

## 4.2 Application Updates

At any time, you can check for updates using the [Help ► Check for Updates...](#) menu command. This will require an Internet connection as it will need to check with our server in order to compare your version number with the latest available copy.

**Software vs. Your Work:** The advent of mobile operating systems for tablets and smart phones has unfortunately blurred the line between what a program is and what your data is. On these devices, deleting or re-installing an application can destroy your work.

Standard operating systems don’t work that way. They are considerably more robust and well designed for content creators, down to the very core of their technology and philosophy. Like all programs on your computer, Scrivener saves your work into files. You’ll learn more about the specifics of this in the coming sections, but for now simply know that there is absolutely no connection between the software and your hard work. 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; it will be accessible once the software is available again).

When new update is available, release notes will be displayed by default, and a button will take you straight to the download. Click [Download and Install Update](#) to fetch the

installer and start the upgrade process automatically. Installing an update is much like installing the application initially, so if you are unsure of how to proceed, review the prior section on installation.

If you are accessing the Internet through a proxy server (common in many offices) the automatic tool may not work. If you need to upgrade the application manually for whatever reason you can use a web browser to visit our main web site and check for the latest version, there. To compare with your currently installed version, see the [Help ► About Scrivener](#) menu item. The version number will be printed below the acknowledgements and credits.

### 4.3 Installing Scrivener on Linux

If you use the Linux operating system, you will find builds of Scrivener which should work on most modern distributions, from our forums. Since this release is, at the time of this writing unsupported, you are invited to participate in the forums and seek support from peers who are also using Scrivener on Linux. If for some reason the native build does not work with your distribution and you can find no tips for installing it, it is possible to run Scrivener through WINE with good results.

The Linux user forum can be found [here](#)<sup>3</sup>.

### 4.4 Staying Informed

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

The “Keep Up To Date...” window also be links to our Twitter feed and Facebook page. We frequently publish small tips and tricks, as well as links to helpful articles online, through these channels.

### 4.5 Installing Scrivener on a Portable Flash Drive

While it is possible to make a copy of your Scrivener application folder on a flash drive, at this time we do not recommend doing this for performance reasons. Flash drives are considerably slower than their hard drive counterparts, but if portability trumps

---

<sup>3</sup><http://www.literatureandlatte.com/forum/viewforum.php?f=33>

performance, it is safe to do so. We recommend installing Scrivener onto your hard drive following the instructions given, and then copying the Scrivener folder from Program Files (or your user folder) to the stick drive using Explorer, rather than using the installer directly on the stick drive. While possible, this method is not currently supported.

# Interface

The Scrivener interface has been carefully designed to cover a wide range of uses. At its most minimal, the interface could look no more complicated than a basic word processor like WordPad. In this section, we will go over some of the basic interface elements that will be present in nearly every project you work in. Advanced features will be gradually introduced in their own sections as they pertain to specific areas of the writing process.

## 5.1 Interface Overview

When you initially run Scrivener, you will be presented with a Getting Started and project template selection window. This is actually a full-featured launching pad for all of your future and existing projects. Whenever you are ready to start a new project (the details of which you'll read about later on), such as a new novel or paper for a journal, you'll use this interface to select a starting template, or a blank starter. You can hide the "Getting Started" interface in the **Options...** drop-down menu in the bottom of this window.

The remainder of this section briefly summarises the main components of a typical project window. Each area will be discussed in greater detail later on, and if you are interested in a particular feature you'll find cross-references to more exhaustive articles on these topics. 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. Skip forward to the Quick Tour ([chapter 6](#)) if you need instructions for doing so, but come back after you make a project, as it assumes a basic understanding of what we will cover in this chapter.

The main project interface comprises five main elements, not all of which are visible when a new project is created.

1. **Toolbar:** Frequently used tools displayed in a standard fashion which can be customised as you see fit.
2. **Binder:** A hierarchal list of all items in your project, including the manuscript; a general collection area for research materials, notes, and so forth; a trash can for

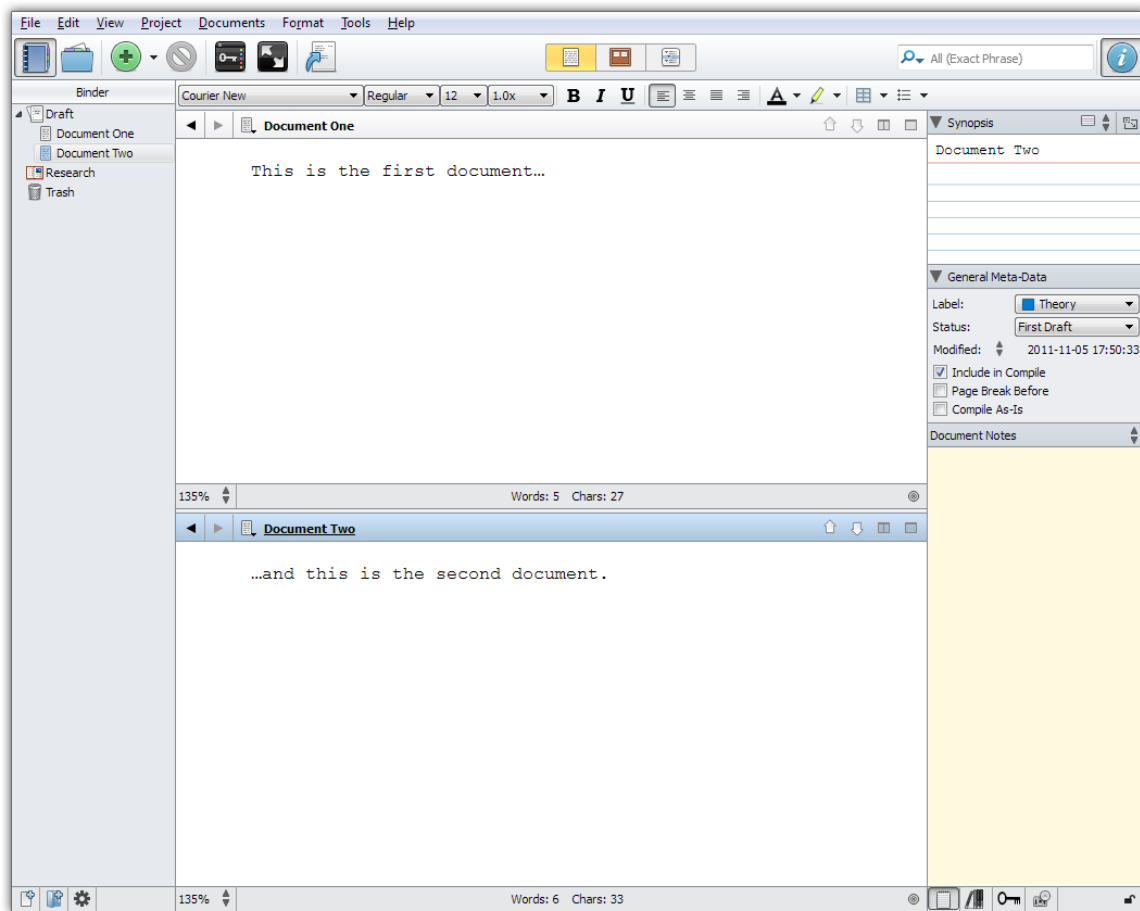


Figure 5.1: The Basic Interface

- collecting deleted items; and finally the top-level area where you can create your own files and folders.
3. **Format Bar:** Supplies frequently used text editing and formatting tools, available to both editor splits.
  4. **Editor:** The editor can be split into two panes which can show two parts of the same document, or two different items altogether. This is where all viewing and editing is done.
  5. **Inspector:** A feature-packed tool providing information about the currently active split. The inspector has four separate sub-panes, which address various aspects of the viewed resources. Not all of them are available for all media types.

Most of the main interface sections (binder, editors, and inspector) can be resized by dragging the divider line between them.

Each of these elements will now be explained in greater detail.

### 5.1.1 Toolbar

Scrivener has a customisable toolbar which includes many common functions, including a project search tool. To change the order, add, or remove icons, use the **Tools ► Customize Toolbars...** menu item.

It is also possible to hide the toolbar by using the **View ► Hide Toolbar** menu item, or by right-clicking on the toolbar and deselecting it from the provided list. Note that when the toolbar is hidden, you can still access project search via the **Edit ► Find ►** sub-menu, or **Ctrl-G, Ctrl-S**, which will provide a pop-up window featuring the same search controls available as on the toolbar.

### 5.1.2 The Binder

The binder is so-called because it really acts as the ring-binder for your project. It provides a hierarchical list of all items in your project. You can drag items around to restructure your project and double-click on any item to rename it (or use the **F2** shortcut key). Selecting an item in the binder displays its content in the editor pane that most recently had the keyboard focus (see below) and its meta-data in the Inspector (again, see below).

You can create as many folders and files as you wish in the binder, but there will always be three top-level folders that cannot be deleted (although they can be moved around within the top level and renamed):

**Draft** The Draft folder is where you place all of the pieces of text you want to include in your final manuscript. As such, the Draft folder is unique in that it can only hold text files and folders—it cannot directly hold image files, PDF files, QuickTime media or web pages as other folders can, though it is of course perfectly possible to embed figures into the text itself, much as you would in a word processor, at the editor level. All of the writing you want as part of your finished work should be placed inside the Draft folder, in the order you want it (which, of course, is easy to change by dragging and dropping). When you are ready to export or print your manuscript, use **File ► Compile....** This will combine all of the individual documents inside the Draft folder into one long document, formatted as you specify. Each file you place inside the Draft may represent a chapter, a scene, a paragraph or whatever you want.

Some templates will have renamed this folder to something else, but its function remains the same.

**Research** The Research folder can hold any type of file supported by Scrivener (text, image, media, web archive or PDF files). It provides a default place for non-text files to be imported and stored (although you can create other top-level folders for this purpose if you so wish).

Some templates will have renamed this folder to something else, but its function remains the same.

**Trash** When you delete an item from the binder, it is not really deleted but just moved to the Trash folder. You can drag items in the Trash folder back into the main binder, but you cannot create new items inside the Trash. To empty the Trash, use the **Project ► Empty Trash...** menu command. This will permanently delete all items inside the Trash folder (and cannot be undone).

The Collection tab interface can be toggled on and off with the associated toolbar button, or the **View ► Collections ► Show Collections** menu item. You can organise material into tabs you create, or save common searches as dynamic collection. In most cases, if this manual refers to having a “binder selection”, it is often accurate to assume that applies to collections and search results as well. Collections and search results can also be dismissed by clicking the **X** button in the footer bar for the Binder. Due to the similarity of these views, the entire sidebar is often referred to in this manual as “the binder sidebar”. This is meant to indicate that you needn’t take the referenced action only in the Binder tab itself, but in any view that presents itself in this sidebar, such as the Search Results lists.

See Also:

- Setting Up the Binder ([chapter 8](#))
- Navigating and Organising ([chapter 9](#))

### 5.1.3 Format Bar

The Format Bar stretches across the top of the main application view, right below the toolbar, and provides easy access to the most common text formatting tools. It can be selectively hidden when not needed with the **Format ► Hide Format Bar** menu command. For more information, read [The Format Bar \(subsection 15.4.2\)](#).

### 5.1.4 Editor

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 whenever you click an item 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 in to shortly, and is a capable multimedia 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. The editor can also be split, either vertically or horizontally, giving you two independent working areas at once.

At the top of each editor is a header bar or view. This contains navigation buttons (the arrows) which allow you to navigate backwards and forwards through the document history. Next to the navigation button pair is the title of the document along with its icon. Clicking on the icon will display a pop-up menu that collects together some common menu commands. The entire header bar has three colour status modes:

- Plain grey is the default, and is what you will see unless you have split the editor.
- Blue appears when you have split editors, and indicates the *active* editor.
- Mauve appears when the editor has been locked (**Ctrl-Shift-L**), which inhibits it from receiving binder clicks.

On the right side are two vertical arrows which allow you to navigate through the binder in a flat list fashion. You can also use the keyboard shortcuts, **Alt-Shift-UpArrow** and **Alt-Shift-DownArrow**. Using these navigation tools will also focus the binder on what you are selecting (and is one of the few instances in which an action taken in the editor will change the Binder selection).

On the far right of the header view you will find two buttons for controlling the editor splits. To create a split, click the one which represents the layout you prefer. Once a split is open, the corresponding opposing button will be replaced with a blank button.

Clicking on this button when both editors are visible will cause the editor associated with that Header Bar to take over and the other editor to close.

Below each editor pane is a Footer Bar. This will change depending on the type of document visible and the current editor mode. When a standard text document is being viewed, for instance, it will display the word and character count along with a pop-up button for changing the text scale and a button for setting document targets. In scriptwriting mode, it will provide shortcuts for accessing various script entry macros. Various types of multimedia, and PDF, will display their own appropriate relevant information in the footer bar. Corkboard and Outliner group view modes have their own footer bar functions, too.

See Also:

- View Modes ([section 5.2](#)): for further information on the various group and text view modes available.



- Viewing Media in the Editor (section 14.3): for details on how various read-only media are displayed in the editor.
- Header Bar (section 14.1): for more information on the informative strip along the top of the editor.
- Footer Bar (section 14.2): on how to use the strip along the bottom of the editor, which will change according to what type of document you are viewing.

### 5.1.5 The Inspector

The Inspector is not always visible by default in new projects, but will always be on the right side of the project window when it is visible. Its purpose is to display all meta-data associated with the document shown in the editor pane that most was recently viewed in either editor. Note that the special top-level folders in the binder—Draft, Research and Trash—have no meta-data associated with them, and therefore the inspector just shows the project notes or references when these items are selected. More on that later; for now, if the inspector appears suspiciously empty, know you have an item selected that does not support meta-data.

The inspector has four different sub-panels, which can all be accessed via buttons along the very bottom of the Inspector. If an associated panel has had data entered into it, an asterisk will appear in the upper-right corner of the button. The sub-panes are listed in the order they appear:

1. Synopsis, Meta-data and Notes
2. Synopsis, Meta-data and References
3. Synopsis, Meta-data and Keywords
4. Snapshots

The lock icon at the end locks the inspector so that you can reference the material in it even when switching to a different editor split.

The first three sub-panes include the synopsis card, and general meta-data in addition to their specialised information. These initial sections can be collapsed by clicking the disclosure arrow on the left side of the header bar for each. The two common meta-data sections are as follows:

**Synopsis & Image** The top frame shows either the synopsis for the binder item, which is what also appears in the index card representation of the item in corkboard mode,

among other places. Using the top drop-down menu, you can optionally also attach an image to the item to represent it in Corkboard mode, instead of a text synopsis. Media files by default will display an image synopsis, but you can use the drop-down menu to switch to text if you prefer.

**General Meta-Data** The middle frame of the inspector shows basic meta-data for the current document. This includes the label and status (which can be renamed to something more appropriate to the current project), the modified and created date of the document (click on the arrows to switch between the modified and created date) and checkboxes that affect how (and whether) the document will appear as part of the compiled draft for exporting or printing. For more information on the types of meta-data available, read Setting Up Meta-Data ([chapter 10](#)).

The Notes, References and Keywords panels all display the above two components. They are described as follows:

**Notes** Displays a notepad for the document. Each document has an auxiliary notepad associated with it; it only appears while you are viewing that item. Clicking the header bar of the notes area will bring up a pop-up menu from which you can select “Document Notes” or “Project Notes”, the latter of which is global to the project and accessible from any part of the project. For more information, read Document and Project Notes ([subsection 18.3.1](#)).

**References** The References table lets you to hold links to related material within the project itself. Links can also point elsewhere on your hard drive or on the Internet. Click the header bar to switch between document references and global project references. Use the latter to make links available throughout the project. For more information, read References ([subsection 18.3.2](#)).

**Keywords** Displays a list of keywords associated with the selected document. Keywords can be used to tag your document. Use the + and - buttons to add and remove keywords, or the gear menu to access the project’s central keyword list. Read more about them in Keywords Pane ([subsection 18.3.3](#)) and Using Keywords ([subsection 10.1.4](#)).

The last sub-pane, Snapshots, take up the entire inspector.

**Snapshots** An integrated view which displays all of the document’s snapshots and the contents of the selected snapshot in the text area below. Snapshots can be created (from the state of the current text) or deleted with the + and - buttons in the header

bar. The **Roll Back** button lets you revert the main editor to an older version of the text. For more information on using this, see Snapshots Pane (subsection 18.3.4), and Using Snapshots (section 15.6) for more general information on Snapshots.

## 5.2 View Modes

There are three primary “group view modes” available in Scrivener. View modes give you a powerful way to view and modify multiple documents at once in a variety of fashions, each tailored to different tasks. View modes are generally only triggered when the editor is asked to view more than one document at a time. This can happen if you Ctrl-click on more than one binder item or even if you just click on a folder. It is also possible to manually enable a view mode, allowing you to easily flesh out a single document with new index cards beneath it.

For the most part, 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 editor. If you then click on the group *containing* that file, it will automatically switch to your preferred group view mode, the one which you used last (corkboard by default).

An exception to this rule is when the view mode has been explicitly set to single text file mode. If you intentionally disable 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.

A key principle is that view modes are just that. They are alternative ways of viewing a binder item and thus, like the binder, are methods for organisation. You can think of view modes as being like an elaborate extension of the binder, and in that sense they work closely with it and display information in the same top-down fashion. You can drag items from a corkboard into the binder to move those cards, or vice versa.

All file or folder item in the binder can *both* contain text and children. This is a departure 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. Consequently, even though it selects the most appropriate view mode depending upon what you click on, you can always override the default. For folders, you can turn off the current view mode and read or add text to it. For files you can select a view mode manually and then add items to them as if they were folders. When performing the latter, you will most likely 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. This is a great way of expanding scenes and fleshing out rough ideas, so take a moment to play around with the view mode buttons in the toolbar, or their corresponding shortcut



Figure 5.2: Segmented control in group mode with Scrivenings mode selected.

keys, to see how you can get the most out of Scrivener’s unique file and folder structure.

The default view mode will prefer whatever mode you have selected last. If you choose outliner, then this is the mode that all containers will default to when clicked upon. It will remain this way until you manually change modes again.

The section on Project Planning ([chapter 12](#)) goes into greater detail on how to use corkboard and outliner modes to their maximum power. The section on Editing Multiple Documents ([section 15.8](#)) will cover the unique Scrivenings mode in detail.

### 5.2.1 The Group Mode Toolbar Control

Included in the default application toolbar is a so-called “segmented control”. It is in the middle of the window, and contains three buttons embedded in a single row ([Figure 5.2](#)).

The figure depicts the control when a group of items or a container has been selected. When a single item has been selected the icon on the far left will depict a single page.

The control displays the active view mode with a shaded background. This is the last view mode that you picked, and it means that editor will always use that method to display groups of content until you change it again while viewing a group. To change a mode, simply click on any of the other buttons in the control. The view for the active editor will immediately change, and simultaneously set your preference for that view mode for that project.

You can also click on the shaded view mode that is already selected to intentionally disable it and view the text of the *container* that you are viewing. If you have clicked on a folder and are viewing its index cards on a corkboard, clicking the corkboard button again will switch to viewing the text of the folder. This is considered to be a view mode setting as well (or rather the absence of a preference). All future clicks to container items will load the text editor in single file mode until you again choose a view mode from the control.

These three options can also be accessed via 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, where invoking corkboard while you are already viewing an item as a corkboard will switch that view mode off and go to single text mode.

### 5.2.2 Corkboard

Scrivener has made the corkboard metaphor popular in modern writing software. The concept of representing many documents as index cards, letting you move them around to restructure your book, or opening them up to edit their text, is quite useful.

The important thing to realise with the corkboard is that each of the displayed index cards represent files or folders 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 meta-data using real-world metaphors as well. Just remember, cards are documents, and conversely documents can all be viewed as cards.

The corkboard view is rich with features, to read more about it and how to best use it, see [The Corkboard \(section 12.1\)](#).

### 5.2.3 Outliner

While the binder is in a sense a very simple outliner, there are often times when you need to view your book, or parts of it, in a much larger context and with a greater degree of information at your disposal. The outliner view lets you see your book structure in a standard outline format with indenting used to indicate depth. It also has the ability to display many built-in aspects of your documents: label colour, word counts, modification dates, keywords, and much more.

Since the outliner provides feedback for so much meta-data, it is a great mode 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. Whether you prefer the analogue experience of the corkboard or the more literal but expanded information in the outliner is up to you.

As with the corkboard, there are many features and uses for the outliner. To read more about it, see [The Outliner \(section 12.2\)](#).

### 5.2.4 Scrivenings

This is a unique view mode in that it works not only with the viewed items themselves, but specifically with their text, and is consequently a tool for working with text (and folder) items, rather than multimedia files. All of the text content (even empty documents) will be stacked together as if on a long spool of paper, letting you read through large sections of your book at once, no matter where they are located. As you make adjustments to the text in this view, each of the corresponding documents will be updated as you work. 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.

Scrivenings (frequently referred to as a “Scrivenings session”) are temporary editing sessions. 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.

For more information on how to best take advantage of this editing mode, see Editing Multiple Documents ([section 15.8](#)). We will also explore a simple usage of it in the following Quick Tour section.

## 5.3 Full Screen Editing Mode

Full Screen mode is a special interface which will blot out 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 most aspects of the inspector, its default state is to simply display the text contents of the selected document (or documents, if you are viewing an combined text) in a “page” in the middle of your screen. Read more about Full Screen ([chapter 16](#)).

## 5.4 Selections

Selection and “Focus” are often important in Scrivener, as there are features which work when only certain types of things are selected in the interface. A good example of this is the [Format ► Convert ► To Uppercase](#) menu command. It has no meaning when a selection of index cards is currently active, rather than some 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.
- **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 ► Formatting to Default Text Style](#), which would reset the text formatting for any selected documents in the active selection.
- **Inactive selection:** A selection can be inactive. If you select the Draft folder in your Binder and then click in the editor, the selection will become dimmer (or grey, depending on your settings). Inactive selections are seldom used, but there are a few cases, especially when exporting.

- Active text editor: the main text editor must be active (the insertion caret must be blinking within it or text must be selected). In cases where the interface is split into two editors, this not only means any editor must be selected, but the editor you wish to perform the function in. Some commands do not require text to be selected, only an editor, others will perform universal actions if no text is selected, but an editor is active; others will refuse to work unless you have a selection.

## 5.5 Interface Language

We are proud to present Scrivener in a growing number of different languages. If you would prefer to use the software in another native language, you will want to use the **Tools ► Options...** menu command. From there, click on the “General” icon in the left list, and select your preferred language from the drop-down menu at the bottom of the window. Chances are, if you have Windows set up to display itself in that language already, this will not need to be adjusted. If you wish to return Scrivener to the system default, you can click the **Reset** button to the right of the menu selector.

You will need to restart Scrivener in order for the language settings to be reloaded.

We currently provide translation for the following languages:

- English,
- Chinese (Traditional),
- German,
- Turkish,
- Dutch,
- Bulgarian,
- Italian.

And for those time travellers from the future, we hope to make things easier for you with our Science Fiction themed interface.

In addition, we expect these languages to be ready soon:

- French,
- Spanish,
- Russian,

• Polish.

If you have translation experience and would like to see your language available in Scrivener, please contact [lee@literatureandlatte.com](mailto:lee@literatureandlatte.com).



# Quick Tour

Before going over this section, it is suggested that you skim through the prior chapter, introducing the interface, if you haven't done so already.

This chapter will get you up and running with a simple sample project and will introduce a few of the most basic concepts you'll need in order to start working in thirty minutes or less. Scrivener is a complex program with a lot of potential depth, but we hope this section will demonstrate that despite this, it is also very straightforward and intuitive, so its complexity can be gradually learned as required. There is often a temptation to dive in and learn as much as possible about a new tool, but if you are here for the same reason as most people who've picked up this software, it is because you are serious about writing, and so the most important thing this manual can do is get you doing that as quickly as possible, in a fashion which is not too dissimilar from what you may already be used to.

We will go over the following tasks:

- Creating a new project from a blank template
- Navigating the binder and adding new documents to the Draft folder
- Working with the corkboard and Scrivenings text view
- Compiling the document into an RTF file

If you need to learn more than the above, we strongly suggest going through the full tutorial, which can be found in the Help menu.

## 6.1 Project Format

Scrivener is a project based application, which means that it stores your projects as separate files, rather than working like a database which hides the location from you and loads everything automatically when you start up the program, or a file based program like Word, which creates an empty document whenever you launch it. The intended use of a project is to store everything relating to a single major work, be it your next novel or

screenplay, a dissertation for your masters, or a serial collection of articles for a magazine or blog.

It is recommended that for each new “real world” project you embark on, a new project file should be created<sup>1</sup>. When you create a new project, Scrivener asks where you want to save the file. By default this location is your personal Documents folder, but you can choose to organise your projects on the computer however you like. It is a good idea to choose a place you will remember. Even though Scrivener provides convenience functions for finding the projects you’ve worked on most recently, and will even reopen your last session for you if you choose, these tools should not be relied upon as your sole reference point for where projects exist.

Okay, enough theory. Let’s create a new project right now. Invoke the **File ► New Project...** menu command (**Ctrl-G, Ctrl-N**) if you do not already see the template selection interface (Figure 6.1). In the left list of categories, click on the “Blank” entry (if necessary), and then select the blank starter project. Choose a location you will remember (the default is your Documents folder, but you might wish to create a special sub-folder just for Scrivener projects), and give the project a name in the “Save As” field. This name is what will be used on your disk, and so should be memorable and placed in a location you’ll easily find in the future. Click the **Create** button once you’ve decided on a file name and location.

## 6.2 The Main Window

The project will be created, and you’ll be presented with a fresh Scrivener project window (Figure 6.2).

The left sidebar panel is your project binder, where everything stored or created in your project will be represented in an outline structure. On the right side of the window we have the editor (which is more than just a text editor, as we will shortly see). In most cases, what you click on in the binder will be displayed on this side of the window. Along the top, above the editor, is the Format Bar, which provides access to common formatting tools such as font selection, justification, line spacing, and so on. If you cannot see all of the tools, try expanding your window horizontally. If you want to see tab stops and margins, use the menu command **Format ► Show Ruler** or press **Ctrl-Shift-R**.

Directly below the Format Bar is the editor Header Bar. The title of the current document is printed here (“Untitled” since we haven’t named it yet). There are also some buttons for moving around in your project and adjusting the editor view, but we’ll get

---

<sup>1</sup>There are, of course, many exceptions to this. A series of articles for a magazine might work very well together in a single project.

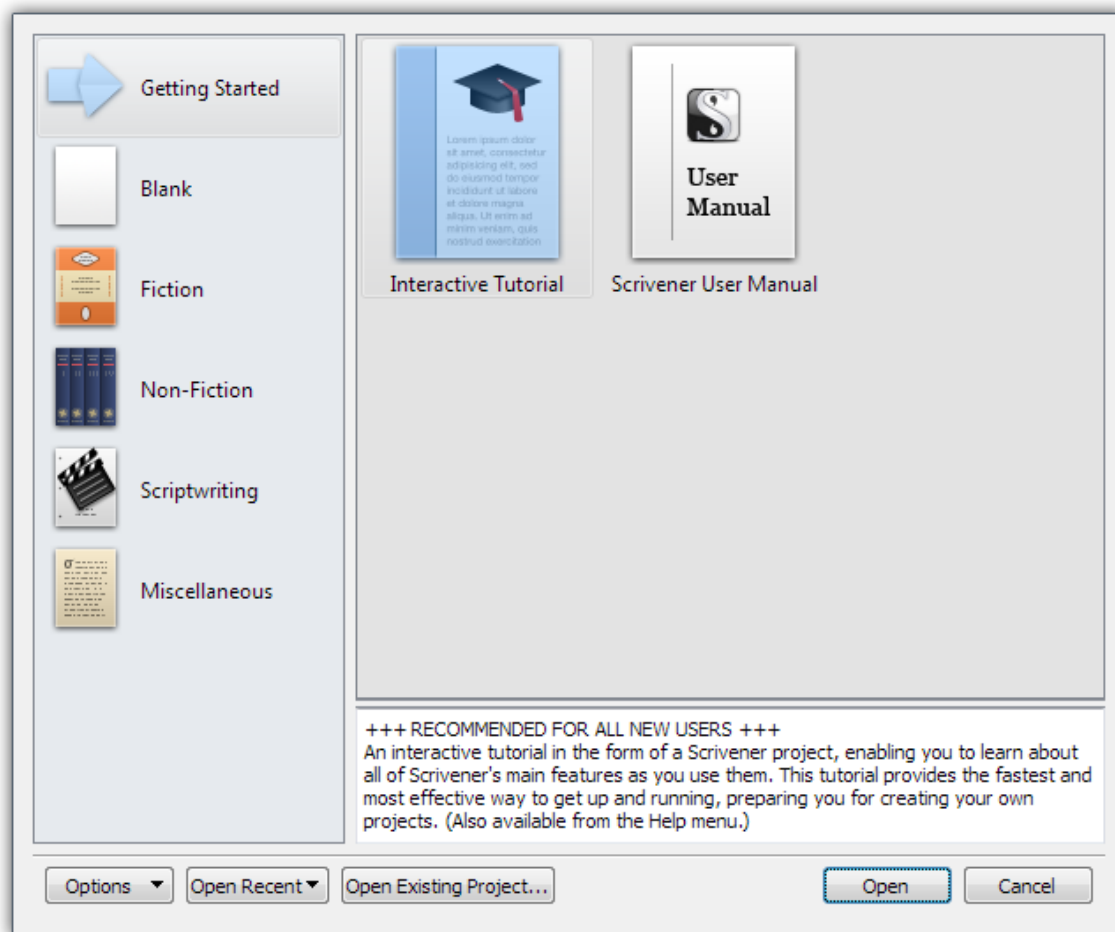


Figure 6.1: Template Wizard

to that later. The important thing to notice right now is that the Untitled document has been selected in the binder, and that this selection is being viewed in the editor.

Below the editor text area is the Footer Bar which contains information about the editor session, such as your current word and character count, and often contains useful buttons for adjusting the view.

The cursor is blinking in the text editor, and with zero words so far, it's time to get to work!

## 6.3 Outlining Your Draft

Why do that when we can procrastinate for another minute, though? Take a look at the binder. Let's discuss how best to use Scrivener's Draft folder. Unlike a word proces-

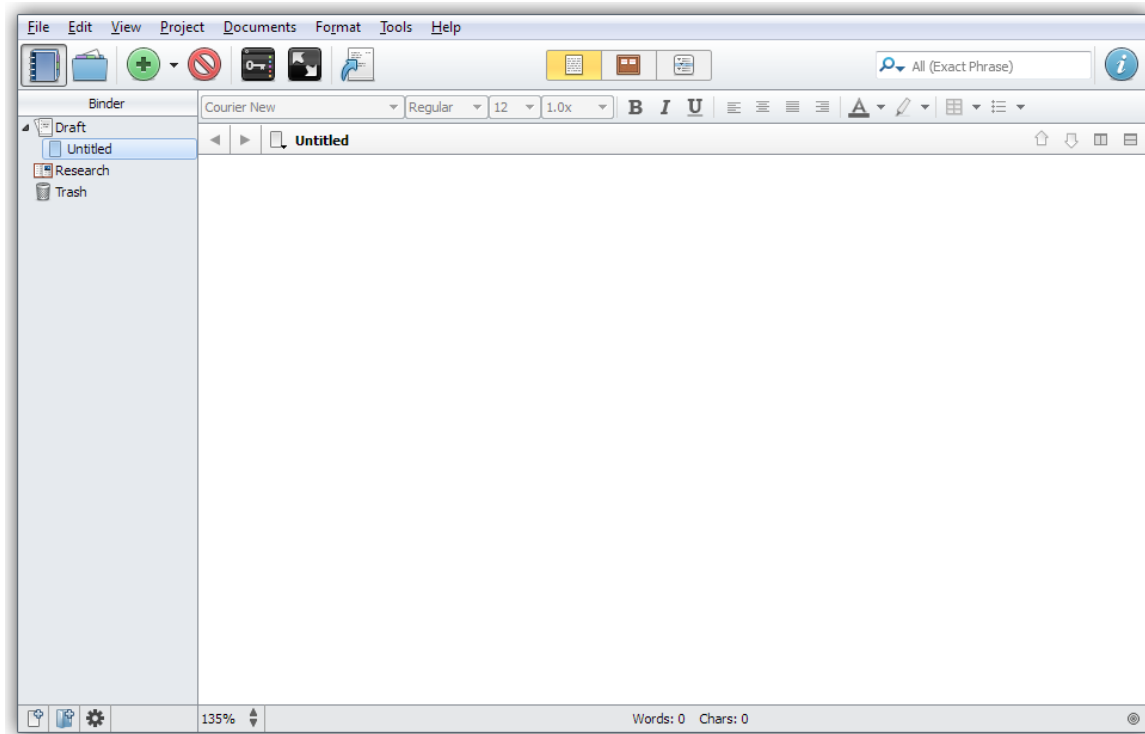


Figure 6.2: A typical new blank project

For example, you could make a new file for each chapter, or take that even further and make a new file for each scene *within* those chapters. It is totally up to you, and unlike in most programs, you won't be penalised for breaking things down as small as you desire. The binder displays these pieces like an indented outline. It is easy to change your mind later, so don't get hung up on which is "right", that may very well change for each project, or as you progress in your familiarity with the program.

Let's try it. If you haven't touched anything yet, the cursor should still be blinking in the main text editor (if it isn't, just click in the middle of this white text area with your mouse). Type a few words in here and then press the **Return** key to create a new paragraph.

Glance up to the window title bar and note the name of your project. You should see a small asterisk after the name. This means your project hasn't been saved yet. If you don't see an asterisk, try typing another word into the editor while watching for it. By the time you finish examining it, the asterisk ought to disappear—your project has been saved.

**The Implications of Auto-Save:** Whenever you pause for two seconds, Scrivener’s auto-save feature will kick in and save all your recent work. This is a very key thing to be aware of, because many people have cultivated the habit of relying upon their last save point as being a bit of an anchor point. They can write freely and in an experimental fashion, and if what they’ve done displeases them they can just revert to the saved copy on the disk. The good news is that Scrivener is designed to work with this method, and in fact its way of doing so is in general superior to using the traditional model, because whenever you put down an anchor point, or a Snapshot, in Scrivener, you leave a trail of them. You can view them all in a list later on, preview their text, and roll back to any point in a document’s history. This is a little out of scope for this quick tour, so for now put it on your list of things to look into if this is how you work, and when you’re ready to start writing, check out Using Snapshots (section 15.6).

You might have noticed that the title of your document (printed above the editor, as well as in the binder) changed when this happened. If you leave a document as “Untitled”, the first line will be used as the title (or the first few words if it is a long line, like a paragraph). This will only happen once, and you can always change the automatic title to something more descriptive if you wish (you can also turn this feature off).

Let’s make a new document in the binder. There are a number of ways to do this, but first let’s try the most direct method: click the large green **+** button in the toolbar just once.

The cursor will flip over to the binder and let you type in the name of the new document. Type something here, like “Masked Avenger”, and press **Enter** to confirm. You’ll notice that the content you just typed into the editor has disappeared. This is because you are now viewing “Masked Avenger”. Let’s fix the name of the first document while we are in the binder. Press the **UpArrow** on your keyboard once. Immediately, you should see the words you just typed re-appear in the editor. That is because you are now viewing the first starter document.

To change the name, just press **F2** and type in “After Math”, once again pressing **Enter** to confirm the name change.

You might notice something going on with the icons in the binder (Figure 6.3). The document “After Math” looks like a page of text, but “Masked Avenger” just looks like a blank sheet of paper. That’s because we haven’t typed anything into the editor for “Masked Avenger” yet. Scrivener lets you know which parts of your project are still placeholders.

For now, let’s use another method for adding new documents. When you are in the binder, there is a simple way to add new ones below the current document. Select

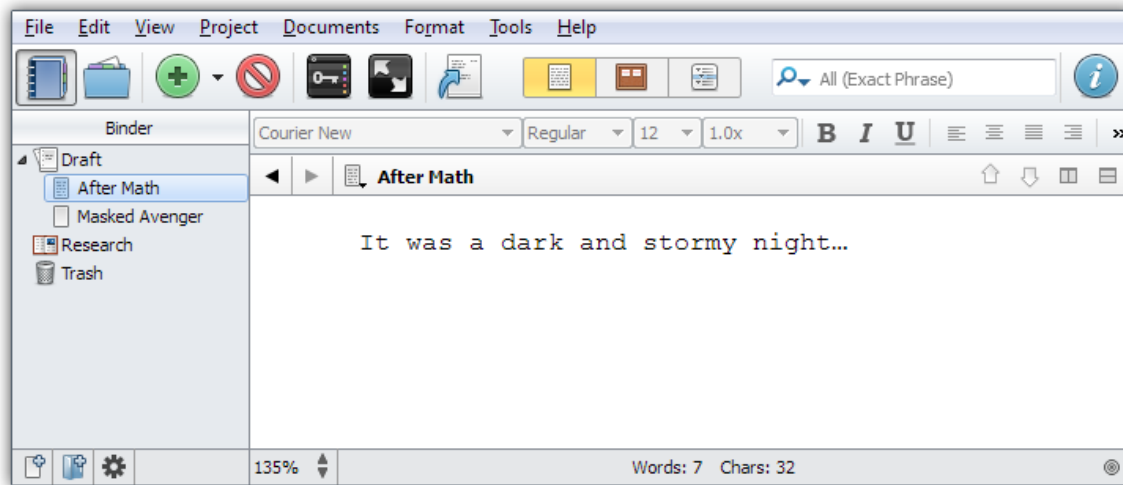


Figure 6.3: Progress so far

“Masked Avenger”, and just press **Enter**. Call this one “Wineries”. You can quickly build outlines this way.

Now we will take a look at a great way to get an overview of your Draft. Either click on “Draft” in the binder with your mouse, or simply press the **Left Arrow** to jump up to it. That key selects the “parent” of whatever item you currently have selected, in this case, the Draft. When you view a folder in Scrivener, by default it will automatically switch to showing you the corkboard in the editor area. You should see three index cards, one for each of the items you’ve created (sometimes referred to as ‘Scrivenings’), but the cards will be blank; this is because they display the *synopsis* for a document, not its internal text. Double-click in the empty ruled text area on the “Wineries” card and type in a short synopsis for this section of the book. When finished, click outside of the card or press **Esc** to confirm.

At this point, if you look in the binder you’ll see a third icon type that looks like a small index card, next to “Wineries”. This new one indicates that you’ve typed in a synopsis, but haven’t yet typed in any text for the section. Once you enter text into the editor for the “Wineries” section, the icon will be replaced with the one you see for “After Math”. Some people like to work in a top-down fashion, building a conceptual outline and then filling in what they intend for each section in the synopsis. These three levels of icons give you an overall idea of your level of progress with this task.

We’ll try that now. Load the “Wineries” card by double-clicking on the icon next to the title. If you can’t find the right spot, just click on it in the binder. Type in a short sentence and observe that, as predicted, the icon has now changed to match the “After

Math” section.

You might be wondering how much text each section should hold. The answer is largely up to you. With a word processor, it is often convenient to work in longer documents because lots of small windows can create unmanageable clutter, but Scrivener allows you to break things down into smaller pieces, since it organises all of these files for you and keeps them in a logical order in the binder. In fact some people break things down into pieces as small as paragraphs, others stick with chapter-sized chunks. You can use index cards to represent beats, scenes, whole chapters, or even a mix of these depending on how much attention you need to give that portion of the book. It’s all very flexible, and in many ways you can use the Scrivener outline to structure things according to how you think, not according to how the book will appear. That may change later on when you are ready to compile, but this can be ignored for now; there are good tools for breaking things down and merging things back together at whim. Working in a fluid fashion at every stage of your project is more important than trying to force yourself to work toward a strict goal.

## 6.4 Scrivenings Mode

Scrivener contains features intended to make that balance a little easier to work with. Unlike most applications of this type, you can choose to view many different sections at once and edit them as if they were a single file without actually merging them all together permanently. Doing this is easy; let’s try it now. Select the Draft item in the binder again, this time by clicking on it once with the mouse. You’ll get a corkboard again; now invoke **View ► Scrivenings**, or press **Ctrl-I**. You are now viewing the collection of documents as a Scrivenings session, rather than a Corkboard.

Each section will be divided with a horizontal rule, letting you know where one ends and the next begins. Try clicking in the blank area between the two sections you’ve already added text to. When working in Scrivenings, you might wish to open the Inspector by clicking the blue ‘i’ icon in the toolbar on the far right; or pressing **Ctrl-Shift-I**. A preview of the index card will appear at the top. This will change when you switch sections in the Scrivenings session, so you can more easily see where you are. Try clicking around a bit or moving the cursor between sections, and then return to the blank “Masked Avenger” section. Add some text by typing normally, then select “Masked Avenger” in the binder all by itself. Note that your edits have been made directly to the document. It couldn’t be easier.

Now we will do something else that is impossible in many programs. Right-click on “After Math” and select **Convert to Folder**. The icon will change to a blue folder with a small page in the corner. That page means the folder has text stored in it, in the same

way a text file might, and indicates that the text you added to this document earlier is still there. This differs from stating that a folder has text items as children beneath it. The page badge will only be applied if the folder item itself has text content. In Scrivener, folders and text files are pretty much the same thing; it's yet another demonstration of how you can work fluidly now, and worry about details later.

Let's move that text into a new child document, with the "After Math" item selected in the Binder, press **Ctrl-N**, which is another way to create new documents. When a folder is selected in the Binder, new documents will automatically be assigned to it as children. Call this new document, "Scene A", and then click on "After Math" again to select the folder and view it as a Scrivenings session. The first line of text you typed will be visible above the divider line, we want to move it below the divider line, moving it into "Scene A". You can select paragraphs in Scrivener by triple-clicking anywhere within them. Try that now, and then cut and paste the text below the divider. Notice that the page icon badge in the Binder has been removed from the folder, and the Scene A icon now has a page of typed text in it. Once again, anything you do in a Scrivenings session is instantly made to the underlying sections it represents.

No doubt you can see that Scrivener is very comfortable with the notion of splitting things up, and with this feature you can feel free to work in small portions without losing sight of the big picture.

## 6.5 More About Corkboards

For our next experiment, select the **View ► Corkboard** menu item to return to the corkboard view. What you are seeing here is an important concept in how view modes integrate together. The editor was in Scrivenings mode, giving you access to each file's text content, but is essentially focussed on the "After Math" folder. When you switched to using Corkboard mode nothing changed about that criteria: what you are looking at is the corkboard for the "After Math" folder, which currently only contains one card, "Scene A". Try clicking the Draft again in the binder. Now you are viewing the index cards that belong to the Draft. "Scene A" has disappeared, because it is a child item of "After Math", not the Draft, but the "After Math" folder looks like a stack of cards. Folders will always look this way, even if they don't have cards within them.

- Corkboard mode will only show you one layer of depth in your outline at once, but you can tell when index cards are folders or files that contain subdocuments by the stacked appearance of their index card. Double-click on the "After Math" index card's icon and you'll return to where you were at the start of this paragraph. You've navigated "into" the "After Math" corkboard.



- Outliner mode, which we haven't explored yet, is a bit more like the Binder in that you can see items at all levels of depth beneath the current selection, but also has the ability to show many columns of tabular data, and is great for tracking word counts per section, just to name one usage.
- Scrivenings mode usually includes all text content beneath the current selection, no matter how deep they are in the outline.

There is one main exception to these rules of thumb: Multiple Selections, which happen when you select more than one binder item at once, but for simple single selections, the above guidelines will help you decide which view mode to use, depending on what you need to do.

You might be wondering about that oddly titled “Wineries” card. Let's move it to the Research section of your Binder, since in our hypothetical little universe here, it would be used to collect research on various wineries that the characters are involved with. There are multiple ways of moving items around in the outline, but the most familiar will be drag and drop. Select the Draft folder again, then click and hold on the Wineries index card in the Corkboard, and drop it onto the Research folder over in the Binder. This demonstrates how easy it is to move material in and out of the drafting area, as well as the fact that you can use a view mode to help organise your material in the Binder. In larger projects, the Binder might become quite large, so remember this trick as it can make moving items from one remote section of the Binder to another very easy. As you learn more about the program, you'll find other ways to specify content as being “meta” to the draft manuscript itself, even while leaving it in the Draft.

Hopefully by now you can see the relationship between index cards, the binder, and the text within the items in the binder a bit better. Feel free to play around with things a bit more before proceeding to the next step.

**Working with Existing Material:** Likely, you already have content from an existing work in progress you would like to bring into Scrivener for your first project. While the methods described here could certainly help you with that, you might find the **Documents ► Split ►** sub-menu to be even more useful in converting a long document into an agile outline. For more information on this technique, explore [Splitting The Document \(section 15.3.3\)](#).

## 6.6 Compiling the Draft

We've only scratched the surface of what can be done, but before ending this tour we'll look at one more thing: compiling. Compile is Scrivener's terminology for what most

applications refer to as Exporting. This name was chosen to reflect the fact that Scrivener is doing quite a bit more than just converting the data from one format to another or moving data to an external location that other programs can access. Scrivener's compiler can generate documents to a number of different formats, do simple or complex transformations on the binder structure, reformat everything to a consistent look, and much more. The philosophy of letting you work the way you want to work is very much supported by the compiler. There is no need to work in Courier, Times New Roman, or whatever your editor or publisher prefers. You can choose an elegant font or one that is easy to read at 3am with sore eyes. In the end Scrivener will generate a document to your required specifications.

Let's try it now. Press **Ctrl-Shift-E** or select the menu command, **File ► Compile...** After all of the features described above, you might think you got the wrong function at first. By default Scrivener presents a very simple form with some industry standard presets and an export file format chooser.

- From the "Format As" menu, select "Novel Standard Manuscript Format"
- Change the "Compile For" option to Rich Text (.rtf)

That's it. Click the **Compile** button, choose a name and location for your RTF file, and once the dialogue box dismisses, open the file using the Explorer. It should open with your system's default RTF editor. As you can see, all of the test lines you typed in are there, with the header names prefixed with chapter numbering, and hash marks in between the sections as well, those came from the Compiler too, as a part of the manuscript preset. This can all be changed, or turned off entirely. Notably, the text you typed into the Wineries card will *not* be here. That's because we moved it outside of the Draft folder. The compile feature works exclusively<sup>2</sup> with the Draft folder.

Let's try one more thing. Close your RTF document, switch back to Scrivener and open up the compile window again. Leave everything the same, but this time click the down arrow button which appears to the right of the "Format As" drop-down menu. The full compile interface will appear, and with that, you'll be able to see just how much power is available. Feel free to explore the interface and do a few test compiles to see how things change. For example, if you don't like the hashmark separator between scenes, click on the Separators panel and change the first option "Text Separator" to something else. Many of the options are self-explanatory, but if you find yourself confused over something, read the relevant sections in the chapter on Compiling the Draft ([chapter 22](#)).

There is much more to learn, but if you are in the middle of a major project and want to get started right away, you should now know enough to make a simple outline and start

---

<sup>2</sup>Well, all right, not entirely exclusively, but that's a topic for another time!

writing content into that outline, and then compile your work as a single draft document. As you've seen from the amount of flexibility we've demonstrated, you can rest easy on the topic of "doing things the right way from the start". Unlike many programs, Scrivener makes it easy to change your mind later on. Folders can become files, items can be easily re-arranged, merged, or split apart, and so on; so don't worry too much about adhering to any proper ways of working. Scrivener has no standard except for *your* standard, which will develop in time as you become more familiar with the software.

**Importing Existing Work:** Chances are, you've already got some words down in another program somewhere. Scrivener supports many different text formats, and even a few data formats as well, like outliner formats ([section 11.1](#)). Importing material is often as simple as dragging and dropping the file into the project binder. After doing so, you may wish to split it up into more manageable pieces ([section 15.3.3](#)). A good rule of thumb is that RTF is the best. It is Scrivener's native file format, so if you can export your existing work to the RTF format before dropping it into the Binder, that will generally be the best solution.

If you have a spare hour or two, it would be really beneficial to go through the full interactive tutorial or watch the [introductory videos](#)<sup>3</sup> available on the Scrivener homepage. Together, they cover Scrivener's features in much more detail and should address most of what you should know to get the best results in your work.

Further recommended reading:

1. Interface ([chapter 5](#))
2. Navigating and Organising ([chapter 9](#))
3. Gathering Material ([chapter 11](#))
4. The Editor ([chapter 14](#)) and Writing and Editing ([chapter 15](#))
5. Compiling the Draft ([chapter 22](#))

For now, happy writing!

---

<sup>3</sup><http://www.literatureandlatte.com/videos/index.php>

# 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

One of the principal goals of Scrivener is to be your first stop when a new major project is embarked upon. The program therefore contains a variety of tools which address the early phases of a writing project. These include: brainstorming, outlining, collecting and organising research, gathering prose snippets, building background and world-building information, and more. 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 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?

This section will cover the majority of the tools that you may use to accomplish these goals. If your methods couple the processes of writing and preparation, you might find it beneficial to read portions of both this section and Writing (Part III). 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 finish off your text with. In Scrivener, there is no separation between outline and book order. This seamless approach will help you get straight into the writing phase, even while you are still planning and evolving the work. Thus, you can just easily start writing prose immediately, and let the bigger picture emerge organically out of what you write.

To better learn the fundamentals of Scrivener—such as where your works will be saved, and how to use the binder to select items within your project to view or edit them—it is recommended that you read the next three chapters. After that point, we'll discuss some more practical research, planning and project usage techniques, which can be learned as needed.

The topics that will be covered in this part are:

- Project Management (chapter 7): Creating, saving, backing up, and managing projects on the disk.
- Setting Up the Binder (chapter 8): Using the built-in sections of the binder, managing collections, organising your work, and setting up document templates.

- Navigating and Organising (chapter 9): Optimise how you navigate within a project.
- Setting Up Meta-Data (chapter 10): The various forms of meta-data available for tracking progress and types.
- Gathering Material (chapter 11): How to collect material from other resources and applications.
- Project Planning (chapter 12): How to rapidly capture your ideas and give them shape, shapes that will eventually become your manuscript.
- Cloud Integration and Sharing (chapter 13): How to take your work with you, collaborate with others, and manage cross-platform usage between Mac OS X and the Windows operating systems.

# Project Management

Scrivener is a project based application and stores your projects on your computer in the locations you designate when you create them. In this way it is much like any program that loads and saves files to your disk—rather than a database style program where all of your information is managed behind the scenes. These project files are saved into your Documents folder by default, though you can choose to organise these projects however you like, even after you’ve created them. Some find it convenient to rely upon the **File ► Recent Projects** menu item to organise their projects, or simply just 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 the tools can sometimes lose track of projects.

The intended use of a project is to store everything relating to a single major work, be it your next novel, screenplay for a film, a doctoral dissertation, or a serial collection of articles for a magazine. This 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, and there are ways to work around the default assumption where the Draft folder contains a single work. These alternatives will be explored in more detail much 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 deeply penalised for doing so. 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.

Going a bit deeper, the project file is a cohesive folder of files containing all of the pieces that make up your project using industry standard formats. The accessibility of this format is meant to be used as a last resort safe-guard, not as a way to allow you to edit a project in places where Scrivener is not available. What is important to understand is that the top-level folder and everything created beneath it is an integral part of your project. This top-level folder will use the name of the project you chose, with “.scriv” on the end of it. If you intend to copy this project from one computer to another, or synchronise it with a cloud service, be sure to copy the whole folder, and not just parts within it (the “project” file directly beneath this folder, for example, only contains the

backbone of the project, not the data.

## 7.1 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 very 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.

## 7.2 Creating a New Project

When you first start Scrivener, you will be presented with the Project Templates window (Figure 6.1), which also includes a handy “Getting Started” section. You may hide (or choose to reveal it again) this at any time using the **Options...** drop-down menu in the lower left of this window. In addition to providing an easy selection mechanism for new projects based on templates, you can also create a blank new document, open existing projects from your disk, or select from a short list of recently opened projects. To call up this window later on, use the **File ► New Project...** menu command, or press **Ctrl-G, Ctrl-N**.

Along the left side of the window are template categories. A number of useful templates have been provided, and it is easy to make your own as well. The **Options** drop-down menu provides template management tools. To read more about creating and managing templates, see Templates (section 7.7).

To create a new project based on an existing template, click on that template to make it active, or select the “Blank” template to start with a fresh project that can be constructed to your needs. A description of the template, if one has been provided, will appear in the grey text box below the selection area. Enter the name of your project in the “Save As”



field. This is what you will use to identify your project in the future, and it will also be used to name the folder (which will contain your entire project) saved onto your disk. It can be changed later, on the disk, if you please, so if your work in progress doesn't have a name yet, don't worry!.

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, after clicking on the Browse button.

The **Create** button will activate once you have supplied a name and valid save location. Click this button to proceed with the settings you have chosen.

## 7.3 Saving and Making Copies

While you work, your projects will be automatically saved on a frequent basis. By default, this means that after two seconds of inactivity, the project will always be saved. You can monitor this process by watching the upper-left corner of the window. The name of the project in the title bar will have an asterisk after it whenever the project contains edits which have yet to be saved to the disk. If you stop typing or clicking in the interface for a moment, the mark should disappear after a brief pause, and now everything is saved to the disk. You can change how rapidly this happens by adjusting the inactivity interval in the General section of the application options.

At times, or if you have selected a very long interval for some reason, you might wish to manually save the project. This can be done as per normal in any application, by selecting **File ► Save**, or pressing **Ctrl-S**. This feature need not be used to fully save the project, however. Very often, by the time you activate this command, the project will already be saved to the disk and it will do nothing. However if you work rapidly and rarely produces pauses, it might be a good habit to get into.

It is also possible to save a separate copy of the project, while working, and continue working in the new copy. To do this, select **File ► Save As...** (**Ctrl-Shift-S**), and choose a new name for the project. After supplying a new name and clicking the **Save** button, the active project will become the new one you created, and the old copy will be closed. This can be a useful tool when you wish to experiment with a series of radical structural changes. If you merely wish to create a backup, or a trail of major milestones, but continue working in the primary project, read about creating backups on the fly ([subsection 7.8.3](#)).

Additionally, since Scrivener projects are just files and folders on your system, you can use Windows Explorer to manage projects, create duplicates, and archive old versions. **Always be sure to close your projects before doing so**, and always treat the top-level folder with “.scriv” in the name as the whole project. Never try to split it up into several

pieces. Everything in this folder needs to be together in order to continue working as a project.

## 7.4 Opening Existing Projects

Existing projects can be opened in a variety of ways. For convenience, Scrivener keeps track of the last several projects you have opened, and stores them in a list, accessible from **File ► Recent Projects...** In addition, this list can be accessed from the Project Template window, via the **Open Recent** drop-down menu at the bottom.

By default, Scrivener will remember any projects you leave open when you quit, and will re-open these for you the next time. This behaviour can be changed in the General options tab.

You can open older projects using **File ► Open...** (**Ctrl-O**), or click the **Open an Existing Project...** button in the Project Templates window.

Projects can also be opened directly using the Explorer, or any shortcuts to the .scrivx file that you have saved in your quick launch bar or desktop. You can also use Explorer's search tool to find missing projects.

## 7.5 Moving and Deleting Projects

Despite 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 main folder, and so moving it will have no detrimental effect on its safety, and is a handy way to transfer all of your working data to another machine.

**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, if it is moved while open the software will no longer save files in the correct location, and can lead to lost work.

It is important to remember that the entire folder needs to stay together. Your project is not just the .scrivx file you use to load it, but all of the other folders and files around the .scrivx file. To safely move a project, always move the entire container folder ending in “.scriv”.

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.

## 7.6 Splitting Projects Up

Occasionally, one or more portions of your project might exceed the scope of your original planning and become what should be 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 novel. Another common example are from those who use a dedicated Scrivener project as a commonplace book, or idea gathering tool. They 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.

Fortunately, nothing could be simpler. 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.
2. Use **File ► Save As...** to create a fork copy of the entire project. Once you run this command, you will immediately begin working in the fork. 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. If you do not have a lot of meta-data and never work with linking items together, it's safe to do. However if you use a lot of labelling or status markings, the second method will be better. Most meta-data *will* transfer, even snapshots, when dragging and dropping. Anything that requires the project itself be set up a certain way however, like particular labels with predefined colours, will be more touchy and transfer, and Scrivener Links will always fail (references, so long as they point to items within the dragged selection, will work, so long as they are all moved at once).

At this point in your reading (if you are going through this in a linear fashion) some of the above might be arcane knowledge, so feel free to bookmark this page if you feel you'll need it in the future, so that once you know more about the things you can do in Scrivener, you can make a more informed decision.

## 7.7 Templates

Simply put, project templates are ordinary projects which have had some basic structure and settings added to them. In some other applications, templates are almost a way of modifying the way the entire application behaves and so in a way the presence (or lack)

of templates as *features* define the shape of what the software can do. With Scrivener, it's best to think of them as example starter projects, or even better yet, a learning tool. Many of them come with a few example items added to the binder, but these are meant to be helpful guides for laying out your book, not forms that you have to fill out or features that must be worked around. The items added to templates are like any other items you've added to projects on your own. They can be deleted, modified, duplicated, or set aside and ignored.

In fact, you can create your own templates for future use, which will be covered in more detail in the following pages. This is often a useful way of further customising Scrivener's default behaviour, as many settings found in menus are specific only to the projects you make them in. Since the built-in templates are just starter projects, you can use them to create your own derivative templates based upon them. Like many others, you may decide a better starting point for you is just the blank starter, or a slightly modified version of it with a few of your preferred project settings.

### 7.7.1 Using Built-in Templates

Most of the built-in templates will provide you with a readme-style document at the very top of the Binder that explains what the purpose of the template is, what has been modified and set up for you, and how best to use it to develop your project. Most also contain a sample PDF showing how the final manuscript will be formatted if all of the project settings are left alone.

For templates that do not come with a help file, here are a few things to keep in mind as you explore templates:

It is worth saying again: templates are just starter projects. With the exception of a few and somewhat technical differences, there is nothing in a template-generated project that is different from a blank project. In fact, a blank project is essentially an extremely minimal template. Everything that you see in a template can be handled just as you would handle any other items in the binder. They can be duplicated, deleted, modified, or what have you. Scrivener templates do not provide you with “forms”, “wizards”, or parts of the interface that you must fill out before proceeding. A sample chapter with an empty scene in it is just what it looks like: a folder with a file in it. To change it, edit it as you would any other piece of outline. If you want another one to make a second chapter, duplicate it. Scrap it if you don't like the style at all and want to design your own structure.

When you use a new template for the first time, check the project meta-data settings ([Project ► Meta-Data Settings...](#)) and make sure that everything is arranged the way you prefer.

Some templates will change the name of the Draft and Research folders to something more appropriate. The novel template, for example, changes the Draft name to Manuscript. Note that when the Draft has been renamed, all interface elements that refer to it will have their labels updated.

### 7.7.2 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 will 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 references: try making a list of writing resources, critique groups, and research portals that you often use.
- Character sheets or research note starters.
- Compile settings: since compile settings are 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!).
- Starter Collections: standard collections as well as saved search collections can be placed in a template as well, though initially they will probably be empty.
- Starter story structure: set up your favourite story-building 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<sup>1</sup>. 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. On average, it will take at least half a dozen new projects before things start to settle down. It is easy to update existing templates with revised material; this will be covered in greater detail in the following pages.

---

<sup>1</sup>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.

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

In the “Template Information” box, provide the following details:

**Title** The visible title of the template (what will appear beneath the thumbnail in the template browser).

**Category** Which section to add the template to in the browser.

**Description** A brief description of what the template provides. You may not find this necessary for yourself, but if you intend to share the template with others, it is a good idea to explain what the template is meant to provide.

The “Icon” box on the right gives you options for setting 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 as File...** button, which will generate an icon file for you based on the currently selected thumbnail. After you have edited this image in your favourite image editor and saved it as a graphic file, you can use the drop-down menu to select **Custom...** and choose the saved file from your computer.

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.

### Updating Templates

To update a template which has already been saved, follow the above instructions to create a new project using the template you wish to alter, make any changes you wish, and then choose **File ► Save As Template...** All of the information from the original template will be filled in for you, so you do not have to worry about replicating this information every single time. If you do not need to make any changes here, just click the **OK** button and then **Yes** to confirm that you wish to update the existing template.

Note that after you have created a project from a template, the two have no further connection. Thus, existing projects that you created from a template will not be modified to reflect changes made to the template. Likewise, if you have created a project purely to update a template, there is no need to keep the project around after that point, as allowing the project to auto-save will not automatically update the template; you must choose **Save As Template...** again.

Built-in templates cannot be overwritten. However, you can create projects from them for editing and then save your modifications as a new template by giving it a new name.

### 7.7.3 Managing Templates

In the Template browser window, the **Options** menu in the bottom left provides the following for managing templates:

**Set Selected Template as Default** Changes the default template selection to whichever template you currently have selected. A yellow border will appear around the selected template. Once this is set, the next time you call up this window it will highlight that template for you.

**Hide Getting Started** Disables the special starter category at the top of the category list, which contains useful links to the Interactive Tutorial and the User Manual. These can also be accessed from the Help menu from within any open project, so it is safe to remove the Getting Started category once you've familiarised yourself with the program. You can also choose to reveal it again in this same option drop-down.

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

## 7.8 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.

### 7.8.1 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:

`C:\Users\YourAccount\AppData\Local\Scrivener\Scrivener\Backups`

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, and Scrivener will rotate the files (delete old backups) to keep the maximum for each project under 5 backups. 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 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.

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, or use the **File ► Back Up ► Exclude From Automatic Backups** menu command to restrict the sensitive file from getting backed up with the rest.

Read about the various options available in the section on the Backup tab ([section B.9](#)) in Scrivener's preferences.

### 7.8.2 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.

While the project is open, use the **File ► Back Up ► Exclude From Automatic Backups** menu command. Once this option is checked off, the automatic backup system will ignore this project 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.

### 7.8.3 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 subsequent 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 Dropbox. By default, backups 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).

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**. This will follow any relevant options that have been set in preferences, such as how many to keep, where to store them, and how many to keep around. 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 this feature 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.

#### 7.8.4 Restoring from Backups

In the unfortunate event that a backup is needed to fix a problem with your current working project, restoring couldn't be easier. As noted previously, backups are by default just Zip archived projects, a compression format that is natively supported by all modern operating systems. They are, compression aside, no different from the project you work with on a daily basis, though they will be successively older revisions, and therein lies their power. The only other thing that makes them special is that they are tucked away in a safe out of the way place, to help avoid being inadvertently modified. If you are unsure of where your backups are located, you can always open the folder they are stored in by opening the Backup tab in **Tools ► Options...** and clicking the button at the bottom, **Open backup folder...**, to reveal them.

**What are Zip files and how do I use them?:** 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. It makes them easier to move around because it makes lots of little things seem like *one thing*. However a box is not the thing(s) you put into it. This might seem like common sense, but a lot of people wonder why they cannot open their .zipped project backups in Scrivener. The reason is simple: so long as it is in a .zip file, it is not a Scrivener project. You would not try to eat a candy bar that had been placed inside of a cardboard box. To eat the candy bar you need to remove it from the box (and probably from its own wrappings) first. The same goes for digital files that you store inside of a digital box. They must be removed from that box before they can be used.

Once there, you'll 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. It is best practice to select the backups you wish to check and copy them to a temporary location, first. The backup folder rotates content, so copying a set out will ensure *these* backups do not get deleted by Scrivener's cleaning routine while you experiment. To do so, simply click and drag the selected .zip files while holding down the Ctrl key, and drop them some place convenient, like your Desktop.

Locate the most recent backup<sup>2</sup> and unzip it by double-clicking on the '.zip' file and dragging the "project name.scriv" folder that you see to the Desktop. Zipped projects cannot be opened directly in Scrivener. The unzipped folder is a full project and can be opened in Scrivener in the usual way. Examine it, and if it will suffice, **close it**, then remove the broken version of the project and replace it with this extracted copy. You can now clean up the temporary working area if you wish and continue working as normal. If it contains the error as well, then proceed to the next older backup, continuing until you find a valid copy.

If you find you need to only restore *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. This can be useful 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 files you've worked with on the Desktop can be removed at your convenience.

## 7.9 Tips for Working Cross-Platform

The Scrivener project format<sup>3</sup> is fully compatible with its Macintosh companion. No conversion is necessary, and both versions can work off of the same source file (at different times; no project should ever be opened more than once). The primary difference in appearance between the two platforms is that Windows does not have a "package" or "bundle" format like the Mac does. 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.

---

<sup>2</sup>The sequence number cannot be used to determine this as Scrivener rotates through numbers. If you have not enabled the date stamp option, you will need to use your operating system to determine the modification date of the file.

<sup>3</sup>On a Mac, it is important to note that only Scrivener version 2.x 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.

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.

# Setting Up the Binder

This chapter will go over the various aspects of the binder, and how to best use Scrivener's features to structure, navigate around in, edit, and write your draft. Scrivener does some things which are unusual, and generally restricted to high-powered expensive applications. As a consequence, this can result in some interesting surprises, some of which are revealed in the built-in templates distributed with Scrivener.

## 8.1 How the Binder is Organised

The binder is the document browser on the left of the main window where you can organise your files (by default it is coloured a pale blue when the project window is active). It allows you to structure and rearrange your work with the utmost flexibility, much like a ring-binder in the real world (hence the name). To provide one example, you can drag files from Windows Explorer straight into the binder to import them, which you can read more about in Gathering Material ([chapter 11](#)), or from one project binder to another.

The binder is an optional element, though in most cases you'll want to rely on its organisational features at all times. To hide the binder, click the Binder icon in the toolbar, select **View ► Layout ► Hide Binder**, or press **Ctrl-Shift-B**. All of these commands can be used to restore its visibility if it has been hidden.

The binder is labelled at the top. Some functions will automatically replace the Binder view with a specialised listing. Project Search Results are the most common of these. This label will change to reflect whatever sidebar view is currently in use, and additionally the colour of the sidebar background will change to help notify you that you are no longer looking at the main project binder. You can always get back to the binder by clicking the small **X** button in the footer bar of the binder, or by clicking on its tab, if Collection tabs are set to visible.

### 8.1.1 The Three Root Folders

The binder has three default root folders which cannot be deleted or moved from the top level (although they can be renamed, moved around amongst each other, but not within

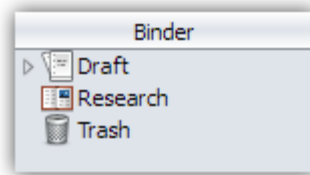


Figure 8.1: The three default top-level folders

each other or other folders, or assigned custom icons). To use Scrivener effectively, it is very important to understand the significance of these folders.

### 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. This user manual, for example, has each section of the manual in its own document. From this header “The Draft Folder” on to the next, “The Research Folder”, is the contents of a single document nested within the “How the Binder is Organised” section, which is turn nested within the “Setting Up the Binder” chapter group, and so on.

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 (excepting documents that have their “Include in Draft” setting unchecked). Thus, the Draft folder is central to Scrivener: ultimately, you are aiming to complete and arrange everything inside the Draft folder so that each of the elements it contains form an organic whole that can then be output as a single file to the format you require (which could be anything from a web page to a Word document to an e-book).

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. While media files cannot be placed directly into the Draft folder, images can be placed into the text files themselves, in their appropriate locations within the text, just like in a normal word processor.

A good way to think of the draft is like 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—what you find useful as a writing tool. The compile feature offers a way to tape the whole thing back together into a single file, which

is what you'll need to work with the rest of the world. Meanwhile the "Scrivenings" editing mode does the same thing, but in a temporary fashion, and as such is an authorial tool which makes working in smaller pieces easier, since you can at any time view and edit a larger segment of your draft as though it were a single file. These features will be discussed in depth in due course, but for now it is good to know that there are no penalties for slicing and dicing up your manuscript into an easy-to-visualise outline.

### The Research Folder

The Research folder is the default 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 you wish to organise your research, or you can rename it and create other folders in the root level to hold different types of research. You aren't limited to using this one location to store material. Anywhere outside of the Draft folder is a valid place to organise old drafts, notes and inspirational materials.

### The Trash Folder

Whenever you delete a file in Scrivener (by pressing **Shift-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 Trash folder. You can tell at a glance if the Trash folder has anything in it, as its icon will appear full. To permanently remove files from the project, select "Empty Trash" from the Project menu after having moved files to the Trash folder. Note that although you can rearrange items inside the Trash folder or view it like any other binder item (and of course, you can move files into the Trash folder by dragging); you cannot create new files inside it. It is not meant to be a working area. If you wish to have a secondary holding area for files you are not sure if you want to really delete, you can create your own folder in the binder to hold them.

## 8.1.2 Scrivener as an Outliner

If, like many authors, you are used to using a word processor for all of your writings, you probably have a few habits that you'll inevitably end up bringing with you to Scrivener. One of these is working with large sections of information. In a word processor, it can be clumsy (and even dangerous) to work in extremely long files the length of an entire book, but at the same time it can also be clumsy working with sections that are too short; it means having so many files open at once and no good way of keeping all the windows organised.

A common tendency is to treat the binder like a flat list of those semi-large files, say the length of a chapter, because even just that small incremental step can be so much better

than working in a word processor. But if you organise your project in this fashion, you could be missing out on a part of the Scrivener experience.

As it uses the convention of folders and files, you might instinctively approach the binder as a sort of “super-Explorer”, but in actuality, it’s far better to think of the binder in terms of being a bonafide outliner. You might have been drawn to Scrivener for this very reason, but because of the daunting number of things that must be learned all at once, have found yourself falling back to using a simplified file-is-chapter approach.

If you have never used or even heard of the type of software referred to as an “outliner”, some of this may not make immediate sense. Outliners are a fairly old software invention (nearly as old as the spreadsheet), that never really caught on as dedicated software. However the *concept* of outlining has become widespread in many programs, even including the way our operating systems show the contents of our disks. 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 something is indented “beneath” an item, then it is considered a child of it, and that item is a parent to it. Items at the same level of indentation are referred to as “siblings”. Another crucial feature of the outlining principle is that you can hide the 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 immediately 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* of the work within the outline. You can imagine it as being as though each line on the paper 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. It is like having your very own table of contents that instead of reflecting upon your choices, allows you to use it like a tool and make those choices with a single and easy drag and drop.

If all of this still does not make sense, I recommend going over the Quick Tour ([chapter 6](#)) provided in this manual. It approaches the concept of outlining and how the text you write can be organised into a topical tree. You might think this is all more complexity than you need, but in my experience, once the lightbulb switches on to this idea, it becomes second nature in the same way we now think of individual files on our computer as holding discrete chunks of information. You will likely find your work becomes more fluid and visible to you as you can now survey not only the major peaks of your work (chapters), but smaller components within those chapters, too.

The next section will demonstrate how Scrivener approaches this concept, and how it can allow your book to emerge as a genuine outline. That is what the binder can do for you, and unlike other word processor applications or plug-ins that tout outlining

capabilities, you can use your own internal “authorial outline” which may or may not have anything to do with the visible structure (e.g. chapter names, space breaks between scenes, subsection numbering) of the book itself.

### 8.1.3 Folders are Files are Folders

As you no doubt instinctively know, 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 corkboard view?

With the folder still selected, click the currently active Group Mode button (Corkboard is the default) in the main toolbar (or use **View ► Corkboard**). You can tell which mode is active by its shaded background. After clicking the shaded button, you should get a blank view that looks suspiciously like an empty text document. In fact, that’s precisely what it is. Type in some text and see what happens. Note that in the binder, the Folder icon has changed, and now sports what looks like a little page of paper in its corner. This means the folder itself has text associated with it.

Let’s try something a little more radical. Right-click on the Folder in the binder and select “Convert to File”. It should change to a stack of papers with words on them. 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”. Nothing fundamental changed here. You didn’t lose any information, the only thing that changed is the underlying type of the item, which is really primarily a visual change. When an item has other items beneath it, but it is a file, it is often referred to as a document stack, or a file group, but fundamentally it can act just like a folder does. This documentation will often refer to either folders or file groups as “containers”, when the distinction does not matter.

Let’s test that theory. Select one of your files in the binder. You should see its text appear in the editor as you always do. But let’s do the same thing we did earlier with the Folder, click on the corkboard icon. It will be empty, but you are now looking at the corkboard for that file. With the focus in the Corkboard, press **Ctrl-N** to make a new document. 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.



Perhaps you can better see why it is a good idea to think of the binder as an outliner, instead of a glorified Explorer. What is all of this flexibility good for? You don't have to be thinking in terms of constrictive structural elements as you flesh out the skeleton for your book. You can just add items as you go, building out new corkboards, and changing things to folders or back to files once a structure begins to emerge.

Scrivener lets you work the way you find most comfortable. There is no need to build out 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, the book can be built onto an outline or even a mix of the two. It's all up to you and how you work best.

In Scrivener, the concept of a Folder 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.

## 8.2 Binder Icons

There are many binder icons (Table 8.1) used to show the type of item, and in the case of folder and text documents, hints about its status as well—the text and folder document icons will change depending upon certain conditions. 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 “words” in it. Likewise, folders will display if they have synopsis or text content (see the prior section if that comment resembled gibberish).










How you use this information is up to you. Some people like to develop their outline in stages, first building the skeleton, then fleshing out the longer descriptions in the synopsis fields, and finally starting to write prose into the pieces of the outline. These indicators can help them see how far along into the process they have gone, and what areas of the draft need more work.

## 8.3 Binder Controls

This section contains some tips for manipulating the outline in the binder.

**Revealing and Hiding Elements** Click on the arrow (on Windows XP or configura-

Table 8.1: 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.
	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 ( <a href="#">section 15.6</a> ) saved for it.

tions using the classic Windows theme, this will be a “+” or “-” icon instead) to the left of any container to reveal or hide its contents in the binder.

Use the **LeftArrow** and **RightArrow** keys to open and close containers.

In conjunction with the Alt key with either of the above methods, expand or collapse an entire portion of the outline tree at once.

**Reveal and Hide All** “Expand All” and “Collapse All” in the **View ► Outline ►** submenu will disclose or reveal the entire outline at once.

**Collapse All to Current Level** Working in a similar fashion to Collapse All, this command will collapse the entire outline, but only up to the currently selected item’s level. So for example if you 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.

**Moving items with drag and drop** Items can be moved around in the binder with drag and drop. Simply pick up the item by clicking and holding the mouse button down,

then drag the mouse pointer to where you wish to drop it. The binder will display a blue target underneath the pointer, showing where the drop will end up.

**Moving items with the keyboard** **Ctrl-ArrowKeys** 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 **Ctrl-DownArrow** 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.

**Keyboard navigation** The arrow keys by themselves will move the selection around in the binder.

- Up and down arrows will move up and down the outline.
- Left and right will ascend and descend the hierarchy. When a container is selected, the first time you press LeftArrow, a container will be closed, the second time will jump up to the next parent.
- Combining the Shift key with the UpArrow and DownArrow keys will select multiple items in the chosen direction.

**Copying Items** While ordinary copy and paste functions are not allowed in the movement or reproduction of binder items, copying a selection of items can be useful. In most cases this will place a list of names on the clipboard.

You can effectively copy any number of items by duplicating them with the **Documents ► Duplicate ►** sub-menu. They will be copied in place, and you can then move them to their intended location.

## 8.4 Using Collections

Collections are a way to augment the binder method of organisation, and they will temporarily supplant the binder view in the sidebar, allowing you to reorganise or arrange collections of items in a flat list, regardless of their position in the actual project or hierarchical relationships with one another.

You can think of collected items as being a bit like an alias, or a shortcut to that item. Changes made in the editor to items within the collection will also be made to the original item in the binder, it is better to think of them as being the same thing. A single item can be in one collection or in many. Each instance will point back to the same original item in the binder.

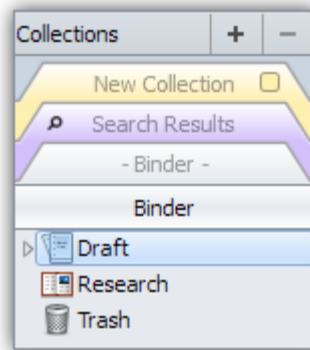


Figure 8.2: The basic Collection interface with the binder selected.

When setting up a new project, you might like to create a few starting Collections to help organise material as it flows into the project, or as you generate new outline ideas. If you find yourself creating a sequence of common Collections at the start of every project, you might want to design a custom template (subsection 7.7.2) for yourself. Here are some example uses for Collections:

- Experiment with an alternate scene flow without disrupting the original layout.
- Collect all scenes which still need to be edited.
- Designate items which 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, clicking on this tab would highlight all of the problem words, listing only those sections of the binder that contain them.
- Create a special compile group with an alternate selection and order than a typical compile of the draft.
- Store saved searches for future use, or to monitor workflows.
- Set aside groups of documents in a day-by-day schedule, removing them from each daily collection as you address them.

#### 8.4.1 The Collection Interface

To reveal the Collection interface, click the Collections icon in the toolbar, or select **View ► Collections ► Show Collections**.

Each tab represents a single collection. In the screenshot (Figure 8.2), the “Binder” tab is selected. While not a true collection, this is how you can navigate back to the Binder after viewing a Collection. Click on any tab to select it; when a tab is selected, it will re-colour and ghost all of the tabs below it to help indicate which is the active tab. Additionally the header bar will change its title to the name of the tab. 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.

When first created, Collections will be assigned with an automatically generated colour, but their tabs can be recoloured by double-clicking the colour chip to the right of the label, and selecting one of the provided options. If you would like to use your own colours, you can select the **More...** option at the bottom of the menu. Likewise, double-click on the title to change the name of a collection. The Binder and Search Results tabs cannot be renamed or permanently removed.

There are three main types (excluding the Binder) of Collections:

1. Standard Collections: allows you to freely add and remove items as you work. This is the most flexible form of Collection.
2. Search Results Collection: a special collection type that cannot be removed. Any project searches will have their query and results stored in this collection. This is persistent between project sessions.
3. Saved Search Collections: indicated with the same magnifying glass used for Search Results, on the left side of the label; they will be dynamically populated with a search query every time you view the tab.

#### 8.4.2 Standard Collections

Creating a new standard Collection is as easy as pressing the **+** button. Any items that you have selected in the active view (including the Binder, other views, or even other Collections), will be automatically added to it.

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. There are several ways to add items to a collection:

- With the tab interface revealed, use drag and drop from any view including the binder.

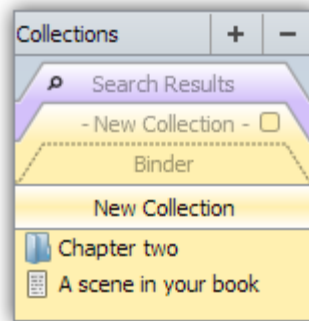


Figure 8.3: Standard Collection Controls

- Using the **Documents ► Add to Collection ►** sub-menu, any selection of items, including the active item that is being edited or viewed in the editor, can be added to the bottom of a collection list.
- In the binder sidebar, you can right-click on an 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”.

Remove items by selecting them, and then simply pressing **Shift-Delete** on your keyboard.

Delete the entire Collection by first selecting the tab you wish to remove, and then clicking the **-** button in the upper title bar.

Items can be re-ordered within the list using click and drag, or the same **Ctrl-ArrowKey** combinations used in the Binder. Since there is no hierarchy in a Collection, you will not be able to promote or demote items.

To view all of the items in a collection as a group view, simply select all of them in the sidebar using **Ctrl-A**. This will create a multiple selection, so you will be unable to move items around within it, but otherwise it creates an effective view of the collection in the order it appears in the sidebar, using any of the three group view modes.

### 8.4.3 Search Result Collection

If you’ve used Project Search, then you’ve been using a Collection without perhaps realizing it. Search results are placed into this special 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 Project Search that you ran will always be available to you, even after you have rebooted

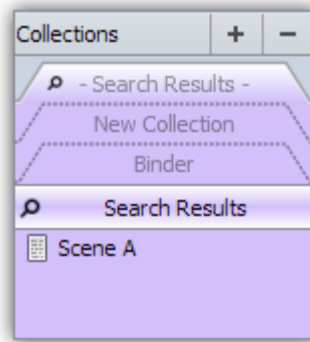


Figure 8.4: The special Search Result collection

your computer. Like the Binder tab, the Search Result tab cannot be deleted, and there is always one included with every new project, even when no search is currently in use.

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 depend upon the search criteria, you cannot manually add, remove, or change the order of items from the Collection, as you can with the standard type.

#### 8.4.4 Saved Search Result Collections

As with the Search Results tab, saved searches have a magnifying glass in the corner to indicate that it is a special type of Collection which automatically generates its contents for you. Saved searches are created by clicking the magnifying glass in the Project Search tool, and selecting **Save Search...**

If the Collection interface is hidden, the sidebar background will change to a random, automatically generated colour, and the header bar title will adjust to the name you provided. To leave the saved search, you can either click the close button in the sidebar footer, or enable the Collection tab interface and select another tab.

The contents of a saved search are dynamic. Every time the tab is clicked on, the saved criteria will be checked against the current state of the project. If you wish to “freeze” a list 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, **View ► Collections ► Convert to Standard Collection**.

2. Copy the contents into a new collection: This is quite easy to do. Simply select all of the items (**Ctrl-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.

Saved Search Collections can be removed in the same fashion as Standard Collections, by selecting the tab and clicking the **-** key, as shown here ([Figure 8.3](#))

See Also:

- Project Search ([subsection 20.1.2](#)).

#### 8.4.5 Back to the Binder

Often you might wish to know the overall disposition of a collection in the binder, or where a specific select few items in the list are located in the overall structure of the project. The **View ► Reveal in Binder** menu command 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.

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. 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 return you to the Binder, with the selection retained. Another method is to select all of the items that you wish to collect together and use the **Documents ► Group** command to assemble them into a new folder in the Binder. You will need to switch back to the Binder to see the grouping.

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.

It can also be useful to experiment and implement an experimental text flow. If a chapter or section just doesn't seem to flow right, you can quickly create a new collection, Option-drag the container of items to the new tab, and then re-organise the flow using the collection's ability to view itself as a corkboard or outliner, then reviewing the text with scrivencings mode. Once you are satisfied with the new layout, select all of the sub-files and drag them back into the original folder in the binder using the above method. The items will be re-organised for you in the actual book structure.



# Navigating and Organising

In this chapter, we will discuss the project window itself, and the many tools that are available for moving around within this window, controlling its appearance, and adjusting how it behaves when you click on things within it.

## 9.1 General Navigation

Moving around within a large project is an important part of writing efficiently, especially in an application like Scrivener, which lets you cut your work up into small pieces. An average book might have anywhere between several dozen and a few hundred sections arranged into many containers. Along with the binder, there are two primary tools for making sense of all of these pieces: The Outliner (section 12.2) and The Corkboard (section 12.1), covered later in this portion of the book. For now we will concentrate on how to use them as navigational tools, and as your skills increase, you'll find other uses for them beyond mere navigation.

Beyond the three primary navigation tools, there are a number of other features which can be used as needed to enhance the main three, and these will be covered in this section as well.

### 9.1.1 Navigation Specifics

Both corkboard and outliner views have an Auto-Load feature which when enabled, will load any selected item(s) in the other split, but only if the editor has already been split.



Figure 9.1: The Auto-Load button will appear blue when activated.

The Auto-Load button (Figure 9.1) is located in the footer bar, and will appear blue when switched on. This action will use the current default view mode for that split.

Additionally, for corkboard and outliner views, you may Double-click on any icon (to the left of the title itself) to load that document in the editor using the current view mode.

## 9.2 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, are working in full screen mode, or just would rather not drill down to a particular spot solely to select an item, the Go To menu is an easy 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.

There are three locations where this menu appears:

- The main application menu, **View ► Go To ►**. It additionally provides four shortcuts for navigating within an editing session:
  - Previous Document (**Alt-Shift-UpArrow**): Selects the previous item in the binder, no matter what the relative depth.
  - Next Document (**Alt-Shift-DownArrow**): Selects the next item in the binder, no matter what the relative depth.
  - Enclosing Group (**Alt-Shift-LeftArrow**): Selects the parent container which the currently selected item is within; in effect traversing up the hierarchy, much as if you had clicked on the parent container using your mouse in the binder. The current view mode preference will be used if possible. If you are editing a section of the Draft using Scrivenings mode, then using this command will expand the Scrivenings session to include the parent and its siblings, just as if you had clicked on the parent; if working on a corkboard, the next corkboard higher up will be viewed. In cases where the current view mode conflicts with the selection, corkboard will be used as a fall-back.

An example might be if you have scrivenings mode enabled as your default view mode, but you are currently viewing an image or PDF. Since scrivenings does not apply to these items, this command will switch you over to corkboard.

- Editor Selection (**Alt-Shift-RightArrow**): Isolates the current selection (or active document in a Scrivenings session) in the editor in single text mode, no matter what type of item it is. Useful for viewing folder text, or eliminating everything else in a Scrivenings session so you can focus on one piece.
- Via clicking on the item icon in the editor header bar.
- In full screen mode, the Go To menu is accessible in the control strip along the bottom of the screen, as a button.

In all cases, both sub-folders and items can be selected. When a sub-folder is selected, it will be loaded with the default view mode (Corkboard by default), and when using the button in full screen mode, it will load the text of that item. The header bar menu is provided as a convenience, and more readily applies the navigation action to a particular split, whereas one must first ensure the right split is active before using the application View menu.

### 9.2.1 Navigating Within Scrivenings

When the active split contains a Scrivenings session ([section 15.8](#)), the “Go To” menu will be transformed into a table of contents for that session, letting you quickly jump to sections exclusively within the session. The main application Go To menu will adhere likewise to this model, but if a split that is not in a Scrivenings session is activated, the menu will return to presenting the entire binder. The header bar menu will always show the contents of the current session for that split.

See also, [Editing Multiple Documents \(section 15.8\)](#).

## 9.3 Controlling Binder Integration

With a 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. When this is the case, the split which has its header bar highlighted in blue will be the one that receives binder clicks. The inverse of this is however not true. The binder will not automatically select items based upon what you are viewing in the editor. The editors can drift from the original selection quite easily. All you need to do is use the history buttons, double-click on an index card icon in a Corkboard, use the **View ► Go To ►** menu, click on a Scrivener Link in the text, and so forth. There are two features available which change how clicking in the binder sidebar work.

### 9.3.1 Locking the Editor

Any editor split can be locked with **View ► Editor ► Lock in Place**, or **Ctrl-Shift-L**. The header bar for the editor will be changed to a dusty red colour which persists even when the split is not active. Either or both splits can be locked independently. This feature can also be accessed from the header bar icon menu (subsection 14.1.2).

Locking the editor prohibits *external clicks* from impacting the locked editor. This means that clicks in the binder, corkboard or outliner (when Auto-load is enabled), will be blocked. This makes it easy to keep your editing session while you use the binder or these other views to explore the outline. What locking will **not** do is prohibit actions taken from within the editor itself. This includes use of the binder navigation arrows, history buttons, manually dragging items to the header bar, or menu navigation commands. Locking is meant primarily to keep the interface from taking actions that it would ordinarily take automatically. Locking will not inhibit intentional actions that *you* make. These actions (as with the **View ► Go To ►** sub-menu) will navigate as requested and cancel the lock state.

A common exception to this rule of thumb is when it makes sense to scroll your view, or select items within it, based on an external selection. If the editor is a locked Scrivenings session (where more than one text file is viewed at once), then if you click on a portion of the outline that is contained within that session, Scrivener will scroll to the right spot for you. In a large locked corkboard, you can jump to individual cards by clicking on them in the binder. Again, this alternate method of navigation only works when the item you click on in the binder sidebar is already contained within the locked view. You will need to unlock the editor in order to use the binder to select items outside of the current view.

### 9.3.2 Binder Affects

**Binder Affects...** The **View ► Binder Affects ►** sub-menu contains several choices for altering the standard behaviour to become locked rather than dynamic. For example, if you have a left and right split, ordinarily when you click in the binder what you click on will be loaded in the active split. If Binder Affects Left Editor has been enabled, however, *all* clicks will be sent to the left split no matter which is active. The precise contents of this menu will alter, depending upon whether you are using a vertical or horizontal split orientation.

Alternately, the “Other Editor” choice will cause the binder click to always act in opposition to its normal behaviour, loading the binder click in whatever editor is *not* currently active.

To reset the binder behaviour to default, use **View ► Binder Affects ► Current Editor**.

See also:

- Splitting the Editor ([section 14.4](#)) for further documentation on how to use editor splits.

## 9.4 Linking Documents Together

Much like hyperlinks on the Web, Scrivener Links make it easy to create and use a network of text-based cross-references within your project. Since in most cases they do not export from your project, it is safe to use them in the text of your work wherever you need them. They can also be used within the various available notepads. Their ability to elevate the visibility of information in your project can turn them into a valuable navigation tool, or as providing convenient uses beyond the project itself, during Compile, such as cross-referencing that will be useful for the reader.

The creation of links can even be a good brainstorming tool, as you can build your outline with links, while typing. Links appear much like URL links do on the Web. They will sport a blue appearance and be underscored. The link colour can be customised in the Appearance option tab, under “Customizable Colors”, in the Editor section. The underscore cannot be removed, so even links set to be the same colour as the base text can be identifiable.

To use links, simply click on them with the mouse pointer, as you would in a web browser. Scrivener Links (but not ordinary hyperlinks to the Web) will open a split editor (if necessary) and load the linked item in the split. This will use the ordinary view mode preference, and so is identical to clicking on that item in the binder. If you link to a folder, it will open in Corkboard view, for example.

### Select Text and Link

Often you will want to link a phrase to a specific document without having the name of that document be a part of the link. To create links in this fashion, select the text you wish to link, and then right-click on the selected text. Use the **Scrivener Link** sub-menu to create a new link. This menu is also located in **Edit ► Scrivener Link**.

This sub-menu functions in a very similar manner to the Go To menu ([section 9.2](#)), by providing an organised list of every item in the binder.

### 9.4.1 Removing Links

Links can be removed (including URLs to external web pages) with the **Edit ► Unlink** menu command, without destroying the underlying text. Any link falling within the currently selected text will be removed, so there is no need to be precise about what you select.

This is an easy way to wipe out dozens of links at once, something which is of particular use when pasting text from web files, or other Scrivener documents with links you no longer require.

## 9.5 Layouts Panel

The Layouts panel allows you to save different layout set-ups and load them again easily. For instance, in the early stages 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. By using the Layouts panel to save different layouts, you can switch between these view settings quickly, without having to go through the process of hiding the binder, toggling the toolbar and so forth. You can even specify that outliner column settings and other details will be restored from a layout. To show the Layouts panel, select **View ► Layout ► Manage Layouts**.

If you do not have any layouts saved yet, you'll be presented with a window and an empty list. You can try the feature out immediately by clicking the **+** button. A screenshot of your current project window will be taken and placed into the right side of the screen, and you'll be given the opportunity to name the layout something meaningful. With a combination of names and screenshots, your layouts should remain accessible and easy to find.

Switching to a saved layout is easy, just select the layout and click the **Use** button in the footer bar of the left list. Another way of switching layouts is to use the main menu instead of the Layout panel. When you save a layout, it will be added to the **View ► Layout ► All Layouts ►** sub-menu.

To remove a layout you no longer need, select the layout and click the **-** button in the footer bar. 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 with the current project window layout, simply select that layout in the list, and click on the gear menu 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. Using the gear menu, select the layout you wish to export, and then choose **Export Selected Layout**. The layout will be saved as an XML file in the location you provide. Use the **Import Layout** menu choice to import a layout you have transferred from another computer or downloaded from the Internet.

Because layouts are separate from projects, there are some things that they cannot save, such as which collection you are viewing. Even within one individual project, these details might change or no longer be relevant. Here is a list of things layouts *do* save:

- Binder visible/hidden
- Collection tab interface visible/hidden
- Inspector visible/hidden
- Split type (horizontal/vertical/none)
- Binder mode (whether it affects one or current editors)
- Header bar visible/hidden
- Footer bar visible/hidden
- Editors locked in place
- Ruler and Format Bar visible/hidden
- Selection auto-load in alternate editor settings
- Window size and position and view sizes
- Corkboard/outliner/scrivenings/editor mode for each split

In addition, optional settings are saved depending upon whether these two checkboxes are enabled for the specific layout.

**Save outliner and corkboard settings** All corkboard display settings, such as card size, ratio, card wrapping, and so on will be produced when using the layout.

In the outliner, which columns are visible or hidden, and column sorting.

**Preserve all meta-data 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 produced when using the layout.

# Setting Up Meta-Data

Documents of any type in Scrivener can have various meta-data associated with them. Meta-data is a way of talking about something without modifying 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 meta-data, as is the paper-clip. In Scrivener, meta-data can be applied in similarly analogue fashion, like stamps across the face of an index card, or what colour of pin or marking it has in its corner. Other types of meta-data in Scrivener are more abstract. Some of this meta-data can be viewed in the corkboard and outliner views, but the main interface for viewing and editing all of the meta-data for a given document is the Inspector ([chapter 18](#)). This section will brief you on the various types of meta-data available, and how to create more types if necessary. It will not go into all of the details of the Inspector pane; please read that section to gain a full understanding of how meta-data is assigned to documents.

## 10.1 Meta-Data Types

The basic meta-data types include the title and a text or image synopsis. The title is the most important component of any item in the binder, as it will be used to identify the document in the many views, menus, and export methods. When exporting as files, filenames will be set according to the title field; when compiling, titles can be used to generate headers for that section; index cards use the title; and the outliner is composed of a list of titles.

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 a variety of ways.

1. Corkboard: used to display the content area of the card.
2. Outliner: will be placed beneath the title (which appears in bold), by default.
3. Binder: revealed in the tooltip when hovering the mouse over the corresponding title.



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 and binder as described above, and in the various export and print options that include a synopsis field. The image is not a *replacement* for the synopsis, but an override for the corkboard.

In all cases, specific types of meta-data can be displayed as columns in the Outliner view.

### 10.1.1 Setting Up Label & Status

The next two forms of meta-data are flexible in what you can refer to them as. Using the **Project ► Meta-Data Settings...** (**Ctrl-Shift-M**) you can set up custom names for both the label and status, and the interface will adjust accordingly, no longer referring to them as “Label” and “Status” for that project. If you wanted, you could have “POV” and “Location” instead of “Label” and “Status”, or “Focus” and “Type”, or “Monkeys” and “Bananas”, or anything else you want. The documentation will continue to refer to them as labels and status for convenience. 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 best administered in the inspector, and are given priority placement in most of the printing methods. Labels, having an associated colour, can also be used to visibly tint various interface elements, such as binder or outliner rows or even the entire index card “paper”.

The default blank project and many of the templates come packaged with a few example status & labels, but you will most likely wish to create your own. To add or remove project labels, use the **Project ► Meta-Data Settings** menu item. The interface provides tabbed tables for two meta-data types.

Label and status lines can be added or removed with the **+** and **-** buttons. Edit an existing entry by double clicking on the text. For labels, the colour can be changed by double-clicking on the colour chip to the left. The order in which these items appear in their respective menus is determined by their arrangement in this table. Click and drag rows to rearrange them.

When you create new items in the binder, either by adding items or importing, they will automatically be supplied by a default label and status value. Ordinarily this will be assigned to the “No Label” or “No Status” choices, which is a special entry that cannot be deleted, but they can be renamed. If you wish to change this behaviour, select the alternate value by clicking on its row, and then click the **Make Default** button.

Once you click the **OK** button, the interface will update with your changes it necessary, and the new options will be available from respective meta-data drop-down menus throughout the interface.

### 10.1.2 Label Colours

Labels can have a colour associated with them and are completely customisable—you can even rename “Label” to “POV” or something that better fits your project (see below). Labels are one of the most flexible meta-data 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 (richer more vibrant colours will stand out amongst pastel choices) depending on their associated material’s content. Label colour can be expressed in the following ways, nearly all of which are optional, and can be toggled in the **View ► Use Label Color In ►** sub-menu.

1. **Corkboard:** will be used to shade the upper right corner of the card when using the “Corner Mark” indicator type. When using the “Pin” type, a graphical thumb tack will be coloured according to the label. Use **View ► Corkboard ► Show Label Indicator** to toggle their visibility in the corkboard. The visual style can be changed in the Corkboard option tab.
2. **Outliner:** can be set to tint the background colour for each row using the label colour with **View ► Use Label Color In ► Outliner Rows**.
3. **Binder:** can be set to highlight the background of each binder item using the label colour in a manner similar to the Finder. Toggle this behaviour with **View ► Use Label Color In ► Binder**.
4. **Icons:** enabled with **View ► Use Label Color In ► Icons**, this option will tint document icons wherever they appear.
5. **Index Cards:** enable with **View ► Use Label Color In ► Index Cards** this optional method will tint the “paper” colour of index cards wherever they appear.

### 10.1.3 Status Stamps

As with the label, 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.

**Corkboard:** can be displayed as a diagonal “stamp” across the face of the card. The appearance of this can be adjusted in the Corkboard and Appearance option tabs, and display can be toggled with **View ► Corkboard ► Show (Hide) Stamps** or **F9**.

### 10.1.4 Using Keywords

Each document can have a list of keywords associated with it. 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.

Like labels, keywords have an associated colour which will be displayed beside the keyword in the inspector list, and in the Project Keywords window (see below). Additionally, keywords can be used in the corkboard as small coloured squares along the right-hand edge of the card. The visibility of these can be toggled with **View ► Corkboard ► Show (Hide) Keyword Colors**, or **Ctrl-F12**. The display preferences for these can be adjusted via the corkboard display controls ([subsection 12.1.4](#)).

When exporting or printing, keywords will be displayed on a single line, separated by commas.

### 10.1.5 General Meta-Data

In addition to the primary types of meta-data, there are a few smaller items, located in the General meta-data section of the Inspector:

**Modified** The date and time the document was last changed and saved.

**Created** The date and time the document was first created. You'll need to click on the "Modified" label, beside the double-arrows, to switch to see this data.

**Include in Compile** A checkbox determining whether the document should be included in the draft when exported or printed via Compile from the File menu. If checked, the document is exported or printed; if not, it is excluded from the export/print (this can be temporarily overridden in Compile). This way, you can have documents in the Draft folder that act as notes that are never included in the final draft. Note that 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 compiled. The checkbox never has any meaning for media entries, as these items can never be compiled.

**Page Break Before** If this checkbox is ticked, a page break will be inserted before the document when compiled. Useful, for instance, if the document marks the beginning of a chapter. As with Include in Compile, this only has any meaning for text documents inside the Draft folder.

**Compile As-Is** All options set in the compile formatting options pane ([section 22.8](#)) will be ignored for the checked document. This means no extra material will be added, nor will text be removed, including titles, and no reformatting of the text will be

done. As with the above two items, this checkbox has no meaning for items outside of the Draft.

### 10.1.6 Exporting Meta-data

All meta-data 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. Meta-data can be exported in the following fashions:

- **Compiling:** when compiling the Draft, enabling meta-data export is an optional feature of the Elements pane. The various types of meta-data will be exported into the Draft in a variety of ways best suited to the type of data involved.
- **File export:** when exporting files from the Binder, meta-data 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 meta-data export in formats which would not otherwise allow text data to be inserted into it, such as mp3 files.
- **Drag and Drop:** When dragging documents from one Scrivener project to another, all meta-data will be preserved where applicable. In some cases, like labels and status, these values will need to already exist in the target project.

## 10.2 Project Keywords

The Project Keywords window holds all of the keywords in use by the project. As you enter new keywords into the Inspector ([subsection 18.3.3](#)), they will be added automatically to the central project list. You can access this pane via **Project ► Show Project Keywords**, or **Ctrl-Shift-OK**.

You can create a new keyword by clicking on one of the buttons at the bottom of the panel. The left-most button creates a new keyword as a sibling of the selected keyword, and the one next to it creates a new keyword as a child of the selection. The third button deletes the selected keyword(s). Keywords are automatically sorted alphabetically. You can drag any keyword onto any other keyword to make the dragged keyword a child of the one onto which it was dropped. For the most part, the hierarchical structure of keywords are really only an organisational convenience with no effect on how the keywords are represented in documents themselves.

To assign keywords in the panel to a document in the main window, select and drag the desired keywords from the panel and drop them into the keywords table in the inspector of the main window or onto the document in the binder, corkboard or outliner. If

there is a multiple selection in the binder, corkboard or outliner and the keywords are dropped onto one of the selected documents, the keywords will be applied to all selected documents.

You don't have to use the keyword panel to create keywords, of course—you can also enter keywords directly into the keywords table, in the inspector ([subsection 18.3.3](#)), to apply them to the current document. Note that whenever you apply keywords via the inspector, they will automatically be added to the list in the panel if they are not already contained there, so that the panel should always have a complete list of the keywords you are using in your project no matter how you have created them.

Keywords can be managed from the panel. By changing the name of a keyword in the project keywords window, all of the associated keywords in the project can be updated in the background. You will be given a choice of doing this, or creating a new keyword from the variant (which won't be assigned to anything by default). When removing keywords from the panel, they will also be removed from all of the documents they had been assigned to. To change the colour that represents the keyword, double-click on the colour chip to the right of the keyword.

The project keywords window also provides an easy way to search for keyword usage in your project. Simply click on the keyword you wish to look for, and click the **Search** button in the bottom of the panel.

You can also search for combinations (using the Boolean AND logic) of keywords by selecting more than one keyword and then clicking **Search**. All documents containing *all* selected keywords will be returned. When searching for only one keyword, the “Exact Phrase” method will be used.

See Also:

- Project Search ([subsection 20.1.2](#))
- Keywords Pane ([subsection 18.3.3](#))

# Gathering Material

## 11.1 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 lying around that you want to bring into an existing project, the 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 better) into the project itself. The original copies on your disk will not be removed or altered in any way. While it is possible to easily edit many file types from within Scrivener with an external editor, if you wish to retain maximum external control over the files, you might wish to use References ([subsection 18.3.2](#)) instead, which are useful when your research material measures in gigabytes.

There are two ways to import documents from other programs:

1. Drag and Drop. In Windows Explorer, select the files you wish to import and then just drag them straight into the binder (not the corkboard or outliner) in Scrivener. 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** sub-menu provides some handy methods for bringing existing material into your project binder, including directly off of the Internet, if you have the URL.

The first time you import anything, you will be presented with an informative window describing important information pertaining to the various limitations of the import feature. If you do not wish to view this warning every time you import files, be sure to check “Do not show this warning again”.

When importing text documents, they are internally converted to the RTF format so that Scrivener can work with them easily (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 data for some file types, or documents with complex embedded data, like Excel spreadsheets.

When using the import menu items, material will be imported according to where the current binder selection is set. If you have a line selected in the binder, the material will be inserted between the selected line and the one below it. If more than one item is selected, the last item (in top-down fashion) will be used as the insertion point. This means that some forms of import will be disabled depending on where your selection is. If your selection is in the Draft, you will be unable to import media of any kind; if your selection is in the Trash, all importing will be disabled. The only exception to this is importing another Scrivener project. These ignore binder selection and create top-level items and/or folders at the bottom of your binder.

When using drag and drop, the dropped material will be placed wherever the insertion indicator in the binder shows. If you try to drag media into the Draft, the drop will be prohibited until you move the mouse out of that folder.

### 11.1.1 Supported File Formats

Scrivener supports the following text types for importing:

- RTF (rich text format): the universal rich text standard. This is often the best format to use for importing from word processors.
- DOC & DOCX (Microsoft Word format): the main formats used by Microsoft Word and the writing industry.
- TXT (plain text): plain-text files use encodings (ways of mapping the text in the file to characters on your screen). If a plain text document gets imported as gibberish you may need to try using another encoding. If all else fails, use copy and paste from a program that opens it fine.
- FDX (Final Draft 8+ format): using the standard document format for Final Draft, you can import scripts directly into any area of the Binder and have them converted to Scrivener's scripting format.
- OPML (Outline Processor Markup Language): commonly exported from outlining style applications, this format will let you transfer an outline tree from one application to another. Many applications will also save text notes attached to outline nodes, and if Scrivener detects them, it will use this material in the main text editing area.

- MM (Mindmap): the primary format for Freemind and related software. Import .mm files directly into Scrivener to translate their hierarchy into outline format.

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 audio/visual formats (WMV, AVI, MPG, WAV, MP3 etc), PDF files and HTML file types.

In addition to the supported types, it is possible to import any type of file at all. While Scrivener will not display unsupported file types in the editor, you can still archive these files into your project for portability or convenience, and use the **Documents ► Open ► in External Editor** menu command to load the file in your preferred editor for that file type. Changes made to the file will be saved into the project automatically.

If you would prefer that Scrivener filter out all unsupported file types when importing, enable “Import supported file types only” in the Import/Export options tab ([section B.8](#)).

### 11.1.2 Web Page

This lets you enter the URL of a web page that you would like to import into Scrivener. The web page will be archived on import, meaning you will no longer need to be connected to the Internet to view it. Web pages cannot be imported into the Draft (which only supports text), so you must ensure that a non-Draft item in the binder is selected for this item to be available, or change your options to import web pages as rich text files. When such conversion is performed, Scrivener will store the page’s original URL in References ([subsection 18.3.2](#)).

**Functional Web Pages:** 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, GoogleDocs, or even a simple search form. These sorts of pages, if they require a login, will 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.

You may also drag and drop URLs from many browsers location bar, into the Binder itself, just as though you were importing a file from your hard drive.

### 11.1.3 OPML and Mindmap File

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 mindmapping application. Some of these applications support attached notes to each outliner header. These can be imported



into the main text area, or the meta-data field of your choice. The import dialogue provides these options:

**Import notes into** Disable this checkbox if you only wish to import the outline structure. When enabled, and if notes exist, Scrivener can be instructed to apply these notes either to the Synopsis alone, the main text, or the document notes for that heading in the outline. Note that in the case of the last two options, the synopsis will still be set automatically to the first few hundred characters of the note.

#### 11.1.4 Scrivener Project

Allows you to import another Scrivener project into the binder of the current project. The files of the other project will be imported into a folder entitled “Imported Project (original project name)” at the bottom of the binder, with the full binder structure intact. Keywords, project references and any project notes are imported as text document(s). Project notes will be imported as files into a “Project Notes” folder, placed at the top of the main import folder. 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 **File ► Import ► Scrivener Project...** to have Scrivener do its best to retrieve all the data. All the text, synopses and notes will be retrieved, along with snapshots. The outline structure may not be recovered from badly corrupted projects, but every attempt to import the raw data will be made.

#### 11.1.5 Import and Split

Supported text file (plain or rich text) is selected you will be presented with a text field. Type in the separator that was used in the text to define sections. A common example might be a “#” character. Any line in the document that contains only the text entered into this box will be removed from the file and used to split the results in two. This process continues, further splitting the work into subsequent binder items, until all of these lines have been processed.

A portion of the first line in each sub-divided section will be used to title the incoming items in the binder.

## 11.2 Linking to Files on Your Computer

It is possible to link to files, rather than import them directly into the project. In order to reduce confusion in the binder, linked items cannot be directly placed into the binder. They must be inserted into the references table for another item. A common tactic is to use the typical filing card system, whereby you create an index card describing the

material, and then link to it and any other relevant files in its references. To read more about references, see References (subsection 18.3.2).

## 11.3 Scratch Pad Panel

The scratch pad is a universal Scrivener tool that is not tied to any particular project. You can use it to jot down notes that don't fit anywhere else, or that arise whilst you are looking at another application. You can access it from the **Tools ▶ Scratch Pad** menu (**Ctrl-Alt-Shift-P**).

The scratchpad is a floating panel, which means that you can have it open over all windows in Scrivener, and even while you work on other documents or even other applications. It is possible to optionally turn the window transparent, to better facilitate its floating nature. Use the “gear” button menu to enable transparency, if you find the window getting in your way.

The top pane contains a list of all the note documents you currently have stored in the scratch pad (by default there will only be one untitled note). New notes can be added by clicking the “new file” button in the lower-left, and notes you no longer need can be deleted with the delete button next to it, or by selecting it in the list and pressing **Shift-Del**.

When projects are open, the **Send file to Scrivener** sub-menu will provide you with a list of all opened projects. There are two ways 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 sub-menu. 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. *Copy Document To*: A new document will be created beneath the selected document. This sub-menu 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.

Since the scratch pad does not store your notes in any projects, it uses ordinary RTF files, which by default will be stored in your Documents folder under “Scrivener Scratch Pad”. These are regular files that you can edit with other programs. You can even create your own rich text files in this directory, making it possible to add notes while Scrivener is closed.

To take a screenshot and save it into the current notepad, click the “hand” icon. A preview of the current screen will be shown—this is what you will get if you click the

**Save and Close** button. The **Grab an Area with Mouse** button will print a grid over the screen and you can then draw a rectangle over the area of the screen you wish to capture. **Print Screen** takes a screenshot of the whole screen, but will wait a predetermined number of seconds (set below) before doing so, giving you a chance to set up the screen.

To change where notes are stored, use the General preferences tab (subsection B.1.4). Pointing your scratch pad folder at a synchronised or shared location (like a DropBox or Google Drive folder) means you can share scratch pads between Scrivener installations if you use multiple computers.

## 11.4 Text Appending Tools

Text selections can be easily appended to other texts within Scrivener, using two handy methods.

**Append Selection to Document** This command is available in the contextual menu, when right-clicking on selected text. The command will provide a sub-menu populated with every item in the binder, prioritising any favourites to the top of the list. 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...** item at the top of it which will let you create a new item in the Binder with the selected text as its contents.

# Project Planning

## 12.1 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 order in the outline structure, or alternatively as a freeform corkboard where cards can be freely moved about without directly impacting the structure of the book. 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.

### 12.1.1 So What are Index Cards, Anyway?

One of the principles that takes 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.

So let's take a look at what an index card really is. There are really seven different components to an index card, but depending on your preferences you might not see them all at once.

1. The document icon
2. The title of the document

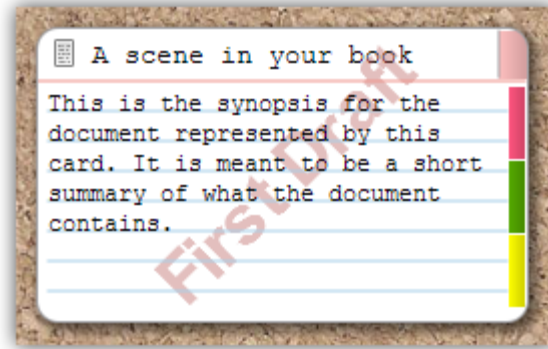


Figure 12.1: A fully loaded index card using the rounded card theme.

3. The synopsis of the document
4. The colour label (colour swatch in top-right corner; optional)
5. The status stamp (diagonal text stamped across the middle; optional)
6. Keywords indicators (swatches along the right; optional)

Also, not shown in the screenshot (Figure 12.1), it is possible to tint the colour of a card based on its label. This can be used in addition to, or instead of, the colour chip in the upper-right corner. To tint cards by label type, use [View ► Use Label Color In ► Index Cards](#) (or the **F7** key).

At its most minimal, you'll only see the icon, title, and synopsis, the three core elements that cannot be removed, all other items must be enabled in the [View ► Corkboard ►](#) sub-menu. All of the options you select for corkboard appearance and label tinting will be saved into the project. If you prefer a certain look by default, consider creating your own custom starter template.

The icon, as you might have guessed, correlates directly with the icon that is visible in the binder. In most cases, this will be one of the variations of a folder or text icon, depending on its content status, or if you are viewing a corkboard outside of the Draft, it might be some kind of media file icon.

**The title** is simply what you would expect: it is an 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. The title is also often used by the compiler if it has been set up to consider it, but we'll come back to that in a later section.

**The synopsis** is 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 or using Scrivener's automatic generation tools to form a synopsis from the first few lines of text. 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.

To edit either the title or the synopsis on a card, double-click in the text field you wish to edit. Once editing, you can use **Tab** and **Shift-Tab** to navigate between cards and their two editable fields, much like you would in a spreadsheet. To terminate editing, press the **Enter** key at any time.

### 12.1.2 Index Cards as Images

Instead of text, you can choose to display a graphic for the synopsis instead. This choice will be reflected in the inspector, as well as on the corkboard. To set an item's synopsis to image mode, click the index card icon in the header area of the synopsis card in the inspector. A blank placeholder will be shown until you drag and drop an image into this spot.

Something to consider is that even if a card has been set to image mode, any text that was typed into it before will be preserved. You may not be able to see it directly, but it will be retained for tooltips and when compiling when the "Synopsis" checkbox is active in the Formatting compile option pane ([subsection 22.8.1](#)). Likewise, if you switch the card back to text mode, the image will be preserved in the background in case you ever switch back.

To remove an assigned image, click the **X** icon in the far right top of the index card in the inspector.

### 12.1.3 Linear Corkboard

The standard, or linear, corkboard displays a linked representation of one level and section in your binder hierarchy. 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 hierarchy you can view different layers.

In 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. If you enable "drop-ons" in the Corkboard options tab ([section B.3](#)),

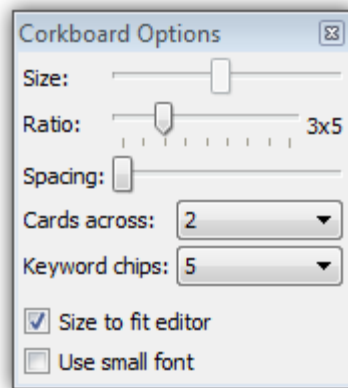


Figure 12.2: Corkboard options pane

you'll also be able to stack cards with other cards, just as you would drop items onto a folder. Many other appearance and behaviour settings can be adjusted in that tab.

#### 12.1.4 Corkboard Options

The Corkboard Options pop-up contains settings which are specific to each project. Changes made to this pane will only impact the active project they have been made in, letting you have two entirely different visual appearances for each major work. Any changes made will be saved with the project, meaning you can save your preferred defaults into a custom project template for future use.

To access the pop-up, click on the “Corkboard Options” footer bar button, as shown (Figure 12.2). To dismiss the pop-up, click the close button in the title bar of it.

Since the interaction between some of these settings are interdependent, here is a useful guide for adjusting the settings based on what you want:

- 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 **Size** slider as desired.
- Cards that have a fixed size that wrap like words depending on the editor width: disable **Size to fit editor**, set **Cards across** to Auto, and adjust the **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**. This is how the screenshot (Figure 12.2) has been configured. It will always print two cards per row, sizing them to be as large or small as necessary, depending upon the width of the corkboard in the editor.

**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.

**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** 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** 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” (below) 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.

**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.

**Size to fit editor** This option is not available when “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.

**Use small font** The small font settings can be configured in the Appearance options tab (subsection B.2.3). Note that since you have full control over this and the standard index card fonts, “small fonts” might not actually be any smaller than the regular font size.



**Further Options:** There are a number of other appearance related preferences that you can set, which can dramatically alter the look and feel of the corkboard. Most of these options are included in the Corkboard preference tab (section B.3), but a few colour related options are also in the Appearance tab (section B.2).

### 12.1.5 Placing 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. They will be supplied with special icons, indicating the type of file they are. Graphics will be represented with a small iconic image of a sunset photograph.

When viewing a corkboard that contains graphics, they will be drawn on the corkboard as a photograph, rather than the typical index card.

You may also change how any item at all in the corkboard is represented in a way which will always override the default. Even folders and standard text files can be represented by a photograph, or *vice versa*, photographs can be represented as index cards with a text synopsis. For further instructions on how to do this, read the Synopsis Card (section 18.1) overview in the chapter on the inspector.

This will only change the item's representation on a corkboard. The text synopsis will not be deleted, and can be edited using the outliner. The binder icon will not change. If your graphic files have text icons, this means they have been dragged into a text file, rather than imported directly.

## 12.2 The Outliner

Outliner mode shows all of the descendants of the current document along with select associated meta-data in a tabular format like a spreadsheet. The default configuration will show the title and synopsis in the main columns on the left. You can edit by double-clicking into the text field you wish to edit. Once editing, you can move between titles with the Tab and Shift-Tab keys, much like in an ordinary text editor or outliner. Press **Enter** to exit editing mode.

You can 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**. Additionally, you can hold down the Alt 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, “Page Break Before” will present a checkbox that you can click to add a page break before that document when exporting. You can impact many checkboxes at once by holding down the Alt key and clicking on a checkbox.
- Labels and Status and Target Type will provide drop-down menus that you can use to adjust the meta-data for a row. For labels and status, If you wish to impact more than one row at once, 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 Title or Target column will not provide a control, but you can simply double-click on the field to change their value.

### 12.2.1 Managing Columns

To add or remove columns from the display, click the small down arrow button in the far-right column header bar, or use the **View ► Outliner Columns ►** sub-menu. 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.

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.

Primary information: These display the Binder name and synopsis card contents in columnar format.

- Title
- Synopsis

Corresponding to the general meta-data: the first two columns allow you to edit the Label and Status of each row individually. The remaining two columns are read-only.

- Label
- Status
- 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.

- Word Count
- Character Count
- Total Word Count
- Total Character Count

Further options from general meta-data: the checkboxes displayed in these columns are directly linked to their respective options in the Inspector, general meta-data section.

- Include in Compile
- Page Break Before
- Compile As-Is

Targets and progress tracking: as with statistics, these show target tracking information for each item in the outliner. The Progress column shows a 0–100% progress percentage bar, in like fashion with the progress metre in the Footer Bar ([section 14.2](#)).

- Target
- Target Type
- Progress

Any custom meta-data fields you have created will be listed at the bottom of the menu as well.

## 12.3 Project Notes

As your project begins to take shape, you will no doubt have many ideas that you wish to capture, but nowhere concrete to put them yet. Documents all have a separate note pane which can be accessed in the Inspector ([subsection 18.3.1](#)), but if you don't even have a stable outline built yet, there are no documents to put notes within. There are a number of ways you could approach this problem, such as keeping notes as documents in the Research folder or somewhere else outside of the Draft, or even in the Draft with their respective "Include in Compile" flags disabled.

You might also find the Project Notes tool to be useful, as it is accessible from everywhere in the interface. Project notes, like document notes, are a rich text field. That means you can format within them however you please, even drag pictures into them. Project notes can have different styling than document notes, making it easy to tell which you are currently viewing or editing. These can be set up in the Appearance options tab ([section B.2](#)).

Their primary advantage is that they can be viewed from any Inspector instance, no matter what you are viewing, and even make an appearance when what you are viewing would ordinarily have no Inspector data, like the Draft or Research folders.

**Project Notes and Exporting:** Project notes are not set up to be easily exported. There is no way to compile them at all, without physically copying and pasting their contents into a document. Document notes can be included in various compile options, but project notes cannot; nor can they be exported via any menu commands. So use project notes only for things which are strictly internal to the project. If you intend to export or compile notes at some point, it might be easier to take the original suggestion of using documents in the binder.

For a simple example, open the inspector if necessary (click the blue 'i' button in the toolbar, or press **Ctrl-Shift-I**) and try clicking on anything in the binder that isn't a document. A good example of that is the Draft folder. The inspector will be replaced by a single pane titled "Project Notes" (if that didn't happen, you might be viewing one of the index cards in the corkboard that pops up. Just click once in the background of the corkboard to de-select any cards.) Type in a sample line here so we can track it around the interface as we explore this feature.

This field is not only accessible from the Draft, but any document can view the project notes in their Inspector as well. Try clicking on one of those index cards in the Draft corkboard and make sure the left-most button in the bottom of the inspector is clicked (the one that looks like a little notepad). By default, this will show you the document notes for that selected card. You can click on the header bar where it says "Document

Notes” and select “Project Notes” instead. The sample line that you typed in above will become visible and the background colour will change. You can see how accessible project notes are. Even when viewing nothing at all they are visible, and they can always be selected when viewing anything in the binder, even a movie.

See also: The Inspector ([subsection 5.1.5](#)).

# Cloud Integration and Sharing

It is becoming increasingly important to have the ability to make your work instantly available everywhere you go, no matter what device you might currently be using. Sharing files amongst a circle of colleagues, proofers, and editors is also becoming common.

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 backup frequently.

## 13.1 Scrivener Everywhere

Keeping more than one computer synchronised can be a lot of work, but keeping one Dropbox folder (or similar service) synchronised between computers requires little thought at all.

It therefore follows that placing your Scrivener project into a Dropbox folder so that it is immediately available to every computer you own, would seem a perfectly natural progression. What you should be cautious of, when entertaining this idea, is that a Scrivener project is comprised of many smaller files, and that they all need to be working together as a cohesive whole. Ordinarily, this is not a problem. Nothing goes into your project and messes with it, except for Scrivener. When you place a project on Dropbox, however, you are giving Dropbox permission to scrutinise and curate the individual pieces of your project that ordinarily only Scrivener messes with. This goes for most synchronisation technologies in general, but since we've tested Dropbox the most, this section will be framed with that service in mind.

By and large, this is safe to do. Dropbox is designed to carefully make sure each computer has a thorough and exact replicate of what all your other computers have. However, a problem can arise when one computer edits your project in a way which conflicts with the copies on the other computers. This can be easy to accidentally do. All it takes is working on a copy of a project that is not yet fully synced. This might happen because your computer has been offline for a while, or because Dropbox never got a chance to fully update all of the files in the project.

When this happens, copies of each “conflicted” file will be duplicated with different names so that you can decide which one is the best copy, or manually merge them together in a text editor. Ordinarily, this is fine and easy to do, but inside of a Scrivener project, it can be more complicated. For one thing, the full contents of your project might not even be easily visible. Secondly, some of the files in a project are very technical in nature, and if you are not fluent in XML (not to mention Scrivener’s schema), it might be impossible to best decide which version is better. The solution is to never get into this position in the first place, and this is fortunately easy to do.

To avoid conflicts with synced projects:

1. **Always make sure your syncing software is done syncing before you open a project.** Good syncing software, like Dropbox, 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 task bar system tray, in the lower-right portion of your screen near the clock. Keep an eye on this indicator and wait for the green checkmark<sup>1</sup> 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.
4. **Reduce auto-save frequency:** this step is optional unless you are working on a very slow connection, such as dial-up. Editing a Scrivener project can generate a lot of network traffic because of how often Scrivener saves files to the disk. With modern high-speed connections, this should pose no problem, but with slower or unreliable connections to the Internet, it’s a good idea to set Auto-Save (in General preferences) a little higher so that changes are sent in larger batches to the Dropbox server, rather than trickling in constantly. When working this way, you might want to save manually more often, with the **File ► Save** menu command.

---

<sup>1</sup>It may be possible to customise the status icon, and not all of the visual themes use a green checkbox to indicate when Dropbox is in standby mode. Familiarise yourself with how your preferred status icon works.

**Alternately, pause Dropbox:** the newer versions of Dropbox allow you to pause syncing. This can be done from the status icon as all other functions with Dropbox can be accessed. Pausing syncing means that all changes made to the Dropbox folder will not be immediately synced to the server. Thus there is no need to mess with changing the auto-save frequency, and it keeps your entire working session in one “version” within the Dropbox system. Of course, you will need to remember to resume syncing prior to shutting down the system.

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.

While this section has primarily focussed on Dropbox as a means for keeping your project files synchronised between computers. There are many technologies for doing so these days. If you have another solution you are using and prefer, test its viability with copies of your project, or test projects, prior to committing important work to it.

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. There are fail-safes programmed into Scrivener to let you know when another session is currently active in a project, but making sure everyone is on the same page with who is editing the project will ensure no mistakes are made. Guideline #2 above deserves to be reiterated: **never open a project for editing if another person is currently editing the project.**



**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. In particular, screenwriting and bibliographies can be safely skipped unless your works require these functions.

The exception are the chapters discussing The Editor, and Writing and Editing. These will be of benefit to all, as this will introduce the text editor, where you will be doing most of your writing, and if you haven't already done so, at least skim the prior part on preparation. Vital concepts have been introduced there, and many of the following chapters will build off of those concepts.

The topics that will be covered in this part are:

- The Editor ([chapter 14](#)): Full documentation of the various editor views and how to use the primary editing and viewing interface.
- Writing and Editing ([chapter 15](#)): Full documentation of the various editor views, formatting, layout, and tools to enhance the editing and revision processes.
- Full Screen ([chapter 16](#)): Scrivener's full screen feature. Wipe aside all distractions and just *write!*
- Annotations and Footnotes ([chapter 17](#)): Learn how to comment as you write using either Scrivener's exclusive inline annotation tool, or linked comments. Also covers footnotes/endnotes; marker highlighting; and revision pens.
- Inspector ([chapter 18](#)): All about the inspector, which displays document meta-data, notation, references, and more.

- Scriptwriting ([chapter 19](#)): While Scrivener is not meant to replace an industrial strength scriptwriting tool, it can still function as a very adept first-draft scriptwriting solution, and produce exports which will work seamlessly in industry standard applications, like Final Draft.
- Writing Tools ([chapter 20](#)): General tools for writing and using the software more effectively, include:
  - Searching and Replacing ([section 20.1](#)): Comprehensive tools for finding items, words, notations, and even formatting are discussed here.
  - Goals and Statistics Tracking Tools ([section 20.2](#)): Scrivener has numerous tools for analysing the content of your project, as well as tracking goals and progression while you work.
  - The Name Generator ([section 20.3](#)): Create fictional names with ease, using the name generator.
  - Using Equations with MathType ([section 20.5](#)): If you need to insert formulae into your work, Scrivener integrates with the popular MathType software. [\* Using MultiMarkdown ([chapter 21](#)): If you prefer to write in a structured or semantic style rather than with a rich text focus, MultiMarkdown provides a simple, easy to read way of defining and using styles which can be exported to word processors, web pages, and LaTeX.]

# The Editor

## 14.1 Header Bar

The header bar appears at the top of each editor pane, and can be toggled on and off via **View ► Layout ► Show (Hide) Header View**. It contains useful navigation tools, displays the current title of the edited document(s), and provides control over editor splits.

The header bar has three colour states:

1. Grey: when no split is active, this is the default state. If splits are turned on, the inactive editor will be grey.
2. Blue: when splits are active, this denotes the *active* split. Any actions taken which will impact the current editor will be made to the blue document. The title itself will also be underlined for the active split.
3. 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.

### 14.1.1 History

On the left-most side of the header bar are history navigation arrows, which should be familiar in appearance and usage from any web browser. Use these to navigate backward and forward through the navigation history.

### 14.1.2 Item Icon Menu

Next to the navigation buttons is the document icon, which shows what type of document is being displayed (which can sometimes be abstract, if more than one item is selected for display in the view). Clicking on this icon brings up a menu containing some useful shortcuts to oft used commands.

**Reveal in Binder (Ctrl-Shift-8)** Will show the location of the currently edited file in the binder, opening any containers as necessary to do so. 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 a such result), or if you are currently viewing a collection and wish to find where the file is actually located in your project outline.

When used with Scrivenings mode, the section you are currently working on will be highlighted in the Binder.

**Path** Reveals the location of the current document in descending order. The top entry will always be the current document; the entry below that its immediate parent; and so on until the top of the project binder is reached. These can be selected to immediately jump to any ancestor of the current item.

**Go To** Offers an alternate way to jump to a specific binder item without using the binder. If the binder is hidden or you would rather not scroll and open folders, for instance, it will let you navigate to other areas of the project without having to alter your layout.

When viewing a Scrivenings session, the contents of this menu will operate within the confines of the current session only, thus providing a miniature “table of contents” for the current session.

**Take Snapshot** This command will take a snapshot ([section 15.6](#)) of the current text and store it for later use. This command always impacts only one file. When used in a Scrivenings session, only the constituent portion of the session your cursor is currently in will be saved to a snapshot.

**Lock In Place (Ctrl-Shift-L)** Locks the editor (or split) so that no Binder clicks will affect it. When an editor is locked, its header bar will turn a shade of red ([subsection 9.3.1](#)).

### 14.1.3 Header Bar Title

To the right of the history navigation buttons, the title of the currently viewed item will be displayed. As you navigate through the project, this title 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. Whenever the presented title consists of only the *title* itself, you may edit the name of it by clicking within the text, and pressing **Enter** to confirm. MAC:When a single item is shown, hovering over the title will display the “path” of the item in a tooltip. This is similar to a file path on your system. It will show all of the folders the item came from. You can also view this information graphically in the header bar icon menu, as already described.

When you are viewing a collection of items in a corkboard or outliner, the name of the viewed container will be displayed. That is, 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. Use the Inspector to view the details of selected items within a group view.

When viewing the result of multiple item selections in the binder (when more than one item is selected via Shift-click or Ctrl-click), the view will display the title, “Multiple Selection”. This is a unique view which lets you know that the corkboard or outliner you are viewing doesn’t “exist” anywhere in the project. You are viewing a selection of items, but you will not be able to add new items or re-arrange existing items because their relationship amongst each other does not correlate with anything in the Binder.

In all cases where the title is not a literal binder title, you will lose the ability to edit the name of the document. You will need to use the inspector or the binder to make this change.

#### 14.1.4 Sidebar Navigation

Moving to the right-hand side of the header bar, you will find two arrows pointing up and down. These arrows allow you to navigate within the project sidebar (be it the binder, Search Results, or Collection that is currently viewed), as though it were a flat list. This form of movement is unique in that the binder selection will update. In most cases, when you navigate to documents using the menus or the various controls available in the header bar, the binder selection will not update to follow your select, by design. In this case however, these two buttons are actually synonymous with clicking on documents in the sidebar.

Clicking the up-arrow 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.

These two buttons can also be activated with the keyboard shortcuts, **Alt-Shift-Left** and **Alt-Shift-Right**, respectively.

#### 14.1.5 Split View Button

Finally, the last set of buttons on the right-hand side controls editor splits. You can split the current editor, close a split, or change the split orientation all from here.

If no splits are open, you will see two buttons with a small line drawn through the icon in the horizontal or vertical orientation to let you know which way the editor will be split if you click on the corresponding button.

Once a split is already open, one of the icons will appear blank. Clicking on it in this state will close the *opposing* split. This means you should click on the blank icon in the header bar for the document view you wish to retain once the split has closed. Either side can be dismissed by clicking on the opposing header bar button. Additionally, if you click on the other button, the split orientations will switch without otherwise disturbing your layout. Both splits will remain open, only the orientation will change.

See also: Splitting the Editor ([section 14.4](#)).

## 14.2 Footer Bar

Below each editor pane is the footer bar. As with the header bar, its visibility can be toggled with **View ► Layout ► Show (Hide) Footer View**. This is the most dynamic part 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 drop-down 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. This section will cover the tools available to the standard text editor.

- For information on the various other document footer bars, see Viewing Media in the Editor ([section 14.3](#))
- For details on how to use the footer bar in script mode, see Scriptwriting ([chapter 19](#)).
- Read the chapters on The Corkboard ([section 12.1](#)) and The Outliner ([section 12.2](#)) for details on how to use the footer bar in those view modes.

### 14.2.1 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.

### 14.2.2 Quick Text Statistics

In the middle of the footer, the current word and character account 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. To view statistics for a selection portion of text, right-click on that text. You will find word and character counts displayed at the bottom of the contextual menu in grey text.

### 14.2.3 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. This button brings up the target options for the current document which will let you set the numeric word or character goal you intend to achieve with the section.

When enabled, the footer bar will acquire a progress bar next to the target icon which will fill as you type, gradually changing colour until you reach your quota. These targets are set individually for each document. You may also monitor and set them using the Outliner, by revealing the “Target”, “Target Type”, and/or “Progress” columns

To disable the goal metre for a document, enter a value of “0” into the target configuration for the document.

## 14.3 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. The footer bar will often change to provide tools for working with different media types. Note that Scrivener is, by and large, not intended to operate as an editor for all types of files. It would take years to design these features, and in most cases your existing applications will do a better job of editing PDFs, movies, and images. Consequently, a convenience button will be provided on the far-right side of the footer bar for most media views. You may also right-click on any media file in the binder and opt to **Open ► in External Editor**, which will open the file using your system default for that type of file, making it simple to load the document in an external editor of your choice. Changes made to the file will be saved within the Scrivener project when you return<sup>1</sup>.

### 14.3.1 Viewing Images

The image view is displayed whenever an image document is selected, allowing you to zoom in and out and rotate it. If the image is larger than the viewable area, click and hold the mouse on the image and move the mouse to drag-scroll the image around. To zoom in and out and rotate, double-click anywhere within the image or the grey area around it. This will bring up the image tools pop-up.

The slider at the top of the pop-up allows you to scale the image. The buttons to the left and right of the slider will zoom in and out by increments.

There are three additional tools available: the two circular arrow buttons will rotate the image 90 degrees clockwise or counter-clockwise, and the third button will reset the

---

<sup>1</sup>If you are already viewing that file in the editor, you may need to refresh its view in order to see the changes.



zoom of the image to fit your editor width. When this button has been clicked, Scrivener will attempt to keep the image at the editor's width, even if it changes, until you use the slider or its buttons.

### 14.3.2 Viewing PDFs

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. The total page count along with a page selector lets you skip from page to page with the arrow buttons, or jump directly to a page by typing it in. The zoom setting for the PDF is next to the page control, and comes with a similar "fit to width" function that images sport. When fit to width is in use, a zoom level will be selected for you, keeping the PDF from requiring horizontal scrolling to see the entire page. As with images, this will adjust as the editor width itself adjusts (such as when opening and closing the Inspector).

### 14.3.3 Viewing Multimedia Documents

The multimedia view is used to display movie and sound files. From the control bar beneath the viewing area (in the footer bar), you can play or pause the movie, change the volume, or step backwards and forwards through it. On the lower right, you'll see a rewind symbol with a pause button in the middle of it. This is an optional mode which will show up as blue if engaged, which will rewind your media by a set number of seconds whenever you pause. This interval can be configured in the Navigation preferences tab ([section B.4](#)).

To further aid in transcribing, you can use the **Ctrl-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. With the focus is in the media editor, you may skip forward and backward by a number of seconds with **Ctrl-}** and **Ctrl-{**, respectively.

### 14.3.4 Viewing Web Pages

The web view displays imported web pages. Note that clicking on any links in a page displayed in the web view will open the link in the system's default internet browser. Scrivener does not allow internal browsing simply because this would mean integrating many of the features of a browser into the web view (such as back and forward buttons, a refresh button, maybe even a download manager...) and this is outside of Scrivener's scope.

### 14.3.5 Viewing Unsupported Document Types

By default, Scrivener will let you import files into the Binder that it cannot display in the editor. You can restrict this by checking off “Import supported file types only”, in the Import/Export options tab ([section B.8](#)).

As with supported files, unsupported files are imported into the project package itself.

## 14.4 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 to new users. Rather than arbitrarily splitting the interface any number of times and ways, the editor uses a two-way split system so that you can easily control the flow of information in the opposing split while you work.

**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 ([section 15.3.3](#)) 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.

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.

We’ll now take a look at all the available possibilities in turn.

### 14.4.1 Horizontal and Vertical Splits

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 ► Layout ►](#) sub-menu, or by clicking the opposing button in the Header Bar ([section 14.1](#)).

When the editor has been split, a header bar and footer bar will be created for each split. This means that each side has its own history queue, view modes, zoom settings, outliner columns and so on. 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.

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. MAC: To reset the splits to be of equal width or height, double-click the splitter bar with your mouse.

### 14.4.2 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 ► Layout ► Swap Editors](#) menu command.

# Writing and Editing

## 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 22](#)), but suffice to say that you can work in one font, say the default Courier New, 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 certain aspects of your editing can be conveyed through the compiler, like a range of italicised text, while the base font itself can be changed beneath that. Scrivener’s editor can thus be used *like* a typical word processor, but with the knowledge that rich text formatting can be altered later on to produce different results.

Scrivener is not intended to be a full-blown word processor, but rather a word generating environment. It is not a layout tool, but a tool for cutting the text that will become your book. Many authors who have stricter formatting requirements, such as scriptwriters, will start their projects in Scrivener, and end in a word processor or desktop publishing, or some other specialised application. Someone writing a novel or short story could very well do the whole thing in Scrivener, including the final print for submission and export to Word format for sending an electronic copy.

You might be wondering if Scrivener offers a plain-text editing environment as well. 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, and annotating. Even if you do not require formatting, you will probably find the formatting tools *useful* in the writing process, especially if you are compiling to plain-text, where they will all be safely stripped out and invisible to the reader.

**Emulating plain-text:** If you find working in a plain-text environment to be more productive, you may wish to set the default editor font to one that does not support rich text features, such as Courier, and change the ruler settings so that there are no

indents or paragraph spacing. This set-up will emulate a plain-text environment whilst still allowing features such as annotations and highlighting.

When using MultiMarkdown to format your work, you will find that some formatting, such as inline images, footnotes, and so forth will be converted to MMD markings during compile. Most formatting, however, will be stripped and thus can be useful in an annotation setting.

## 15.2 Editing Basics

### 15.2.1 Keyboard Shortcuts and Modifier Keys

#### Caret Movement and Selection

The following list shows the shortcuts for changing the current selection. To simply instead move the current insertion caret position, omit the shift key where applicable.

- **Shift-ArrowKeys**: Extend current selection in the direction of the arrow key that is used.
- **Ctrl-Shift-LeftArrow or RightArrow**: Extend the selection by word.
- **Ctrl-Shift-UpArrow or DownArrow**: Extend the selection by paragraphs.
- **Ctrl-Shift-Home**: Select from the caret position to the top of the editor.
- **Ctrl-Shift-End**: Select from the caret position to the bottom of the editor.
- **Shift-Home or End**: Select from caret position to beginning or end of line respectively. Note the definition of line is not a paragraph, but is instead dependent upon the current soft wrap. To select to the beginning or end of the paragraph, use the above paragraph selection shortcut.
- **Shift-MouseDrag**: Extend the current selection using the mouse.
- **Double-click**: Select word. Can be used in conjunction with dragging to select a range by word.
- **Triple-click**: Select paragraph. Can be used in conjunction with dragging to select a range by paragraph.

## Deletion

Simple character deletion can be performed with the **backspace** key to delete the character prior to the caret position. The **delete** (sometimes abbreviated to **del**) key will delete the character *after* the caret position.

- Delete to beginning of current word: **Ctrl-backspace**
- Delete to end of current word: **Ctrl-delete**

### 15.2.2 Spell Checking

Spell checking in Scrivener can be accomplished either as you type, or after you are done writing a section. Depending on your preferred way of working, you'll want to set how this works in the Auto-Correction options tab (section B.6), where you will find other options such as which language to use, the ability to administrate your personal words, and whether or not Scrivener should try and fix common typos for you as you write.

With active spell checking turned on, if 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.

Alternatively, if you prefer to not be alerted to misspellings while writing, you can defer the spell check process for later. When you are ready to spell check a document, right-click anywhere in the document, and select **Spelling ► Spelling** from that menu. A familiar floating window will let you step through the document error by error, fixing mistakes and adding unique words to your personal dictionary as you go. As with most tools of this nature, if you type anything into the top field and click “Change” you can add a correction that is not shown in the suggestion list.

The **Ignore** button will temporarily ignore the misspelled word, while **Learn** will add the word to Scrivener's permanent ignore list. The **Guess** button can come in handy if none of the suggestions are correct. You can change the word itself in the top field, or select one of the suggestions and click this button in an attempt to narrow the list down to the correct word.

## 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 in 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, without affecting the font size, via the ability to zoom the scale of the text. Each split can have its own zoom setting; additionally full screen mode can have its own independent zoom setting as well. All of these settings are preserved between sessions and saved into the project file itself.

In the standard text editor interface, zoom can be set using the zoom tool in the footer bar (section 14.2). In full screen mode, this same tool is provided within The Control Strip (section 16.2).

You can also use the menu commands:

1. **View ► Zoom ► Zoom In (Ctrl=)** to increase the zoom factor of the text.
2. **View ► Zoom ► Zoom Out (Ctrl-Hyphen)** to decrease.

If you prefer to always work from a certain zoom setting, the default zoom level for both standard editors and the full screen mode editor can be adjusted in the Editor option tab (section B.5). Note that this setting will impact all *new* projects, or recently upgraded projects, not projects that have already been created, as each project can save its own individual settings.

### 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. The Scrivener-specific commands (the appearance of which will depend on the selection) are listed below:

#### Text Editing Contextual Menu

**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).

**Split with Selection as Title** Splits the current document into two documents using the current 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 Managing Documents with Split and Merge (subsection 15.3.3).

**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 11](#)) for more tips on moving text around and organising information in the Binder.

**Add Selection to Auto-Complete List** Adds the selected text to the project's auto-complete list, which can be maintained with the **Project ► Auto-Complete List...** menu command. Read more about this feature in *Auto-Completion* ([section 15.7](#)).

**Scrivener Link** 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 9.4](#)).

**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.

**Highlight Color** Provides the highlight menu, allowing you to select a highlight colour for the selected text, or to clear it. Read more about this and the above feature in *Text Colour and Highlights* ([section 17.4](#)).

**Font** Provides basic character attributes, such as bold and italic, as well as access to the main font window.

**Writing Tools ►** Quick links to a few handy web searches. The current word, or selected text, will be sent to the search engine or reference site of your choice.

**Insert Table** Inserts a table at the current caret position and brings up the table configuration window.

**Statistics Readout** This non-selectable entry at the bottom of the contextual menu displays the word and character count when an active text selection exists in the editor. It is thus useful for getting statistics on a portion of the currently edited file.

### 15.3.3 Managing Documents with Split and Merge

The ability to expand and contract the outline in a fashion which best suits how your work is important, so there are two tools which make it easy to fashion your outline into as broad or detailed a map as you require.

Before discussing the specific tools which let you cut apart documents or sew them back together, it would be useful to also discuss a few methods that you can use to mitigate the regular need for them.



When you wish to merge two or more documents together, it is good to ask whether or not this action is necessary. With most applications, and thus the habits we have cultivated from using them, the answer to whether or not one should merge is, yes. However Scrivener offers two powerful answers for the merging problem. First, for visualising smaller pieces of a document as a single document, but only temporarily, you can select any container in the binder, or even just a scattering of items, and choose to view and edit them much as though they were a single document (section 15.8). Second, the compilation (chapter 22) system enables you to publish your final work as a single document, no matter how many pieces it may be divided into within Scrivener.

For splitting long documents, in some cases you may be able to utilise use import tools which do most, if not all, of the splitting for you. **File ► Import ► Import and Split...**, will let you supply a separator, or a string of characters, that mark natural sections within the original file. Whenever Scrivener encounters this separator, it will automatically start putting the remaining content into a new binder item, continuing to split into new binder items until it runs out of material. Secondly, if you've done your original outlining in a dedicated outliner program, it will likely support an OPML export, which you can drag into Scrivener's binder. It will recreate the outline structure, placing any original outline notes into the text area for each outline header.

For most other cases, read on.

### Splitting The Document

Often, when importing material that you have composed in other applications that do not support an easy to use built-in outline, your text will arrive in Scrivener in one large chunk. Where you choose to break this long document apart is up to you, and the split tools make it easy to do so.

To split in a document 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 (Ctrl-K)** This will create the new document directly below the current document in the outline. It will contain everything that followed the caret position, and this material will be removed from the original document. 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.

After splitting, you will be given the option to name the new document in the Binder.

**Documents ► Split ► with Selection as Title (Ctrl-Shift-K)** This alternative method will only appear when a range of text has been selected in the editor. This method works identically as “at Selection” does, only the selected text will automatically become the title of the new document. This text will remain selected after you split, making it easy to remove it, or style it as a header.

It is not possible to undo a split action, but you can use the Merge feature to effectively undo any unwanted splits.

**Pro Tip:** You might also find the ability to search by formatting (subsection 20.1.4) to be of use as well, as often the places you will want to split the document will coincide with headers. Once you have used the Find by Formatting panel to find a header, you can close it, then use the **Split with Selection as Title** function (**Shift-Ctrl-K**) with the text the search selected for you, continue finding results with **Shift-Ctrl-F3**, splitting and repeating until the document has been completely split up.

### 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. 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, then the merged document will retain the original outline order.
- When using a Collection to select items, the *collection order* will be used to establish the structure of the merged document.

When it comes to merging, Scrivener will attempt to retain as much meta-data as is logically possible. The synopses, notes, keywords, references, and list of snapshots will be combined together, much in the same fashion that the main text will be. Meta-data

which cannot be combined (such as Title, Label, any custom meta-data, compile option flags, and so forth) will use the top-most document as a reference point. For example:

• Document A

- Synopsis: “This is the first document”
- Keyword: “Apples”
- Status: “First Draft”

• Document B

- Synopsis: “This is the second document”
- Keyword: “Carrots”
- Status: “To Do”
- Notes: “Mix these two together in a blender for a great tasting, healthy beverage.”

Both Document A and Document B are selected in the Binder, and **Documents ► Merge** is invoked (**Ctrl-M**), producing the following single document (with the bold items indicating material used from Document B):

• Document A

- Synopsis: “This is the first document

**This is the second document”**

- Keywords: “Apples”, “**Carrots**”
- Status: “First Draft”
- Notes: “**Mix these two together in a blender for a great tasting, healthy beverage.**”

#### 15.3.4 Typewriter Scrolling

You can turn “typewriter scrolling<sup>1</sup>” on for the main editors and for the composition mode editor independently via the **Format ► Options ►** sub-menu (**Ctrl-Meta-T**). When it is switched on, just as with a typewriter, the line of text you are typing will remain centred vertically in the editor, once your text is long enough to reach the middle of the

---

<sup>1</sup>Typewriter scrolling was an innovation of the The Soulmen’s *Ulysses*.

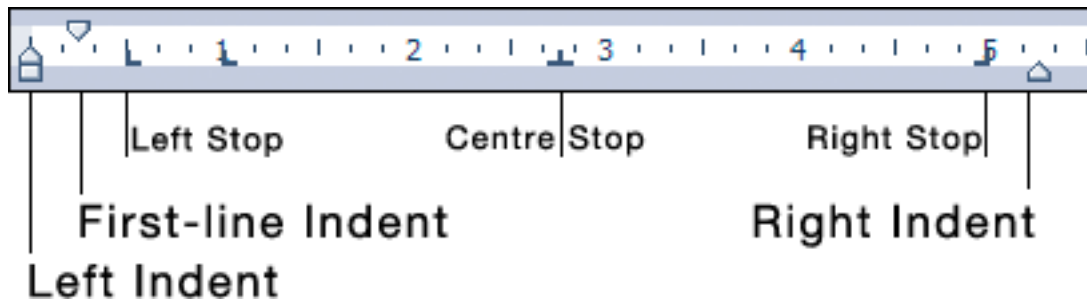


Figure 15.1: Example Ruler Settings

screen. This means that your eyes do not have to be glued to the bottom of the screen all the time when typing in a long piece of text.

When Typewriter Scrolling is enabled, your scrollbar will act as though there is more content than the document actually has. This is because we add extra spaces below the last line in the document in order to keep it in the middle. These spaces do not actually exist in the file, and will vanish when the feature is disabled.

## 15.4 Formatting Tools

### 15.4.1 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 **Format ► Show (Hide) Ruler** or **Ctrl-R**. This will present a simple indenting and tab stop interface (Figure 15.1).

The following elements can be found in the ruler:

**Tab Stops** Left stops can be created by double-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:

- Left: This is the standard type. Text will be left-aligned, with the first character indented to the position of the tab stop.
- Center: Text will be centre-aligned, using the position of the tab stop as an anchor point; this can be anywhere on the line.
- Right: Will right-align text with the right-indent set to the point of the tab stop.

- Decimal: Most often used for aligning rows of numbers, so that the system decimal character is lined up vertically with everything 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.

**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).

Scrivener’s default formatting settings demonstrate the best way to use indentation, if you need to use it at all.

**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 it to wrap before it gets to the edge of Scrivener’s editor window. 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 ► Formatting to Default Text Style](#) menu command to clean up large amounts of text at once.



Figure 15.2: Format Bar: displayed in sections

**Ruler Conversions:** When working with units in Scrivener, keep 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 paper left edge, 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 by adding in the margin to the value you see in the ruler. For example, if you are using 1” margins and you need a right-indent of 1”, the right-indent should be set at 5.5” for US Letter, not the standard 6.5”. Optimally, your formatting can be deferred and set in the compiler, where formatting and paper dimensions can be operated upon in concert.

Using tab stops while writing is as simple as pressing the tab key to advance to the next available stop. 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 [Format ► Options ► Show \(Hide\) Invisibles](#).

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 ► Text ► Copy Ruler](#) and [► Paste Ruler](#) menu commands.

#### 15.4.2 The Format Bar

The Format Bar provides quick access to common formatting features, and can be customised in the [Tools ► Customize Toolbars...](#) menu item. If you are looking for tab and indent controls, you can still toggle that per text view with [Ctrl-Shift-R](#). The visibility of the Format Bar is toggled with [Alt-RightArrow](#).

The first section ([Figure 15.2](#)) of controls handle font face, size, variant, and line spacing. All of these tools, with the exception of line spacing, 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. Line spacing will impact the entire paragraph immediately, it is not possible to have more than one line-spacing setting per paragraph.

- The first drop-down menu lets you select from all of the available font families installed on your computer.
- The second provides the variants available for that family. Not all fonts come with a full complement of variants. If for some reason you cannot change a selection of text to bold or italic, check this menu to make sure that the current font supports these variants.
- The third menu provides a quick list of common font sizes.
- The line spacing drop-down provides some quick presets, but for more complex multi-type spacing, select the **Other Spacing...** item at the bottom of this menu.

The next three buttons (section two ([Figure 15.2](#))) provide toggling shortcuts to the most common text formatting tools: bold, italic, and underscore, respectively. When the caret is in a range of text that matches the format style, the appropriate buttons will appear activated.

The next portion in this section allows control of paragraph-level text alignment. Paragraph-level alignment comes in four flavours:

- Left Align (default)
- Centre Align
- Right Align
- Full Justification

In all cases, these adjust the entire paragraph. If you need multiple alignments on a single line, you will need to add special alignment tab-stops to the ruler ([Ctrl-Shift-R](#)).

In the third section the left colour selection drop-down will alter the foreground text colour. Click the button once to set the selected text to the depicted colour (black by default), or click the arrow to access further options. You can opt to strip out all colour code from here, and access the standard colour palette for custom colours. For more information on using the colour palette, see [Text Colour and Highlights \(section 17.4\)](#).

The second colour selection drop-down works in an identical fashion to the first, but it sets the background highlight colour for the selected range. Click once to toggle the

current highlight (or use **Ctrl-Shift-H**) for the selected text, or use the arrow button to access further options as described above.

The final, fourth section of the Format Bar accesses tables and lists.

Clicking the Table button will bring up a configuration palette so you can change the selected table's dimensions and aesthetic characteristics. To create a new table, click the arrow button and select **Insert Table...**

The list tool provides most common list formats. Clicking the button once will start a standard bullet list, but lists can be enumerated or bulleted in a variety of methods by clicking the down-arrow. They will transform the current paragraph (even if empty) into the first list item, and will continue generating new list lines until you enter a blank line. While using lists, at the beginning of the list line, it is possible to indent or outdent lists with **Tab** and **Shift-Tab**.

The format bar can be customised completely using the **Tools ► Customize Toolbars** menu command ([Appendix C](#)).

### 15.4.3 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 the Editor options tab ([subsection B.5.1](#)), 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 ► Formatting to Default Text Style** 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 exception is the "Convert font only" option, which acts as a handy way of preserving all paragraph level formatting, without individually checking off their options.

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, so it is safe to use, but if you are unsure of whether or not the result will be favourable, either Snapshot (section 15.6) 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.

Some simple tips for preserving common formatting based ranges:

- **Preserving Block Quotes:** Block quotes are usually indicated entirely by some form of increased left, and sometimes right as well, indent. Preserving tabs and indents will protect most block quotes from being reduced to normal text by this tool.
- **Preserving List Spacing:** Some styles call for lists to be single-spaced even if the text around them has a greater line-height or inter-paragraph spacing. Using “Preserve line spacing” will protect these lists at the expense of not applying your preferred line spacing to standard paragraphs.
- **Preserving Titles:** If you’ve got a document with a lot of headers already formatted into it, you could try preserving font size, style and line spacing, but this will in many cases be the three things you *want* to apply. Titles are going to be harder to manage automatically, and may require manual effort to keep their formatting as intended. Another thing you can consider doing is switching to using Scrivener’s philosophy of using the compile feature to generate titles from your binder names, and leave worrying about special formatting for headers to the computer.

One last tip is to use “pre-emptive” format conversion. If you know that you will wish to discard the formatting of the text you are pasting, 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 Inline Images

While images themselves cannot be placed into the Draft as binder items, you can insert an image into the text either by dragging an image file in from Windows Explorer; dragging an image document in from the Binder, Outliner or Corkboard; or by selecting **Edit ► 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 Inline Images (subsection 15.5.2).

To resize the inline image, right-click on it, and select **Edit Image...** 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. The **Reset** button will return the image settings to the state they were in when you opened the panel, but will leave the panel open. Note that some images types 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.

When using MultiMarkdown, inline images (and linked images for that matter) will be converted to MMD syntax, and the images themselves will be gathered into the compile folder for you.

### 15.5.1 Working with Print-Ready Images

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.

### 15.5.2 Linked Inline Images

Linked images are useful when the actual graphics in your project text are incomplete or placeholders for larger, production-ready graphics. 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 project*.

Note that for increased performance, Scrivener uses a cached version of your image when you create a link to it while the project is open. Whenever you re-open the project, this thumbnail will be updated with the current image on the disk.

To create a linked image, use the **Edit ► Insert ► Image Linked to File...** menu command.

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.

**A Note on Links and 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. 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.

## 15.6 Using Snapshots

In essence, the theory behind a Snapshot is very simple. You can think of it as a way of manually saving your file for future reference. In a program like Scrivener which automatically saves your work as you work, you lose the benefit of being able to work "off the disk" for a while, and choosing at defined moments in time to either save what you've done, or discard it and return to what you have on the disk at the last save point. Of course, on the other hand, if your computer crashes (and as much as we try to avoid it, sometimes Scrivener crashes too) with an auto-saving program, you might only lose a few minutes of work, or even nothing at all, rather than everything you've done since the last save.

Snapshots help you bridge the gap between these two ways of working. It is like saving your document as you go, the only difference is, you get a *record* of each save point.

That means you can go back and examine the evolution of the file and pick out just the right spot to restore from. 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. With Scrivener, you can just go back through your Snapshot history, 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 this usage, some people like to use Snapshots to set “milestones” as well as just capture periodic changes. Scrivener accommodates both styles of working, as you can take “Untitled” Snapshots with a single keystroke. You can also name your snapshots either when you take them, or later on. Named snapshots are easy to pick out of the crowd of Untitled ones, and it also makes it easier to clean up old routine Untitled ones once they've outlived their usefulness.

Theory aside, snapshots provide a way of backing up and saving different versions of individual documents. Taking a “snapshot” of a document takes a copy of the text of that document as it exists 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 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. Snapshots only affect the *text* of documents (not notes, synopses or meta-data), and are therefore only available for documents that contain text (files and folders).

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”. You can also tell by looking at the bottom of the Inspector pane on the right. If there is an asterisk in the corner of the Snapshot button, then it has Snapshots.

There are two primary ways to manage Snapshots: using the **Documents ▶ Snapshots ▶** sub-menu, and the Inspector pane (subsection 18.3.4), the latter of which will be covered completely in the referenced section. You can also conveniently use the **Ctrl-5** keyboard shortcut while editing a document to take a quick snapshot in the background.

### 15.6.1 Creating Snapshots

While working on a document, there are three easy ways to snapshot it for future reference.

The key thing to understand about Snapshots is that they provide a way to set save points for *individual* items in the binder. 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.

That aside, it is simple to quickly snapshot multiple documents.

1. Select the documents you wish to snapshot in the binder sidebar, corkboard, or outliner
2. Press **Ctrl-5**, or **Documents ► Snapshots ► Take Snapshots of Selected Documents** to create a snapshot into each item's individual snapshot list.

Snapshots can later be accessed in the Inspector pane, or **Documents ► Snapshots ► Show Snapshots**.

3. The alternate way of taking a snapshot is to supply them with a title as they are created, use **Documents ► Snapshots ► Take Snapshot with Title**, or **Ctrl-Shift-5**.

This can be done for single or multiple selections as well. In the latter case, the title will be added to each document in the selection. Snapshots can also be titled at a later time using the Inspector.

**Extensive Annotation:** A trick for more thoroughly commenting on the purpose or state of a Snapshot can be to combine it with the use of inline annotations. Consider writing a short synopsis of what the snapshot was taken for at the top of the document before snapshotting, and then removing 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.

### 15.6.2 Managing and Comparing Snapshots

Existing snapshots are all managed in the inspector pane. To jump to snapshots, click the button with the camera in the Inspector footer bar or use the **Documents ► Snapshots ► Show Snapshots** menu item. In the snapshots pane, you can create new snapshots from the current text with the **+** button, remove old snapshots by selecting them in the snapshot list and clicking the **-** button, or roll back to a prior version of your text using the **Roll Back** button.

More details on the functions available in the inspector can be found in Snapshots Pane (subsection 18.3.4).

## 15.7 Auto-Completion

There are three forms of 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.

**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 definition ([Format ► Scriptwriting ► Script Settings](#)).

### 15.7.1 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 a customisable substitution engine. These can all be set in the Auto-Correction options tab ([section B.6](#)). Each option is documented more fully, there.

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.

### 15.7.2 Scriptwriting Auto-Completion

Scriptwriting mode, by default, engages a more aggressive auto-completion method, which 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 tied to the element you are currently working in. For example, “MOMENTS LATER” will appear whilst typing in a Scene Heading element, but will not appear in a Dialogue element.

## 15.8 Editing Multiple Documents

The “Scrivenings” view mode ([View ► Scrivenings](#)) allows you to edit multiple text documents as though they were one long document. 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 scenes for a chapter, and then edit them all together in an Scrivenings session to see how the chapter works as a whole.

When non-text items are included in a Scrivenings session, an uneditable grey bar will be inserted in the place where media would ordinarily be shown.

### 15.8.1 Viewing Multiple Texts as One Document

Scrivenings mode is, like corkboard and outliner view modes, another way to view multiple selections of documents. How you view more than one document changes, depending on your approach. Viewing more than one document can be as simple as clicking on a folder. By default, this will view the items within that container as index cards on a corkboard. However using the Group Mode segmented control, you can easily switch your view over to Scrivenings, the left-most icon showing a stack of papers. Another way of switching is to use the **View ► Scrivenings** menu command, or pressing **Ctrl-1** on your keyboard. As with corkboard or outliner, you can also form scrivenings sessions by selecting multiple items in the binder.

Unlike corkboard and outliner, scrivenings can be taken into full screen mode. Simply press the Full Screen button in the toolbar after setting up the session. Read more about full screen mode ([chapter 16](#)) for general information on this feature.

You can tell when you are in Scrivenings mode by the horizontal divider lines that will be drawn between each section represented in the session. Meanwhile, the inspector will follow along so you can see which document you are in by glancing at the index card in the inspector.

Scrivenings sessions are one of the few areas in Scrivener that consider your text from a “flat” perspective. There is no indication of depth while editing documents. A prologue at the very top will be printed right along with items which are buried in sub-folders. This is intentional, so you can concentrate solely on the same flow of text that your readers will use.

The per-document goal meter ([subsection 14.2.3](#)) in Scrivenings mode acts differently from when editing a single document. If the session was formed by selecting a container, then adjustments made to the goal will impact the container, and the progress metre will count up all constituent text within the session, not just the container’s own text field. When using a session that has been formed from an ad hoc group of items selected individually, then the target button will be non-functional.

### 15.8.2 Editing with Scrivenings

For the most part, editing in Scrivenings mode is just like editing single documents one at a time, thus most of the tools that you have learned so far will be available to you. You

can select text and format it, copy and paste it to another application, create snapshots of individual portions of the session, and so on.

Due to the manner in which multiple real files are being combined as though they were one “file”, it is not possible to select across a file boundary, or to delete text from across multiple files at once.

These limitations are unavoidable. If you were to delete a range of text that spanned two or more documents, Scrivener would have to guess where one document now ended and the other now began—and if the whole of the text of one document had been deleted in the process, Scrivener would have to decide whether to delete that document, too. These are all decisions that are better left to you.



# Full Screen

Full Screen mode allows you to edit your text in a distraction-free environment. 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 Full Screen mode, with a virtual page treatment on a background (by default, black). Meanwhile you will have full access to your document’s meta-data in a floating low-impact Inspector panel. 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 Full Screen 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 Full Screen, 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 Full Screen, 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 Full Screen mode will use that same mode. This is the easiest way to load an entire container’s text into Full Screen mode, as it requires no further selection.
2. *Pre-load documents into history*: If multiple items are selected and Scrivenings mode is not enabled, then each item will be loaded into the Full Screen mode’s history queue. Working in this fashion, you can use ⌘-( and ⌘-) to flip between the documents without returning to the main project window.

Launch Full Screen mode either by hitting **F11**, or select **View ► Enter Full Screen**.

When you first enter Full Screen 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 Full Screen mode. You will also briefly see the Control Strip (detailed below). To retrieve it, 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. Once moved back up, the control strip will disappear out of view.

## 16.1 Using Background Image and Textures

Textures can be assigned to the background area on either side of the “paper”. Textures are graphic files you select which will be tiled to fill the entire visible area. Textures can be configured in the Appearance preference tab (section B.2), as an additional option in the customisable colours section.

## 16.2 The Control Strip

The Full Screen control strip contains various controls for manipulating the interface elements available to Full Screen mode. All of these settings operate on a project-by-project basis, as opposed to the global scope of the application options.

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 Full Screen mode stores its own scale setting, independent of the editor splits.

**Paper Position** Select where the “paper” (the column of text) should be anchored on the screen: on the left, in the centre or on the right. You may want it in a non-central position, for instance, if you have a lot of notes to which to refer and you want the inspector open (and wide) next to the text.

**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 Alt key. It will replace the Paper Width slider, from above. This sets how tall the paper column should be, giving you a “letterbox presentation” at its more extreme settings.

**Keywords** Brings up the standard Project Keywords ([section 10.2](#)) which you can use in conjunction with the inspector (below) to assign keywords to the current document.

**Inspector** Brings up the floating Inspector panel (see below).

**Go To** Operates as a handy navigation tool, and functions identically to the main [View ► Go To ►](#) menu. Using this, you can switch between documents without returning to the main project window.

**Dynamic Controls & Statistics** 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.

In scripting mode, this portion of the strip will display element hints as well as the element selection menu (which you can always call up with **Ctrl-\`\`**, even if the strip is not visible).

**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 full screen mode.

**Exit Full Screen Mode** The button at the far right will exit Full Screen mode. You may also press the **Esc** key, or **F11** to return to the main project window.

## 16.3 Floating Inspector HUD

The full screen mode version of the inspector gives you full access to the inspector data which will float over the editor so it doesn't get hidden behind the background. For a full discussion on these views and what they represent, read [Meta-Data Types \(section 10.1\)](#), and [Inspector \(chapter 18\)](#).

## 16.4 Customising Appearance

Appearance settings can be set in the Appearance option tab ([section B.2](#)). Refer to the documentation provided for this pane for greater detail on available options.

On account of limitations in the text system, it is not possible to temporarily override the font family in full screen mode. The text colour can be overridden, but changing the font would require an irreversible adjustment to your formatting.

# Annotations and Footnotes

As a tool designed for the production of texts, there are methods for facilitating not only the writing process, but the editing process, and the “public notation” process which is typically expressed via some form of page footnoting, or endnoting at the conclusion of the work. This chapter will cover all of the various methods that Scrivener provides. We will cover:

- **Annotation:** 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 supports the RTF standards for commentary, and will convert any word processor applied notes to annotations for your own use, which can in turn be exported back out as comments when compiling.
- **Footnotes and/or Endnotes:** being a drafting tool, Scrivener makes no attempt to typeset either of these as you write, however it is easy to insert notes precisely where you need them, and when you compile they will be presented according to typical standards. 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.

Marking your text is one thing, finding those markings amongst a 100,000 word draft is another. The robust Find by Format and Text Function ([subsection 20.1.4](#)) makes it easy to quickly step through your draft, point by point, isolating and addressing issues or reviewing changes that you’ve made.

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, which are attached

as a special form of “link” to the base text of your book, with the content of the note stored off to the side in the inspector, much as in many 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 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 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. In some special cases, using all four methods together will afford you with additional compile powers, such as handling multiple note streams, or using annotations as “functional” text and linked comments strictly as notes. The next few sections will address each of these types in more depth.

## 17.1 Inline Notation

Scrivener sports a unique method for taking notes and indicating footnotes, by placing them directly into the main text stream itself. 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 in-line 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.

Despite the advantages, there are still cases and tastes which would prefer to keep notation out of the main text area. If you have interest in commenting on your work or adding footnotes to it without adding text to the main editing area, please skip down to Linked Notation ([section 17.2](#)).

**What They Are:** Inline notation is fundamentally a form of formatting, just like any other formatting spans in your document, such as bold or italic. Yes, Scrivener has tools for working with notations in unique ways, but they will respond to many of the actions that are designed for working around formatting, such as Paste and Match Style. When cutting and pasting text into and out of notations, try to think of them as analogous to bold text. To paste some external text into a note, you’ll need to use Paste and Match Style for the text to acquire the notation formatting. The same is true

going in other direction. 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 at the writing phase. 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 create an annotation, you can use **Format ► Inline Annotation** or press **Ctrl-Shift-A** on your keyboard. 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. For footnotes, everything is the same; just use **Format ► Inline Footnote** or **Ctrl-Shift-F**.

To create a new footnote or annotation from text that does not yet exist, you can use these same commands to start typing in the desired notation mode wherever your cursor is placed in the editor.

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.

### 17.1.1 Inline Annotations

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). In other words, if you place an annotation between two sentences, make sure that there is only one space (or two if you use two spaces between sentences) separating the sentences outside of the annotation. On the other hand, when you choose to compile annotations into the document inline, all spacing within the annotation will be preserved; the end result will look like it does in the main editor.

Since you can choose to omit annotations from the text when you export or print it, they are very useful for jotting down notes and ideas inside the main text that are not really part of the main text, but which would be better placed next to the section of text to which they refer, than in the Notes pane. When you export to RTF format, annotations can also be converted to RTF comments that appear in marginal boxes in compatible word processors, rather than omitted, as well as a few other display options for other word processors or formats. For details on what options are available for omission or conversion during compile, see Footnotes/Comments]cpane-footnotescomments ([section 22.12](#)).

While it is possible to use annotations to create footnotes, this compile-time conversion has fewer options available to it than standard footnotes. For fully featured footnotes, continue to the next section.

### 17.1.2 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 upon export or print-outs (though do not have to be) whilst footnotes become “real” footnotes (or endnotes) upon export, and thus have more options available to them for styling footnotes, particularly with the RTF format option. 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.

As a drafting application, Scrivener does not do typeset footnotes itself; rather, you define ranges of text that you want to be exported as footnotes. Thus, when the text in Scrivener is exported to RTF format, and opened in a compliant word processor, it will take on the familiar appearance of a numeral reference linked to a line at the bottom of the page with a matching numeral. Footnotes will also be exported in a variety of fashions for other formats as well, such as eBooks, MultiMarkdown syntax (and therefore as proper footnotes in LaTeX), as look-alike footnotes in HTML and plain-text, amongst others. For all options available, read Footnotes/Comments ([section 22.12](#)).

### 17.1.3 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 Format and Text Function ([subsection 20.1.4](#)).

## 17.2 Linked Notation

Creating and reviewing linked notation is made very simple with Scrivener’s innovative sidebar approach. Unlike methods which use positional margin areas to display the note,



or typesetting driven methods which attempt to display a footnote exactly as it will appear in print, linked notes give you the power of placing meta text outside of the base text, while presenting this information in a stack which, while ordered as they appear in the document, does not otherwise reflect the spatial distance between notes. This means 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. Click on the disclosure arrow to collapse or expand a note.

Meanwhile, 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
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.

Clicking on any anchor highlight will highlight 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.

To use linked notes you must in most cases have anchor text to attach the note to, in the same way you need some text to create a hyperlink with. 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, it might be easier to use inline annotations or the document notes sidebar in the Inspector ([subsection 18.3.1](#)), instead.

To create a new note, either:

1. Select the text you wish to notate and then invoke the appropriate menu command or keyboard shortcut. To make a comment, use **Format ► Comment** or **Shift-F4**. To make a footnote, use **Format ► Footnote** or **Shift-F5**. This method is quite useful for comments. The highlight range doubles as the specifier; if you intend to address a piece of prose in your text, the highlighted text is immediately obvious as the problem area.
2. Place the caret 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.

In addition to the menu and shortcut commands, you can also click either the **+** button or the **+fn** button in the header bar of the Comments & Footnotes inspector to add comments and footnotes to the selected text, respectively.

**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.

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.

### 17.2.1 Managing Linked Notes

Deleting notes can be accomplished by selecting the note(s) in the inspector and then clicking the **-** button, by clicking the small **X** button in the top-right of each note, tapping **Shift-Del** or by simply deleting the text that the note is anchored to in the editor.

### 17.2.2 Linked Comments

Linked comments can be assigned a custom 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, a convenience tool has been designed into the right-click contextual menu when clicking on the note in the inspector. Six default colours have been provided for rapid usage, but you can also opt to use the colour palette to change the colour to a custom selection.

Scrivener will also 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.

As with inline annotations, the primary usage for comments will be to add commentary to your text; commentary that isn't intended to appear in the final cut. However it

is possible to adjust this behaviour in the compiler, and even opt to compile comments as footnotes. Read Footnotes/Comments ([section 22.12](#)) for further details on available options.

### 17.2.3 Linked Footnotes

As with inline footnotes, linked footnotes will by default appear with a grey background, and will likewise appear grey in the inspector. Footnote colours cannot be individually changed, though you can adjust the global application default for footnote presentation in the Appearance option tab ([subsection B.2.2](#)).

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. The positioning point is at the end of the anchor highlight area, not the beginning. If you wish to place a note at the end of a sentence, you would highlight up to the last word of that sentence (where you choose to start highlighting is at your discretion). This will be done for you automatically if you place the cursor anywhere in that word or at the end of the word, and create a footnote.

#### Footnote Highlight Style

See Inline Footnotes ([subsection 17.1.2](#)) for general tips on footnotes in Scrivener.

#### Compiled Footnote Numbering

By default, when you compile the document to any format, linked footnotes will be numbered in the order they appear, and this number will be saved into the footnote as a static identification of it. This is entirely unlike standard word processors, which generate the final document appearance while you are working. Scrivener's numbering system is designed to work in a more stable fashion, allowing author and editors to refer to static numbering that is tied to the last generated hard copy.

Consequently, the addition or removal of footnotes from the sidebar will not disrupt the stored numbering of existing footnotes. This way you can continue to write and make changes to the source project without losing frame of reference with the hard copy that is out for proofing. An option has been provided to prompt you whenever you compile. This way you can do test compiles on your own without disturbing the original numbering that is out for edit.

Compile footnote numbering can, with very large texts that have a lot of footnotes, take a long time to process. If you find compiling takes an inordinate amount of time, you should try disabling the automatic feature (causing it to work on-demand) via the **Format ► Options ► Prompt Before Updating Footnote Numbers** menu command.

### 17.2.4 Finding Linked Notations

Scrivener provides tools for stepping through your entire project, searching for inspector comments and footnotes. Read more about this feature in Find by Format and Text (section 20.1.4).

## 17.3 General Usage Tips for Notation

### 17.3.1 Stripping Out All Notation

If you wish to strip out all (inline and linked) notes from a text at once, this can be easily done by copying the text with **Edit ► Copy Special ► Copy Without Comments or Footnotes**, or **⌘-C**, and then pasting directly over the copied text. Do note this will also destroy footnotes, so use this command with care.

### 17.3.2 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 formatting). Consequently, adjusting the Formatting preferences (subsection B.5.1) 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**. This menu option will only be available when the note is out of editing mode.

### 17.3.3 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**. Optionally, you can 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.

## 17.4 Text Colour and Highlights

Text colour can be applied to your document, like switching to a coloured pen, in a fashion similar to any other formatting range. Select the text you wish to change the colour on, and either use the **Format ► Font ► Text Colors ►** sub-menu, or use the text colour selector (letter 'A' with an underscore in colour) in the format bar 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. If you wish to change the colour, click the down-arrow beside the text colour control.

The menu will contain the following functions:

1. Built-in presets (black, red, etc.)
2. “More...” which presents the custom colour chooser
3. “Remove Color”

Setting a marker highlight to a range of text is quite similar, use the icon of a highlighter pen (which will be coloured depending upon the last highlight you used; yellow by default) on the format bar, instead. As with the text colour tool, clicking once will re-use the last highlighter colour, and clicking on the down-arrow will access the full menu. For highlighters, this can also be accessed in the [Format ► Highlight ►](#) sub-menu.

**Colours and Compiling:** It is important to know that text colour will be exported when compiling to most formats. They cannot be automatically removed without overwriting your styling, and so should be used either when it is your intention to have them in the final draft, or as temporary tools that you periodically strip out when done with them, or otherwise intend to use the compiler to reformat the manuscript.

To wipe out colours, set the Formatting options pane to override text and notes formatting, and then specifically set the text colour to black and the highlight to white. Please read about the Compile Formatting options pane ([section 22.8](#)), for more information.

Some formats (and some word processors) do not support colour and highlights, and so even if they are compiled in, they may not always show up. When using a workflow that omits these tools, you can feel safer in leaving them in when it comes time to compile.

Scrivener provides tools for stepping through your entire project, searching for text colour and highlighters. Read more about this feature in Find by Format and Text ([section 20.1.4](#)).

## 17.5 Text Markings and MultiMarkdown

Since MultiMarkdown has scant tools for working with text formatting, many of the above marking techniques have no direct analogue, with the exception of footnotes. In the case of annotations, it is possible to use embedded HTML coding to simulate comments, and they will be marked in such a way that it is easy to get both the colour information and the fact that they are comments, from an XSLT stylesheet. Other markings, such as text colour, highlighting, and revision levels, will be completely dropped from the compile.

### 17.5.1 Using Comments with MultiMarkdown

Inline annotations can be independently stripped from the document during compile, or left in.

Since there is no direct analogue for comments<sup>1</sup>, Scrivener employs the usage of MMD's ability to handle raw HTML coding in the document. When using MMD, it is possible to pass HTML through the parsing engine so that it lands directly in an HTML document as a final product.

This behaviour can be disabled in the compile settings, and comments will be passed through to the document in the same manner that they are using the plain-text export, enclosed in markers.

### 17.5.2 Using Footnotes with MultiMarkdown

When compiling a draft using MultiMarkdown, the compiler can be instructed to produce proper footnote syntax for all of your inline footnotes in the project. This will be done using MMD's referencing syntax, where a footnote reference is placed at the point of the note in the text, and the content of that note is placed at the bottom of the file with a link to that reference marker. Scrivener will generate all of the reference tokens so that you do not have to worry about coming up with unique markers. Just use Scrivener's footnote tools, compile, and you'll have a completed document with footnotes.

If you do not wish to see footnotes in the final product, you can disable this function in the compiler, under the Footnotes/Comments pane, by checking "Remove footnotes".

### 17.5.3 Text and Highlight Colour with MultiMarkdown

In short, nothing is done to export Scrivener's text colour and highlighting features to MultiMarkdown, as there is no suitable analogue for these tools, and in most cases since they represent a function of a document's meta commentary, are not considered to be suitable for *semantic* transfer.

Rather than looking at this as a limitation, you can consider it to be a feature. As with most purely rich text features in Scrivener, they can be used liberally to annotate and comment upon your work, without fear of them every showing up in the final product.

---

<sup>1</sup>While it is true, MMD uses HTML, and HTML has a perfectly valid syntax for comments, it is not viable to use it because MMD already has an advanced secondary usage for HTML comments, in that raw LaTeX can be passed directly through the parsing engine and land in the final .tex file. Thus, usage of the HTML comment engine for general comments or other purposes would cause undesirable side-effects.

# Inspector

The inspector is a panel on the right-hand side of the project window which displays all meta-data, snapshots, and notes associated with the document shown in the editor that most recently had the keyboard focus, or in the case of a corkboard or outliner, the last card or row selected (which implies the container itself when all cards/rows are deselected). Since the special root folders in the binder—Draft, Research and Trash—have no meta-data associated with them, the inspector just shows the project notes or references when these items are selected.

The inspector is broken down into sections, or “panes”. These panes can be toggled by clicking the buttons along the very bottom of the inspector. The first three display the common meta-data features of the item, while the snapshot section has dedicated panes that take up the entire inspector.

With the exception of the aforementioned root folders, any item in Scrivener has access to all meta-data and notation capability. You can use Notes to write about a painting you are studying in the editor, attach gallery URLs and research papers to it with References, or organise it with keywords. Snapshots however will only be available to items which allow text editing in the editor, namely folders and files.

You may toggle the visibility of the inspector with **View ► Layout ► Show (Hide) Inspector**, or by pressing **Shift-Ctrl-I**.

## 18.1 Synopsis Card

At the top of the inspector (for the Notes, References and Keywords panes), you will find a depiction of the document’s index card. (The corkboard’s special features, such as pins and stamps, will not be rendered on the card, as they can be readily viewed in expanded fashion in the inspector) As with the corkboard, you can edit the title and synopsis right on the card, updating the document’s corresponding information.

There are two controls along the top of the card on the right-hand side. The first is a drop-down menu with an index card graphic in it. Click here to select between text synopsis and image synopsis. In graphical mode, you can drop in an image from the Finder (or the binder) which will be used to “represent” this document on the corkboard. The

text synopsis will not be erased, and in fact will still be used in the outliner and various export methods, but in the inspector and corkboard, the graphic will be used instead. Read [Placing Images on the Corkboard \(subsection 12.1.5\)](#) for further information.

The button on the far right will automatically generate a synopsis based on the current text in the document. If text is selected in the editor, it will be used to fill in the synopsis; otherwise, the first few lines will be used. When using image synopsis mode ([subsection 12.1.2](#)), this button will be replaced with an **X** which when clicked will erase your graphic selection.

The Synopsis section can be collapsed to make space for the other elements of the inspector.

See also: [So What are Index Cards, Anyway? \(subsection 12.1.1\)](#)

## 18.2 General Meta-Data View

This collapsible section appears in the middle portion of the Notes, References and Keywords panes of the Inspector. It displays the basic document meta-data and provides controls for editing it. Read [Meta-Data Types \(section 10.1\)](#) for further detail.

## 18.3 Document Support Panes

The four views of the inspector present further access to notes, reference links, keywords, and text snapshots, respectively. The first three share space with the synopsis and meta-data panels, the last takes up the entire inspector interface.

The four panes can be accessed at any time by clicking on the icons in the bottom of the inspector.

### 18.3.1 Document and Project Notes

Document notes provide a rich text writing area that is outside of the main text area. You can use notes to store any auxiliary information about the current document for reference whilst writing, without cluttering up the binder with note files or the main text with comments and annotations. Clicking the header title of the notes area will bring up a pop-up menu from which you can select “Document Notes” or “Project Notes”. Document notes are always attached to the current document in the sense that you cannot view them unless you have loaded their respective document in the binder.

Project notes are similar to document notes, except they are visible to the entire project, no matter what document you have selected.



### 18.3.2 References

The References table allows you to hold links to related material within the project itself, elsewhere on your hard drive or on the Internet.

General, or vague linking between items can be a useful thing to do. Linking specifically from a piece of text, using the hyperlink format, certainly has its place, but sometimes an item only generally applies to a large group of words. Fortunately Scrivener has a tool for this called References. 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 References inspector pane establishes a *general* link to them pertaining to the whole section of the draft—that document.

To create links, simply drag documents from the binder or the Finder, and URLs from your browser, into the table to create links. You can also use the + button in the References header bar.

Double-clicking on the icon of the reference will open the document, either inside Scrivener or in the default application or browser depending on the reference type. References will by default be opened in the opposing split, opening one if necessary, but you can opt to open a reference using an alternate method by right-clicking on it in the table. It is possible to open references in a:

- Current View, replacing the current editor
- Other View, as a split view using the last split type you used if one needs to be created

The + button has three options available:

- *Add Internal Reference*: presents a menu containing all of the items in your project binder. Click on any item, even a folder, to create a reference to it. The default description for it will be its binder title.
- *Look Up & Add External Reference*: use this menu to load a file browser. Any file you select in this browser will be added as a link to the reference list. The default description for it will be the file name.
- *Create External Reference*: creates a new row in which you can manually copy and paste or type in the internal title 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 local file URIs are also valid.

Any existing reference can be edited by double clicking on the text labels. To finish editing, press Return or click anywhere else in the table with your mouse. 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 reference and then remove the old one.

Clicking the header bar of the references area will bring up a pop-up menu from which you can select "Document References" or "Project References". As with notes, document references are attached to the current document whereas project notes are global and remain the same no matter which document you are viewing.

To delete a reference, select it and click the - button.

### 18.3.3 Keywords Pane

Keywords can be used to hold arbitrary information. A few examples might be character names, locations, interviewees, or workflow hints such as "Needs Illustrations". You could assign the name of a theme to a document as a keyword, and then later you could perform a keyword search to find all documents that deal with that theme (especially useful as you are unlikely ever to use the name of a theme directly in the text itself).

You can associate keywords with documents to help you find them when performing a project search. Use the plus and minus buttons in the header bar above the keywords table to add and remove keywords (or you can drag words and phrases from text to the keywords table to create a new keyword). You can also use the Project Keywords window, which stores a repository of all keywords used in the project, to assign keywords by dragging them from the project keywords into the keywords table in the inspector; this is an easy way to add several keywords at once.

There are several ways to add new keywords to your document:

1. Drag the keyword(s) from the Keywords panel to keywords table in the inspector. Useful for adding multiple keywords to an item at once.
2. Drag the keyword(s) from the Keywords panel to the name of an item in the binder, corkboard or outliner view. If multiple items are selected, then they will all have the keyword selection assigned to them.
3. Click the + button in the keywords table header bar to type in a new keyword
4. Using the gear menu in the keywords table header bar to access the **Add Keyword** menu, which will contain a sorted list of all keywords in the project Keywords panel.

Keywords can be unassigned from a document by selecting them in the keywords table and clicking the - button, or pressing the **Delete** key on your keyboard.

To change the order of the keywords, drag and drop them in this table. This can be useful if you are viewing keywords on the corkboard as colours. Since only the first few (the precise number of which you can modify) will be shown, it is good to move the most important keywords you intend to track to the top of the list.

See Also:

1. Using Keywords ([subsection 10.1.4](#))
2. Project Keywords ([section 10.2](#))

#### 18.3.4 Snapshots Pane

This section of the Inspector gives you easy access to all of your existing snapshots, the ability to create new ones, and delete old ones.

At the top of the Inspector, you will find two buttons. Clicking the + button will create a new untitled snapshot for the current document (you can also use **Ctrl-5** to do this while typing). To remove a snapshot, select it in the table, and click the - button. To read the contents of a snapshot, click on any of the rows and its text will appear in the text preview area below the table. It is not possible to edit snapshots, but you can select text from them and use copy and paste or drag and drop to restore bits of them.

The **Roll Back** button will revert the current editor text to the text in the selected snapshot. To use it, select the snapshot you wish to revert to, and click the **Roll Back** button. Scrivener will request confirmation from you, and give you the opportunity to snapshot the *current* text if you wish. Note this cannot be undone, but if you opt to create a snapshot from your current text, it is possible to effectively undo a roll-back by rolling “up” to the latest revision.

See also: Using Snapshots ([section 15.6](#)).

## 18.4 Locking the Inspector

You can lock the inspector to the current editor so that when the other editor in the split view receives the focus, the inspector does not change contents but instead continues to display information about the document in the first editor. To lock it, just click on the icon of the padlock in the bottom right of the inspector (in the footer view).

# Scriptwriting

Although Scrivener is not intended to be a dedicated scriptwriting program (for such a program you might want to try Final Draft or Celtx if you have not done so already), it does allow for basic script formatting and is thus great for first drafts.

## 19.1 Formatting a Script in Scrivener

To format a script in Scrivener, select the format you want to use from the **Format ▶ Scriptwriting ▶** sub-menu, or use choose one of the pre-built project templates in the “Scriptwriting” category. When in scriptwriting mode, the top item in the Scriptwriting sub-menu will be checked and will display the name of the format you are using. This preferred format can be toggled on individual documents with **Ctrl-4**.

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)
- Stage Play (UK)
- Stage Play (US)

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 drop-down 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 **Ctrl-\** at any time from within a scriptwriting document (even in full screen 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.

Upon hitting enter and typing, the paragraph formatting will be changed to that of the next set element. So, for instance, if you hit **Enter** 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. 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 **Format ▶ Convert ▶** sub-menu 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 19.4.1), though note you should in general not use this list to assign project specific completions like scene locations. It is best to use the **Project ▶ Auto-Complete List...** window for this. 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.

## 19.2 Printing or Exporting a Script

You can use the compiler (**File ▶ Compile...**) 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).

## 19.3 Working with Final Draft 8

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.

### 19.3.1 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...”
4. In the open panel that appears, select the FDX file that contains the elements and formatting you want to use and then click “OK” and “OK” again on the Script Settings panel to accept the changes.

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 scriptwriting format (**Format ► Scriptwriting ► Script Mode - FDX Script Settings**) 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.

### 19.3.2 Importing FDX Files

You can import FDX files into Scrivener using the standard file import methods; drag and drop into the project binder, or with **File ► Import ► Files...**

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.

### 19.3.3 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 8 (FDX)” as the export format.

The exported file should open in Final Draft 8 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. Go to **File ► Compile...**
2. Select “Final Draft 8 (FDX)” as the file format.
3. You may need to tweak the various settings of the Compile Draft sheet to get the script formatted correctly, just as you would with any Scrivener project.
4. Click on “Compile...” and choose a filename.
5. Open the exported file in Final Draft 8.

You should see 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.

## 19.4 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.

To create your own script format mode, select **Format ► Scriptwriting ► Script Settings...** and use the window that appears to name your format and define its various elements. At the top of the panel is the Format Title text field. Enter the name of your format here (e.g. “Television (BBC Comedy)”). This is the name which will be visible in the menu, if you later choose to save your settings to a script format that other projects can use as well.



On the left of the panel is the Elements list. This is a list of 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. Note that “General Text” will be added to the end of any element list you create but will not appear here as they have no special properties by definition.

**Script Formats are Static:** Creating a script format is a static process. That is to say that if you create a script format and then use it to write a script, if you then go back and edit the script format in the Script Settings panel, the elements of the script you created using the older script format will no longer be recognised as anything other than “General Text” (until you click into each element in your text and select the correct element from the script elements pop-up menu to update the text). Thus, to save yourself lots of painful reformatting later, it is strongly recommended that you ensure you get your script format right before you start using it widely.

Next to the “Manage...” menu ([subsection 19.4.2](#)) is another drop-down menu that allows you to choose the units used to define your style: inches, centimetres or points (most script formats use inches, of course).

On the bottom-right of the panel are the “OK” and “Cancel” buttons for saving or cancelling the changes you have made to the format mode respectively. Clicking **OK** saves the script format into the *current project package* only. This means that if you share your .scriv file with someone else, that someone else will be able to open the project and use the script format regardless of whether they have this script format installed on their machine. The format will not be available for use with other projects until you select “Save for use with other projects” from the “Manage...” menu. Once you are happy with the format, it is thus recommended that you make it available for use with other projects, as you might not remember which project file you saved it in.

It’s worth reiterating that script settings are saved directly into the project itself automatically, when you click the **OK** button. There is no need to repeatedly save a customised script. Saving a script is for when you want to use that format for another project entirely, or to a file for transport to other computers.

#### 19.4.1 Format Tabs

On the right of the panel is the tab view. This is used for setting the format for each element. Select the element from the list on the left, and then set the formatting for that element using the controls in the tab view on the right. The setting controls are grouped under four different tabs, which this section will describe.

## 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 for the current element (selected in the list on the left). Scrivener does not use the font to identify elements, so you can change the font in options without messing up Scrivener’s recognition of script elements. This also holds true for font changes made in the editor.

**Size** Allows you to set the font size in points.

**Style** Provides a number of options for determining the appearance of the current element, mostly self-explanatory.

- *All caps*: If this is checked, the element will be capitalised. If “End of line” is selected, the element will be capitalised all the way to the end of the element; for the purposes of this definition, a paragraph is a line. If “First tab” is selected, the element will be capitalised only up to the first tab character. 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(s)” is selected, you can enter the keyboard characters into the text field 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.
- *Underline*: allows you to set the underline style for the current element.

Colour and Adornment \* *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 inter-line spacing (single, 1.5 or double). Use the **Other Spacing...** option to also set pre-line spacing to buffer it away from the prior element. The read-out for this is “Line-spacing Before-spacing/After-spacing”. So “12 24.0/0.0” means 12pt line-spacing (or single, if the main font is 12pt) with 24pt before-line spacing, or two lines with a 12pt font. Spacing before can also be set independently, below.

**Ruler Conversions:** When working with units in Scrivener, keep 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 paper left edge, 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. For example, if the Scene Heading is specified to begin at 1.5”, 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.)

#### 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.

**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:” pop-up button to set which element will be formatted when you hit tab on an empty line. This allows you to cycle through elements 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. Select “Go to:” and an element from the pop-up button to have tab automatically take you to the next desired element (inserting a new line automatically). Use this to define convenient alternate common elements. Select “Insert” and type characters in the text field to have the tab key insert a character sequence. In 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.

### 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. When typing your script, hit **Alt=** to bring up the suggestion list beneath your cursor. Additionally, if you have your options set up to offer suggestion 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.

### 19.4.2 Managing Scripts

At the bottom-left of the main script settings panel is a drop-down menu entitled “Manage...”. This allows you to reset your script elements to the default screenplay settings, to save your format for use with other projects, to save it for back-up somewhere on your hard drive (or to share it with others) and to load from file an existing format mode created either by yourself or someone else.

**Reset to defaults** Reverts all customisations that you have made to the settings panel to the Screenplay default. If you wish to revert to a saved script setting, simply select that script from the Scriptwriting menu, instead of using this tool.

**Use current font & paragraph settings** Will attempt to import all available character and paragraph level formatting attributes 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. Simply click through the document, locating element types, and use this tool to import the correct formatting into each element type.

**Load from file...** Loads a Scrivener script format file from the disk. Use this if you have downloaded a script format from the Internet, or are in the process of transferring formats from one computer to another.

**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. **MAC:** This will not install the saved script into your Scrivener support folder. Use the below command for that.

**Save for use with other projects** Essentially accomplishes the above, but automatically installs it into your Scrivener support folder for immediate 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 as well as this one.

## 19.5 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.

For instance, you could use it to aid in the transcription of an interview. By setting up formatting “elements” for each participant in the conversation, you could easily attribute each portion of text to the appropriate speaker, and if the interview is a back and forth between two individuals, you could even set it up to alternate back and forth between speakers whenever you press return.

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.

# Writing Tools

## 20.1 Searching and Replacing

There are three primary tools for searching for items in a project. Project searching has numerous options for searching for particular types of information (like say, searching just in synopses), and will always scan the entire project for matches. Document searching is more familiar in that it provides a Search and Replace window that will act on the current text you have loaded. Third, format searching has many ways to search for particular types of formatting in your project.

Depending on the type of search being used, you will be faced with two different types of response from the application. Project searching *gathers* all documents that match the criteria together, and displays them 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 and format searching on the other hand do not collect results, but rather step through the search results one by one, in a familiar fashion. Document searches will never “escape” the text editor. That is, they will not go off into other documents you do not have loaded, looking for results. Format searching, as with document searching, operates in a step-by-step manner, but will by default step through every matching item in the entire project.

### 20.1.1 Document Find and Replace

As with many applications, you are provided with a standard Find and Replace panel which can be called up in any text view with **Ctrl-F**. The find panel operates only in the current text editor (which might include multiple documents when using Scrivenings mode). To replace text in all documents use Project Replace (see below); to search for text in all documents, use Project Search.

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.

- **Return**: is 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
- **F3**: 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)
- **Shift-F3**: 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.

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 **Next** and **Previous** buttons will jump from one result to the next. When the last match is discovered, the Find tool will wrap around to the beginning of the document again.

There are three replacement options.

- **Replace**: replaces the currently matched item
- **Replace & Find**: equivalent to clicking the **Replace** button and then the **Next** button
- **Replace All**: will replace all matches with the provided replacement value with no further interaction.

To leave the Find tool, you can either click the close window button, or tap the **Esc** key.



### 20.1.2 Project Search

When you use project search, the results will be placed into a temporary, dedicated collection called “Search Results” (section 8.4). If the collections interface has been hidden, you will only see the Search Results tab along with the Binder tab, but the collection will be created, and can be accessed later so long as you do not run another search.

To use project search, click in the search tool located by default on the right side of the main toolbar and type in the phrase you wish to search for.

**Search Results and the Sidebar:** Since search results temporarily replace the binder in the sidebar, it can be confusing for new users to figure out how to get back to the binder. When you are done, you can switch back clicking on the “Binder” tab, but otherwise you can use the small **X** button in the Search Results tab to dismiss the search results. Using the former method retains the search for future use. Clicking the close button on the Search Results tab will reset the project search tool. You can also erase the contents of the toolbar search to switch back to the binder.

By default searches will match all searchable elements in the project and be analysed with the exact phrase operator. The search scope, data type, and operator mode can be adjusted via the project search options menu, by clicking on the magnifying glass to the left of the search field. The options are as follows:

**Search In** Sets the data type of the search. You can set the search to only query a certain type of meta-data or text field, rather than everything.

- *All*: The default search mode. Every available type of searchable field will be queried for matches.
- *Title*: Only the titles of binder items will be searched for. This is similar to title filtering in the binder.
- *Text*: The text contents of files and folders will be queried. Note this does not include notes and synopsis.
- *Notes*: The auxiliary document notes will be searched. Note this does not include project notes.
- *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*: Only keywords will be searched. Note you can also perform keyword searches quickly by using the Keywords HUD.

- *Label*: The text (not colour) of the label meta-data will be searched. Note the name of this meta-data field can be changed per project.
- *Status*: The text of the status meta-data will be searched. Note the name of this meta-data field can be changed per project.

**Operator** Set the method by which search terms will be handled as you type them in.

- *Exact Phrase*: The default method. What is typed 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”. It will also return documents that contain “the books”. For exclusive matching, use Whole Word (below).
- *All Words*: Every word entered into the search field must appear in the selected data type. Documents which only match some of the words will not be returned. Words can be entered in any order. Analogous to logical AND.
- *Any Word*: Queried documents must contain at least one of the words typed into the search field. Analogous to logical OR.
- *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 “Joseph”.

**Options** Provides a few extra options, as well as searching scope limiters. Scope limiters instruct the search engine to only analyse parts of your binder, rather than the entire thing.

- *Search Draft Only*: Will only look in the Draft folder of the binder. Note that if the name of Draft has been changed in the project, the title of this option will reflect that name change.
- *Search “Included” Documents*: Include documents that have been marked as “Include in Draft” in their meta-data. This option can be combined with the below.
- *Search “Excluded” Documents*: Include documents that have *not* been marked as “Include in Draft” in their meta-data. 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.

**Save Search...** Creates special Saved Search Result Collections ([subsection 8.4.4](#)), which will keep themselves dynamically updated whenever you view them.

### 20.1.3 Project Replace

The menu command, **Edit ► Find ► Project Replace...** provides the ability to replace text throughout the entire project. Set the text to find, the text with which to replace it, whether the search should be case-sensitive or apply to whole words only and specify what should be affected by the replace (any combination of text, titles, notes, synopses and snapshots). Note that a Project Replace cannot be undone (except by using “swap” to exchange the search term for the replacement text and repeating the Replace All operation, which in some cases might have unexpected results). 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.

As with all tools that make widespread changes in a fashion which cannot be undone, it is often a good idea to back up your project prior to use it. A simple mistake can render your entire draft illegible, or even worse, produce subtle flaws that even your editor ends up missing. Using the “Whole words only” option can mitigate this a great deal. Without that option, 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”.

You can choose the scope to optionally affect a variety of document meta-data, as well as just the text of a document. The following options are available:

1. Text: standard text editor contents (including any linked footnotes and comments from within that text)
2. Titles: the titles of the documents as they appear in the binder
3. Notes: any inspector note fields that are attached to documents
4. Synopses: the text content of each index card
5. Project notes: any project notes will be impacted as well.

### 20.1.4 Find by Format and Text Function

The Formatting Finder palette, **Edit ► Find ► Find by Formatting (Ctrl-F3)**, gathers together a number of powerful project-wide search tools for otherwise difficult to locate things, such as italicised text; cross-links; annotations & footnotes; and so on. The basic panel has a drop-down menu at the top which is the primary formatting mode selector. Below that there are two common tools to all modes.

**Containing text** Limits the search to only return results in which the specified text falls within the formatted range.

**Search in** There are two choices available: “All Documents” will search all text documents in the binder (even outside of the Draft); “Selected Documents” will constrict the search to only those items you have currently selected in the sidebar or a view.

If you leave all of the criteria types empty, it is possible to use this panel to find *all* instances of the particular formatting type.

The rest of the panel will change depending upon the current formatting 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, allowing you to 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 Find panel it can run in “background mode” as well. Once a search criteria has been set up, you can close the window and use menu commands or keyboard shortcuts for stepping back and forth from one match to the next, in the **Edit ► Find ►** sub-menu.

The rest of this section will go over each of the individual formatting search types.

### Highlighted Text

Looks for text that has been highlighted using the highlight feature ([section 17.4](#)).

**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.

Click on the colour chip to display a list of the built-in highlight colours, or use the “More...” entry to search for a custom colour.

### Comments and Footnotes

Will search within any inspector-based comments or footnotes in the project. By default all will be considered as potential matches, but you can narrow this down by selecting one of Comments or Footnotes from the “Type” menu.

### Inline Annotations and Footnotes

Tools for stepping through all inline annotations or footnotes in the project. This search tool only has the two standard filter criteria available to it.

### Coloured 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. Do note the colour choice still needs to be precise, so using custom swatches or built-in presets will generally be easiest. As with the highlight tool, you can click on the colour swatch to select from a built-in default.

### Links

You may search for URL or internal Scrivener Links 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” menu to make this choice.

## 20.2 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:

- To get word, character and page counts for the whole of the draft (that is, the contents of the Draft folder) or for any documents selected in the binder, use **Project ► Project Statistics**. To calculate the printed page count statistics, Scrivener internally generates a compile document, based upon your compile settings—so you need to note that if you have compile set up to print off synopses and nothing else, only the synopses will be counted. This can also mean that there will be a delay in displaying this panel with larger projects, and with projects over 100,000 words, you’ll need to manually update the printed page count, to avoid lengthy waits whenever the tool is used.
- To get word, character, paragraph and line counts for the current document, and a breakdown of word frequency (the number of times you use different words), select the **Project ► Text Statistics** menu command (this will only be available if a text document has the keyboard focus). When using Scrivenings mode, the full word count will be tabulated, not just the section you are currently editing.
- To set a target word or character count and track your progress for the entire draft or the current writing session, select **Project ► Project Targets**.
- 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 chapter folder, tracking the latter folder as what you want to have a goal on, 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 since it 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 goals. Targets for folders can be easily set by using the Target column.
- To get the word or character count for a selection of text, select some text in a document and right-click on the selected text. The word and character count will appear in the bottom of the menu in grey text.
- 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 down-arrow button in the outliner column bar (or by selecting **View ► Outliner Columns**). 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 switch to Scrivenings view to get the total word count in the footer bar. Note this method only counts items in the session, so if the selection you made does not include a container’s children, you’ll need to make sure to select them as well in the binder.

**Usage Tip:** The word frequency tool in Text Statistics only counts text within a current view, which can include a Scrivenings session. Thus, to perform this analysis on the entire Draft or large portions of it, create a Scrivenings session first, and then use the **Project ► Text Statistics** tool.

### 20.2.1 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. To bring it up, select **Project ► Show Project Targets** or press **Ctrl-**, to toggle its display.

The targets panel displays two progress bars: one showing the progress of the draft and the other showing the progress of the current session, being defined as the period between the project being opened and closed again. Using this panel, you can set a target word count for the draft (that is, the contents of the Draft folder) and for the current session, and keep track of how far you have to go before you reach your goals. To change the

targets, just click in the appropriate text field and enter a new target word count (or target character count—you can set whether the target should be in words or characters by clicking on “words” so that a pop-up menu appears, providing a choice between “words” and “characters”).

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 Text Goals (subsection 14.2.3) tool in the footer bar.

### 20.2.2 Project Statistics

Project statistics can be called up at any time with **Project ► Project Statistics** or **Ctrl-** on the keyboard. This sheet has two sections to it:

- **Draft:** everything included in the Draft folder. Note that compile options which modify export content in any way can impact this counter. If your compile is set up to only export titles in an outline format, you’ll get a pretty small word count. In general, consider what your document looks like when you compile: that is what will be counted.
- **Selection:** a count of everything you have currently selected in the Binder, including all of their children items as well (which can be optionally disabled).

Each of these sections have identical statistics available to them. The word and character counters are self-explanatory. The two page count estimates are based on different algorithms.

*Pages by paperback* uses 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  $\frac{720000}{250 \cdot 6} = 480$ . If you need to adjust the average number of characters per word, switch the counting mode to Characters and provide the product of the words-per-page multiplied by average word length. For example, if the average word length is 8 and the number of words per page is 200, you would enter 1600 into this box to produce a custom result.

*Pages printed* will be more accurate for your own printouts, 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. Consequently, in large projects it may take a few moments for this window to display.

### Project Statistics Options

To access options for how project statistics are calculated, click the **Options** tab in the window.

**Selection Statistics Options** Tuning options for how selected items should be counted in statistics.

- *Count all documents*: In this context, all documents means: regardless of whether the document meta-data flag, “Include in Compile” is on or off.
- *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 annotations*: All annotations will be disregarded by default. If you wish to count them too, un-check this option.
- *Exclude footnotes*: Footnotes are by default included in the count.
- *Count sub-documents*: When disabled, only the literal selection will be counted. With this option on, the selected items *and* all of their children will be counted, all the way down to the bottom of the outline.

**Paperback Page Count Options** Set the counting algorithm used to estimate paperback page counting. This figure will be divided from the total word count to produce an estimate. Read the prior section for details on how this works.

### 20.2.3 Text Statistics

The Text Statistics panel can be accessed with **Project ► Text Statistics** or **Ctrl-.** It is only available when the active editor is displaying text, and it will produce statistics for *all* of the text being displayed, even if that editor is currently displaying many documents in a Scrivenings session. The types of information it provides are more pertinent to the topology of the text document, as opposed to the end product (where page counts and paperback estimations come into play).

In addition to the simple counter statistics, you can flip down the Word Frequency chart to display a table of discovered words, how often they have been used numerically, and graphically how often in comparison with other words in the table. These columns can be sorted by clicking on their header fields.



## 20.3 The Name Generator

Scrivener comes with an exhaustive name generator, which includes many thousands of common names in several languages. To access the name generator, invoke the **Tools ► Writing Tools ► Name Generator...** menu command. You can adjust how many names will be generated, and then click the **Generate Names** button.

The available options are:

- *Gender*: select one option here. The default is either gender
- *First & Last Name Origin*: for each name you can select to pull from one particular region, language, or randomly select from the entire database
- *Number of names to generate*: controls how many names are produced in the list to the left
- *First & Last Name constraints*: These two sets of controls will enact constraints on the types of names that will be generated. Each has the following options:
  - Start letter with...
  - End letter with...
  - End and start with...

From the second drop-down menu, you can either select a letter (Roman alphabet only) or set it to use any letter at the top

- *Number last names*: produces names like, “Otis Cowie-Milburn” when set to two, by example
- *Display*: switch between showing full names, and first or last names alone.

The list will update as you make changes to the options. You can click the **Generate Names** button at any time to see an entirely new list of names. The **Reset all to Defaults** button will put all options above back to default.

To select names you like, click on them in the list and then click the **Add Selected to Shortlist** button, below the name list. This will copy the name to the shortlist at the bottom of the window. Once you get a list of names you wish to save, click the **Copy Shortlist to Clipboard** button and it will be safe to close the window and paste the names wherever you like.

Closing the name generator window will erase the shortlist, so if you intend to keep it, be sure to copy the list to the clipboard before you go.

### 20.3.1 Name Meanings

Along the top of the window, you will find two tabs, one labelled “First Name Meanings”. Click on this tab to reveal a spreadsheet with all of the names in the database, organised by gender, origin, name, and meaning. You can narrow down the list using several optional filters at the top of the window, and even search for letters or words in either the meaning or the original name.

As with the main name generator interface, you can copy rows out of the spreadsheet with the **Add Selected to Shortlist** button. This will copy all of the data in the row for your future reference. The shortlist is shared with the name generation tab, so you can mix meanings with full names that you’ve generated.

### 20.3.2 Adding Your Own Name Lists

You can add your own custom name lists to the generator. Name lists should be formatted so that each name is on its own line, with the origin of the name after it, separated by a comma:

```
Parker,English  
Paino,Italian
```

It might be easiest to produce these lists in a spreadsheet and export as a CSV file.

Click on the “Import” tab in the name generator window, and click on the **Browse...** button for first or last name. Locate the file in the chooser dialogue and click **Open**, then the **Import File** button beside it. You will be presented with report of how many names were imported into the database.

If you wish to reset the name generator to default, use the “Restore Defaults” tab in the name generator window and click the **Restore Defaults** button. This will return the name lists to their factory default condition. You will need to re-import your revised name lists at this point, if desired.

## 20.4 Bibliography Management

Scrivener offers simple integration with your favourite bibliography or citation manager (such as EndNote) for academic work. To set it up, visit the General options tab ([subsection B.1.3](#)). Click on **Choose** to choose your citation manager.

Once the citation manager has been set up, you can hit **Ctrl-G**, **Ctrl-O** (or **Format ► Bibliography \ ► Citations...**) while working in the editor, 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 your manager to scan these placeholders into final print form. Not every program or service provides RTF scanning, however. You should research the software you intend to use and make certain it designed to work with programs other than Microsoft Word.

## 20.5 Using Equations with MathType

If part of your writing involves the addition or construction of equations in your draft, then you can use Scrivener's [Design Science MathType](#)<sup>1</sup> integration to insert 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 **Edit ► Insert ► MathType Equation ►** . If you have MathType correctly installed on your machine, you will see the equation interface 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**.

To edit an equation later on, simply right-click on the equation in the Scrivener editor and choose **Edit MathType...** from the top of the contextual menu. The MathType interface will pop up again, and any changes you make will be saved back into the file when you close it.

Equations act much like ordinary images. They can be aligned or styled 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.

Equations can be saved and edited from any rich text field, including project or document notes.

## 20.6 Using Equations with MathML

If you do not own MathType, or your system cannot run it, an alternative is to use the MathML equation language. Inserting a MathML equation is done via **Edit ► Insert ► MathML Equation....**

The equation can be named for future reference. The name will be used when the equation needs to be exported as a file (such as when compiling to HTML).

As you paste or type MathML code into the main editor, a preview will be generated below in the grey area. There are three buttons which can adjust the display. The **+** and **-** buttons are used to zoom the size of the rendered equation up or down. The **Grid** button

---

<sup>1</sup><http://www.dessci.com/en/products/mathtype/>

enables the formatting grid aid. The spacing metrics for each element will be drawn as well as the elements.

The final size of the window determines the size of the graphic, so be sure to shrink the window to the size of the equation, prior to closing it.

# Using MultiMarkdown

For those who prefer structural writing solutions to rich text (and if this means nothing to you, you can entirely skip this section), Scrivener allows you to import and export using Fletcher T. Penney’s MultiMarkdown syntax. MultiMarkdown makes it easy to generate documents in any number of formats—for instance, L<sup>A</sup>T<sub>E</sub>X and HTML5 - using a basic markup syntax (for instance, using asterisks to define **bold** and *italics*).

It is important to note that if you have used normal word processors (such as Microsoft Word or OpenOffice.org) and rich text for years and have never heard of MultiMarkdown, structural writing or such formats as L<sup>A</sup>T<sub>E</sub>X, you can ignore this aspect of Scrivener. Scrivener was designed primarily as a rich text application; MultiMarkdown is implemented in such a way that those who want it can access it easily but those who have no need of it need never be bothered by it. If you are curious about the process and would like to consider adapting it, you should read *What is MultiMarkdown?* (section 21.2) for an overview of its philosophy, limitations and capabilities.

## 21.1 What is Markdown?

The MultiMarkdown syntax is based on the “Markdown” syntax created by John Gruber, and the best description of what Markdown is comes from his [Markdown web site](#), [Daring Fireball](#)<sup>1</sup>:

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

---

<sup>1</sup><http://daringfireball.net/projects/markdown/>

should be publishable as-is, as plain text, without looking like it has 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.

## 21.2 What is MultiMarkdown?

For comprehensive information about [MultiMarkdown](#) and [how to use it](#), see [Fletcher T. Penney's web site](#)<sup>2</sup>. The purpose of MultiMarkdown is to extend Markdown's goal (which is primarily toward web publishing) into the realms of traditional publishing, while still retaining a solid platform for web publishing, and adhering to the technical and aesthetic principles of Markdown. Toward this goal, it provides output formats for OpenOffice's ODT format<sup>3</sup> and the L<sup>A</sup>T<sub>E</sub>X typesetting engine.

To augment the expansion into document formats, additional features have been added which will be of interest to authors, such as footnotes, tables, and cross-referencing.

The basics aside, you might be wondering if any of this is relevant to you. With the exception of generating L<sup>A</sup>T<sub>E</sub>X documents, it should be made clear that there is nothing you can do with MultiMarkdown that you cannot do using Scrivener like a typical word processor. That's a very broad statement, specifically there are many differences, but most of these revolve around how you work rather than what your end product of that work will be. We'll get to the specifics shortly, but for now here is a list of differences in the end product that you can expect:

- MultiMarkdown (hereafter shortened to MMD) can produce *styled* word processor documents, when using ODT. That means you document titles will be classed according to their depth in the outline, block quotes will be assigned to the appropriate style, and so forth. If you require a stylesheet-ready word processor document at the conclusion of the writing process, MMD is something to consider.
- As already stated, MMD supports easy to write L<sup>A</sup>T<sub>E</sub>X document generation. Without having to bother with the nitty gritty codes, you can produce clean, attractive documents using a typesetting system that has been in refinement for nearly thirty years. How good are these results? Well you're look at an example right now. The Scrivener user manual is typeset using this system. If you require L<sup>A</sup>T<sub>E</sub>X documents, but would rather not *compose* L<sup>A</sup>T<sub>E</sub>X documents, MMD might be a good choice for you.

---

<sup>2</sup><http://fletcherpenney.net/multimarkdown/>

<sup>3</sup>via the Flat XML format (.fodt) which is currently best supported by the OpenOffice version, LibreOffice.

- For web publishing of all kinds, MMD outputs clean and semantically valid HTML5. What does that mean? It means you titles will be defined into outline logical web header depths, you quotes will be blockquotes, your tables will be clean, and most importantly the output will be easy to apply CSS stylesheets to. This will produce a result you can just drop into your blogging software without any stripping of codes or fiddling with styles.
- MMD currently has no easy path toward e-book publishing. If you intend to put your work out primarily in an e-book format, right now it's probably not the best choice for you. It was designed to facilitate document generation, and e-book formats are typically composed of many small documents instead.

Now for the specifics, or what we might call the 'workflow'. One of the thing that attracts people to MMD is that it focusses solely upon the text. All formatting is done with easy-to-read text codes. For the most part, these are also very easy to type in. You don't have to memorise keyboard shortcuts for formatting commands because they all use common punctuation marks.

On that note, you're going to have a stare at a little more punctuation than you might be used to. This isn't as bad as writing in HTML, not by a long shot, but there is very little rich text to MMD conversion in Scrivener. You won't be looking at normal italicised fonts and such. Some people welcome that, others are glad that 1995 happened and never want to see codes again. If you'd like to see just how much punctuation we're talking about, you can find the Scrivener user manual project [on our support page](#)<sup>4</sup>. Download that and browse through the project. Keep in mind that, as a technical document, it has a more dense formatting requirement than a novel or non-fiction document might. Even so, you should find that it doesn't look all that much different from a standard document.

This chapter will not attempt to teach you MultiMarkdown. There are excellent resources available on the Internet for this, including the link at the top of this section. If you already know Markdown, then you already pretty much know how to write in MultiMarkdown, especially with Scrivener, which handles some of the MMD extensions for you. It will instead cover the integration features provided by Scrivener, and assumes a basic working knowledge of the system.

## 21.3 Getting Started with MMD

There is very little you need to do to prepare for using MultiMarkdown. Scrivener comes pre-loaded with the latest version of it (at the time of its release), and also includes several optional packages which will be employed as needed. This means that even if you

---

<sup>4</sup><http://www.literatureandlatte.com/support.php>

intend to use a more advanced workflow, like  $\text{\LaTeX}$  or XSLT post-processing, 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. You could right now start testing it out and compiling finished documents.

Mac: If you've already got a Markdown or MultiMarkdown document in a partial state of composition, you might like to know you can import it into Scrivener. If the document does not have titles in it, then this method will be of no use to you, you might as well just drag and drop the file straight into the binder to import it. If it does have titles, then use the **File ▶ Import ▶ MultiMarkdown File...** menu command. This will read in the title structure and convert it to a Scrivener outline, parcelling out all of the text in between titles into the appropriate sections of the binder. As you can see, this makes for a good way of “round-tripping” a document if you need to pass it along to someone else for editing, or wish to edit it yourself in another piece of software.

## 21.4 MMD and Scrivener

Because MultiMarkdown requires a certain degree of textual precision, many of the visual tools that rich text users commonly use can have a negative impact on how well you can “see” your document. For example, in MMD, a tab in front of a line turns it into a code block; thus indenting the first line of each paragraph could lead to confusing results. Likewise, using paragraph spacing could make for confusing results in an environment where every block needs a clear space around it. Pseudo-spacing might lead you to believe a space exists where in reality there is none. Consequently it is often a good idea to reduce the level of formatting in your projects as much as possible.

This doesn't mean that a project needs to look unappealing. If you reduce the paragraph spacing to zero, the double-newline that is required between paragraphs looks only slightly larger than normal. Meanwhile, because formatting is ignored by the compile process, you can feel free to take advantage of any formatting tools you like to pretty up your document, safe with the knowledge that none of it will ever appear outside of Scrivener. Some like to use a monospace font, like in many plain-text editors. Indeed, with tables, this can make editing much easier. However many others use whatever font is appealing to them, maybe saving monospaced fonts for those sections that require a little perfect placement.

What about special characters? This is often a matter of taste and workflow, but you should be aware that many of the default outputs do not handle Unicode well, and it is often better to either use codes for special characters, or let MMD handle the more common ones, such as typographic quotation marks. You will often find that MMD works



in a more error-free fashion if you disable all of Scrivener's Corrections (section B.6) preferences. You won't need to worry about which way your quotation marks are pointing, just leave them straight and let MMD handle the typesetting details for you.

There is one exception to that. An option in the compiler can be set to attempt to convert rich text formatting to real plain-text whitespace. That is, a paragraph with a full line of pseudo-spacing after it will be converted to an actual empty newline. Indents will become tabs (so be careful with that one). This option is mainly intended to be a convenience for people who are switching to MMD from a rich text environment, and have a lot of existing material that is single-line spaced. However you could use this option to take advantage of rich text aesthetics while working in MMD, if you wished.

When exporting via MultiMarkdown using any of the above methods, all rich text formatting is completely cleaned from your draft, as it passes through a plain-text engine; there is no need to clean up extraneous highlighting and other markings when working with MMD. There are three exceptions:

1. Any inline or linked images will be converted to MMD syntax, and exported into the compile folder, ready to use by the HTML, Markdown plain, and L<sup>A</sup>T<sub>E</sub>X formats. Any embedded images will be converted to MMD syntax and exported into the compile folder.
2. Images that have been linked to, using Scrivener Links, will have the linked text replaced with the image name, and the image itself exported to the compile folder. Thus you can type custom image syntax into the document, and still benefit from automatic resource gathering.
3. Scrivener's notation features (both inline and linked) can be used to generate MMD footnotes, and optionally its annotations and comments can be inserted into the document as well. Inline annotations on their own line (not embedded in another paragraph in any way) will be exported as DIVs, all other annotations, and all comments, will be exported as SPANs, to preserve the original paragraph flow around the comment.
4. Bold and italics cannot be compiled, but if you prefer to write using them, they can later be converted to MMD asterisks with the **Format ► Convert ► Bold and Italics to MultiMarkdown Syntax** menu command. Since rich text formatting can be less precise, you should verify that the syntax produced by this command is valid.

Some of Scrivener's tools which are intended for a rich text workflow have alternate purposes with MultiMarkdown.

**Document Titles in the Compiler** Document titles will be converted to MMD syntax.

While this is rather a function of the compiler (which may be set to not include titles) rather than the editor, it is important to know that your outline structure can still be seamlessly impressed upon the final work as output structure.

When titles are set use run-in formatting so that they appear in the same line as the first paragraph of the text, hashmark generation will be disabled.

**Compile As-Is** This checkbox in the inspector will protect an entire document from formatting alterations. In MMD, all this checkbox does is restrict the document from emitting a title header if it otherwise would have. Thus it can be useful for injecting sections of pure  $\text{\LaTeX}$ , use custom header depths that otherwise violate the outline’s literal depth, and so on.

### 21.4.1 Images

As mentioned before briefly, images can be dropped into the editor where you would like them to appear<sup>5</sup>, just as you would do so in a rich text work flow.

‘Also as with rich text, if you resize the graphic in the editor, instructions will be sent to the display formats to size the image in accordance with your specifications. This uses a soft sizing technique which merely resizes the graphic in real-time using the display or typesetting image, it does not harm the original pixels.

There are cases where you might not want the image to be in the editor. The image might be too large to comfortably work around, being print resolution, or you might find that the inclusion of many graphics in your files slows Scrivener’s saving routine down. While you could use the image placeholder tag that is available to rich text authors, you might as well use MultiMarkdown’s image syntax for its expanded feature set. To do so, simply write the image syntax as you normally would, and where you need the image’s path to be printed, create a Scrivener Link pointing to that image in the binder. It is important to note that any text in the hyperlink 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 avoid linking to graphics in such a way that the hyperlink extends beyond the area of the syntax where the path name should be printed.

---

<sup>5</sup>Though do note that the  $\text{\LaTeX}$  typesetting engine rarely takes such placements as literal. It will place the image (and tables for that matter) as close to the point you specify as possible, while maintaining a aesthetic appeal to the text blocks around it. For instance, if the placement of an image would push three words of the last paragraph onto the next page, it would prefer to keep the paragraph intact, and move the image to the next page instead.

### Cross-referencing Images

The manner in which Scrivener generates image syntax is conducive to cross-referencing to them. It will use the image filename for the ID. All figures with a generated ID will be placed in a list at the bottom of the compiled document for easy reference, along with any footnotes. Images that do not require an ID (such as inline paragraph images) will not be handled this way and cannot be linked to<sup>6</sup>. For images to have a generated ID, they must be placed in their own paragraph with no other text (excluding captions, described in the next section).

In 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 21.5.1](#)) and generate syntax where the position was located, like so:

```
! [] [analysis_of_derivatives-2008.png]
```

Then placing a the data for the graphic at the bottom of the compiled document:

```
[analysis_of_derivatives-2008.png]: analysis_of_derivatives-2008.png
```

You can thus refer to this figure in another portion of the document with the standard MMD syntax, ([see chart](#))(#analysis\_of\_derivatives-2008.png).

### 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-double-quotes on the same line after the image, separating the image from the quote with a space
- Using the same method as above, replace the double-quotes with matched square brackets
- Or you can use square brackets on the line *directly* following the graphic.

In all cases, there must be one whitespace character between the image and the caption syntax (either a single space or a single carriage-return), and there can be no text following the second quote or close bracket, otherwise it will be interpreted as a normal paragraph and the image will be exported using non-referenced syntax.

---

<sup>6</sup>However if an inline images has been resized, it will used the referenced syntax to apply the dimensions code to the image.

Captions will be exported to all formats which have a way of recognising them. If the format does not recognise captions, then nothing will be shown. ODT, HTML, and  $\LaTeX$  all support captioning. HTML5's `figcaption` syntax may not be visible in all browsers unless it is specially handled by CSS.

### 21.4.2 Footnotes

You needn't worry about footnote syntax and identification at all, when using Scrivener—though if you wish to, there is nothing stopping you from typing in the codes yourself. You can simply use Scrivener's tools for marking text as footnotes. Whether they will ultimately become footnotes or endnotes is not dependent upon anything Scrivener can do, but upon the export format's limitations or options. Under the hood, Scrivener will automatically number your footnotes as they occur in the document, and generate the proper MMD syntax for them.

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.

### 21.4.3 Annotations and Comments

For most uses, inline annotations 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.

Since MMD itself does not have any syntax for commenting a work (unless of course you consider HTML comments, which reduces the flexibility of your document as these are otherwise used to pass raw  $\LaTeX$  code to the final output), Scrivener provides two choices to you for handling them:

1. Inline annotations can, as with the other plain-text output formats, be wrapped in enclosing markers (square brackets by default). Creative use of this can allow you to use the inline annotations as shorthand for a longer sequence of coding you typically use—for example you could use them to supply HTML comment markers around the annotated text, thus providing an easy to read way of pushing raw  $\LaTeX$  code into the final compile.
2. Inline annotations can optionally be compiled using HTML syntax, instead.

When converted to HTML, inline footnotes which are in one way or another attached to another paragraph will be converted to `<span>` elements, which have been classed as “annotation”, and coloured with inline CSS in accordance with their appearance in the editor or sidebar. If however an inline footnote falls on its own paragraph, it will

use a `<div>` element instead. While simple MMD syntax can be applied to a span range (such as italicised text), a div block can contain complex MMD syntax, and thus an inline footnote can feasibly support all of the things that normal text in the editor could, such as images, lists, tables and so forth. The only thing you cannot do within such a block of text is footnotes, since footnotes and annotations cannot overlap in Scrivener. When a span style annotation has paragraph breaks within it, Scrivener will inject `<br/>` elements to emulate paragraphs without causing syntax errors. All of this means that the original flow of the document, with or without notation, will be preserved.

**Using Annotations as Functional Ranges:** Furthermore, and especially in conjunction with post-processing systems like XSL, the usage of HTML elements for notation allows for some creative post-processing uses, effectively allowing you to expand the syntax of MMD as needed, or even just as a way to better style your document with CSS. In fact the Scrivener user manual that you are reading right now makes heavy use of this technique. Whenever you see a tip box, such as this one, you are looking at what started out as an inline annotation in Scrivener. Ranges of text which pertain to only one platform or another (such as keyboard shortcuts) are blocked out with annotations as well, so that only one type appears. There are numerous strategies for doing so, but it is not within the remit of this manual to get into detailed examples of post-processing. Visit the MMD section of our [user forums](#)<sup>7</sup> for archived discussions on the topic, or start one of your own.

## 21.5 Compiling

This section of the documentation will not cover compilation in detail. You should consult the chapter on Compiling the Draft ([chapter 22](#)) 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.

### 21.5.1 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.

You may find that you want to keep this output folder “static” instead of having Scrivener completely regenerate it each time you compile. You might wish to assemble

CSS files, custom .tex files, styles, and other bits of supporting data 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 for the most part handle this transparently for you with natural file extensions. If the compile folder, for example, is called “my\_novel.tex”, then when you compile to that folder Scrivener will leave what is already there alone, unless it is a file that Scrivener is scheduled to produce. This way graphics will be updated, but any logos or other elements from the title page that are not in the project itself will be retained. Extension detection is done for “.tex”, “.md”, “.mmd”, “.fodt” and “.html”. If the file extension you wish to use is not one of those, you can instead rename the compile folder to “my\_novel\_mmd” and Scrivener will leave it alone the next time around. Any folder ending in “\_mmd” or “-mmd” will be preserved.

### 21.5.2 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 MultiMarkdown format makes for an excellent long-term archival document. It’s compact, easy to read, and easy to transform into a variety of output formats. It’s also very portable, being plain-text. While dropping an archived Scrivener project that you are done with into a database system or even just in your file system somewhere is something you should always do, 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 MMD, when generated a direct formatted output, and so is a critical troubleshooting step when faced with difficulties.

It is also, as noted above, a necessary format if you intend to use the MultiMarkdown version 2 system, as there is no longer an automated way of doing so in Scrivener.

### 21.5.3 LaTeX

If you’ve used MMD in the past, you’ll have probably noticed that how  $\LaTeX$  documents are set up in MMD3 is quite a bit different than in the past. In the past you could just tell a minimally annotated document to be a .tex file and it would do so. If you didn’t like the default Memoir class, you could throw in one line of meta-data to change it. This system, while easy to use from the document end, was rife with difficulties if you ever wished to slightly tweak the .tex output. You had to learn at least a little of the complex XSL stylesheet language to make your own class definitions.

The new system alleviates much of these problems by using .tex boilerplate files that you can easily edit in any plain-text or TeX editor. These are then imported into the final .tex file with the `\input{}` command, leaving the base document clean looking. This has however come at the expense of a slightly more involved meta-data block, and an awareness of how to install custom .tex include files into your system. We have implemented a system to reduce the necessary “plumbing” required to set up and switch between L<sup>A</sup>T<sub>E</sub>X stylesheets, as well as handling the production of gathering of any necessary files for one-click typesetting. In fact, you may not have to touch a single setting to get a ready-to-typeset document.

**Been Using MMD3 for Some Time?** When you select the “MultiMarkdown -> LaTeX” compile method, a “LaTeX” compile option pane will be added to the list. By default, your document will be compiled to the Memoir (book) class as it always has in Scrivener. If you wish to change that, or already have your project set up to define the class, you’ll want to visit this pane and change the provided setting. If you don’t do that, then end result could have doubled-up meta-data keys.

The automatic class selection system will insert the necessary meta-data keys where they should go, amongst any you have declared, and if necessary it will produce for you any of the boilerplate .tex files these classes depend upon. This will all be done during the compile phase, so the meta-data keys being used will not be displayed in the Meta-Data options pane. If you already have these files installed in your L<sup>A</sup>T<sub>E</sub>X distribution, then they will not be produced into the compile folder. If you would like to install these, you will need to download them from the [MMD LaTeX support file repository](#)<sup>8</sup>, and install them wherever your distribution handles user texmf files.

You can mingle “LaTeX Input” keys in your own meta-data, if they are meant to be used in conjunction with the defaults, so long as you place them so that they appear in the appropriate position upon final output. An example might be a custom chapter title typesetting rewrite that you’d prefer to use instead of the stock default.

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) using the “LaTeX Options” compile pane, and selecting “Custom”. Three text fields will be provided, where you can supply the initial preamble, and further preamble that is dependent upon any meta-data values (such as a title page) and beginning the document, and finally a footer placed at the bottom of the document. They will be exported into the compile folder and linked up in the generated meta-data for you. So long as the syntax is correct, your document should be ready to typeset moments later.

---

<sup>8</sup><https://github.com/fletcher/peg-multimarkdown-latex-support>

#### 21.5.4 HTML

Produces a clean, semantic HTML document. In older versions of MMD, this used to be the base format upon which all other formats were derived (using post-processing scripts). It is now simply an parallel format on equal footing with the rest. It is however still used as a base format if you invoke the XSLT post-processing switch in the Compatibility option pane. The default output will be basic HTML. If you wish to supply CSS to the document, you can do so with the “CSS” meta-data key (specifying a file or URL), or type one in manually into the “HTML Header” key, which can also be used to add Javascript and other functionality to the document.

#### 21.5.5 ODF (Flat ODT)

If you need your work to enter a standard word processor workflow, this will be your go-to method. MultiMarkdown’s ODF support is comprehensive, supporting most data types, and will utilise stylesheets, meaning the document can be easily adjusted aesthetically, and use the table of contents feature of a word processor. It represents the only output method in Scrivener that offers this support, making MMD an enticing option for those who do not mind learning how to use it, if they need stylesheet driven word processor documents.

The OpenDocument Format is an open specification office format, and specifically the ODT format within this specification is for word processing. OpenOffice (and the many forks off of it) is one of the few word processors that fully addresses the format. MultiMarkdown uses the *flat XML* variant of ODT, which has the extension of .fodt. At the time of this writing, LibreOffice (an OpenOffice fork) offers the best support for .fodt files, though OpenOffice proper can be updated to handle it as well—in fact LibreOffice just has this update preconfigured. Once the file has been loaded in LibreOffice, it can be saved out to many other word processor formats, a task you will almost always need to do if working with others.

When exporting to .fodt, you should use the UTF-8 text encoding.

#### 21.5.6 OPML (Outline Processor Markup Language)

The Outliner Processor Markup Language is a common standard format for transferring outline hierarchy between programs. The base format itself only supplies a specification for transferring headers, but many programs also conform to a non-standard convention of using a notes field to transfer content as well. MMD supports the latter ability, making it possible to transfer simple outlines with named headings and text content between supporting software.



### 21.5.7 PDF (via LaTeX)

The PDF output choice utilises the  $\text{\LaTeX}$  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 (three times, to get all cross-references properly registered). Consequently, if the meta-data calls for “snippet” mode instead of full document mode, are fatal errors in the typesetting process or if it requires additional steps to be completely typeset (glossaries and bibliographies foremost), it will not produce the desired result, if any.

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” option in the Compatibility option pane. 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  $\text{\LaTeX}$  distribution with *pdflatex* installed on your computer in an executable path, then this option will not appear in the compile format drop-down.

## 21.6 MMD Meta-Data

MultiMarkdown supports a free-form meta-data system, with some meta-data keys being used intelligently by the system. You should read the MMD documentation about these fields, and you can add your own without detrimental effect. To set up project-level meta-data, which will be attached to every compile you produce, use the Meta-data pane in the compiler.

Additionally, you can add a document to the draft titled “Meta-Data”. If this document is the very first document that Scrivener processes during compile, it will be tacked on to whatever project compile meta-data has been defined. Owing to how the way MMD meta-data works, fields which duplicate will be ignored the second time. So if you wish to use both compiler meta-data and document meta-data together, you should keep in mind that if you have a “Title” in the compile settings, any “Title” in the document will be ignored.

## 21.7 Further Information

Fletcher T. Penney, the creator of MultiMarkdown, is actively involved in supporting MultiMarkdown in Scrivener. He has generously contributed time and effort in writing tutorials and answering users' questions on the forums. See his web site and the Scrivener forums for more information. Thank you, Fletcher!

We also have an active MultiMarkdown section on our own software forum, where other authors have come together to share tips and tutorials. If you have any questions, it's one of the best places to go on the 'net.

**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, e-books, 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 it is comprised of. Due to the level of control that can be exercised at this stage, learning how to export can be itself a part of the learning process, but one that will in time benefit your ability to write freely in whatever manner you choose.

Most word processors are designed for business use, and approach document creation with the philosophy of showing you a good approximation of the final product as you work. This integrated way of working means that you are at once presented with the tools for writing with the same level of priority as the tools for formatting and layout. This not only creates a more confusing and distracting interface, but can lead to writing in a constricted environment, where you must work in a format which you are required to deliver.

Scrivener's approach, as with many things, allows flexibility. You can work in the described manner to a degree, where the text as you write it is formatted to how it will be delivered (although it will never present a full working preview in all ways), but you can also choose to completely separate that aspect from your writing experience, and let the compiler handle the details of formatting for you, from how titles are named, all the way down to whether or not your paragraphs are first-line indented or spaced apart.

Fortunately, the distinction between these two ways of working can be made as fuzzy as you like. That is, you can gradually introduce more automation into the compiler as you learn how to use it, rather than having to make a big jump all at once to a new way of working. In the chapter on compiling, you will be introduced to several basic presets that ship with Scrivener. These are intended to be starting points, and will not address everyone's individual ways of working or every university's style standards. If you are accustomed to working in a word processor, you might want to start with the Original preset and work from there. 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 other more comprehensive presets, such as the novel standard manuscript, or 12pt Times presets.

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 meta-data. 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. The topics that will be covered in this part are:

- \* **Compiling the Draft (chapter 22)**: Discusses compile strategies, format compatibilities, and gives a detailed run-down of every possible compile option.
- \* **Exporting (chapter 23)**: Transferring the contents of your project to your hard drive is a great way to back up your work outside of Scrivener entirely, or share bits of your project with other people. This chapter discusses the options available to you and how best to use them.
- \* **Printing (chapter 24)**: There are two ways of printing in Scrivener. One operates as an extension of the compile feature itself, and is the preferred way to print proofs or final submission-ready manuscripts. The second method lets you print quick one-offs of any file in the binder.

# Compiling the Draft

The main purpose of Scrivener is to provide a place that will help you write a long piece of text (whether novel, thesis or factual book), which may be structured as individual pieces in the binder but which can easily be output as one large file for working with “the rest of the world”. Thus, you write, gather and organise the material for your manuscript in the Draft folder, and when you are ready to export or print the manuscript as a whole, you use **File ► Compile...** (**Ctrl-Shift-E**) which 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.

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 it to another machine or share it with a collaborator. The available compile presets (which will be discussed shortly) are global to your machine, and can be applied to any of your projects, but once they are applied those settings are in your project. You can also save custom setups as your own presets to be used in this same fashion, or export them as portable files which can be used to share settings with colleagues, or update a second computer.

At its most basic usage, you can select from a number of handy presets using the **Format As:** drop-down menu on the top, and then supply a target file format using the **Compile For:** drop-down menu at the bottom. If you require more detailed control of the exported file, you can click the expansion button with the blue arrow to the right of the preset selection menu to reveal the full Compile interface. From here it is possible to control practically every element of your draft as it is processed into a single file.

## 22.1 Built-in Presets

In some cases, you may not need to look any further than the supplied presets that have been bundled with Scrivener. 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.

If you wish to customise these presets, you will want to read the section on Expanded Compile Interface, directly following, and any of the Option Panes which are relevant to you. Once you have changed a preset and saved or compiled, the selection menu will revert to “Custom”. Your custom settings will always be saved in the project, but if you change presets you will lose your custom settings. If you wish to save your settings for future use, read the following section ([section 22.4](#)) for instructions on how to do so.

In addition to these global presets, many of the project templates that ship with Scrivener come with useful compile settings as well, saved into the projects that are created from those templates. You can preserve these settings by exporting them as files with the **Save...** button available in the expanded compile interface.

The available built-in presets are:

- *Original*: This is the standard compiler setting. It will preserve your draft as closely as possible without adding any additional page breaks, titles, formatting adjustments, and so on. If you have very carefully designed your book from the start, using the text editor, this might be the best option, or at least the best starting point.
- *Enumerated Outline*: Only outputs the title for each document, rather than all of its text. It will use hierarchal numbering, and indenting to indicate the depth of items according to the structure of your book (up to six levels of indentation, though you can add further levels if you require). It has also been configured to accommodate synopses, which you can optionally enable in the Formatting pane. This is a useful starting point if you want a simple data sheet print out of all the pieces in your Draft folder.
- *Novel Standard Manuscript Format*: Formats your book using standard Courier 12pt type and a number of common conventions such as scene separators, double-spacing, underlined emphasis (instead of italic), page numbers, and so forth. Note to get the full benefit of this preset, you will want to use one of the rich text formats, such as RTF.
- *Times 12pt with Bold Folder Titles*: The use of 12pt double-spaced Times New Roman is increasingly becoming the submission standard in certain parts of the world. This preset will handle most cases, but you’ll want to check with your publisher for the precise specifications and adjust it accordingly. By default it will treat folders as chapters and leave regular text files (and text groups) without titles.

## 22.2 Available Compile Formats

The compile formats table (Table 22.1) shows all the formats supported by Scrivener, with commentary on their usage.

For most word processor-based workflows, including Microsoft Word, 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.

## 22.3 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. The important thing to know here is that each project's compile settings are saved as a part of its persistent settings. When you change settings and compile, they are saved back into the project and will stay that way until you change them. There is nothing special you have to do beyond this to save your export settings. In a moment we will discuss presets, which are a way of saving *sets* of settings for future use, so that they can be applied to projects. Thus, you only ever need use presets for the purpose of backing up or transferring settings between projects, since settings are automatically saved otherwise.

You can cancel any changes you've made by clicking the **Cancel** button at the bottom of the compile window. This will exit the dialogue without compiling, and in addition, it will reset any changes you've made to the compile interface since you loaded it.

The **Save & Close** button will perform all of the above steps, however it will not compile an output document. The settings will be saved into the project and the compile interface dismissed, but you needn't wait for a lengthy compile process to finish.

## 22.4 Expanded Compile Interface

The full compile interface (accessed by clicking the expansion button with the down-arrow to the right of the preset selection menu, labelled "Format As") is where fine-grained tweaks can be made to the compile process.

The full compile window is arranged into two main sections. On the left is a list of options pertaining to the various export formats you can choose from. Clicking on an option title in this list will reveal an associated configuration screen on the right side.

### 22.4.1 Managing Custom Presets

Compile settings are saved into the project whenever you compile. There is no need to repeatedly save the preset in most cases.



Table 22.1: Compile Formats Table

Format	Extension	Description
<b>General Purpose Formats</b>		
Rich Text Format	.rtf	General purpose rich text format supporting multiple fonts, images, tables, bullet points, footnotes and comments. Almost always the best option when exporting for use in a general-purpose word processor. This is a Microsoft Word compatible format.
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 and as such is useful for long-term archival.
Word Document	.doc	Industry standard Word .doc file.
Open Document Format	.odt	Format used by OpenOffice, LibreOffice, and other word processors.
Final Draft 8	.fdx	For transferring your script to Final Draft, this is the best option unless you have an older version of Final Draft than version 8. Maintains synopses (as scene summaries), scene titles and custom script element formatting.
<b>Web Support</b>		
Web Page	.html	Creates a single HTML file suitable for web-publishing.
eXtensible Web Page	.xhtml	Likewise, though using the newer XHTML format and is thus better suited toward modern web publishing platforms.
<b>Static Layout Support</b>		
Print PDF	N/A .pdf	Use to send the compiled draft directly to your printer. The Portable Document Format is useful for archival and sending copies out for read-only use, as it will look identical on all machines.
PostScript	.ps	A typesetting standard similar to PDF in what it provides, a stable format that will appear identical on any machine or printout, used more commonly in the design and publishing industry.
ePub eBook	.epub	Generate feature-rich e-books for use in portable reading devices that support the ePub format, such as the Sony Reader, Nook or iPad (ePub files can be dragged into the iTunes Library to import them into iBooks).
Kindle eBook	.mobi	Generate feature-rich eBooks for use in portable reading devices that support the .mobi format, such as the Amazon Kindle. Requires Amazon's KindleGen.

Table 22.2: Compile Formats Table Cont'd

Format	Extension	Description
<b>MultiMarkdown Post-processing</b>		
MultiMarkdown	.md	Export a plain-text MultiMarkdown file, useful for archiving, or further custom post-processing.
MultiMarkdown -> 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 exporter, above.
MultiMarkdown -> RTF	.rtf	Export an HTML-derived RTF file with partial MultiMarkdown support. Note that the RTF files generated in this fashion will be limited in features and formatting support in the same way as HTML, and is not equivalent to Scrivener's standard RTF exporter in the word processor compatibility tables.
MultiMarkdown -> Web Page	.html	Generates a W3C compliant HTML5 file, suitable for web-publishing or further XML post-processing.
MultiMarkdown -> Flat XML	.fodt	This ODF compatible format is suitable for bringing your MMD project into the word processor world. At the time of this writing, LibreOffice is the main OpenOffice fork that can read .fodt files without modification.

So when should you save compile settings? You may wish to save them as a preset for use in other projects, or simply to temporarily use a special-purpose compile preset (like Enumerated Outline) without losing all of the work you have put into your settings. Saving presets will add them to the main **Format As:** drop-down menu at the top of the compile sheet, meaning from that point on, you can apply them to any other project you create.

Below the list of option panes on the left, are two buttons which let you save your current settings to a preset.

**Save Preset...** Presents a dialogue box in which you can title the preset. This is how it will appear in the preset menu, so choose a meaningful name that you will remember in the future.

**Load Preset...** Provides management features for your custom presets. You can import saved settings that you may have downloaded from the Internet; export a custom preset to a file that you can save for backup purposes, or share with others; and remove presets you no longer need. The load button needn't be used to apply presets to your current settings (although it can do so as well by selecting one and

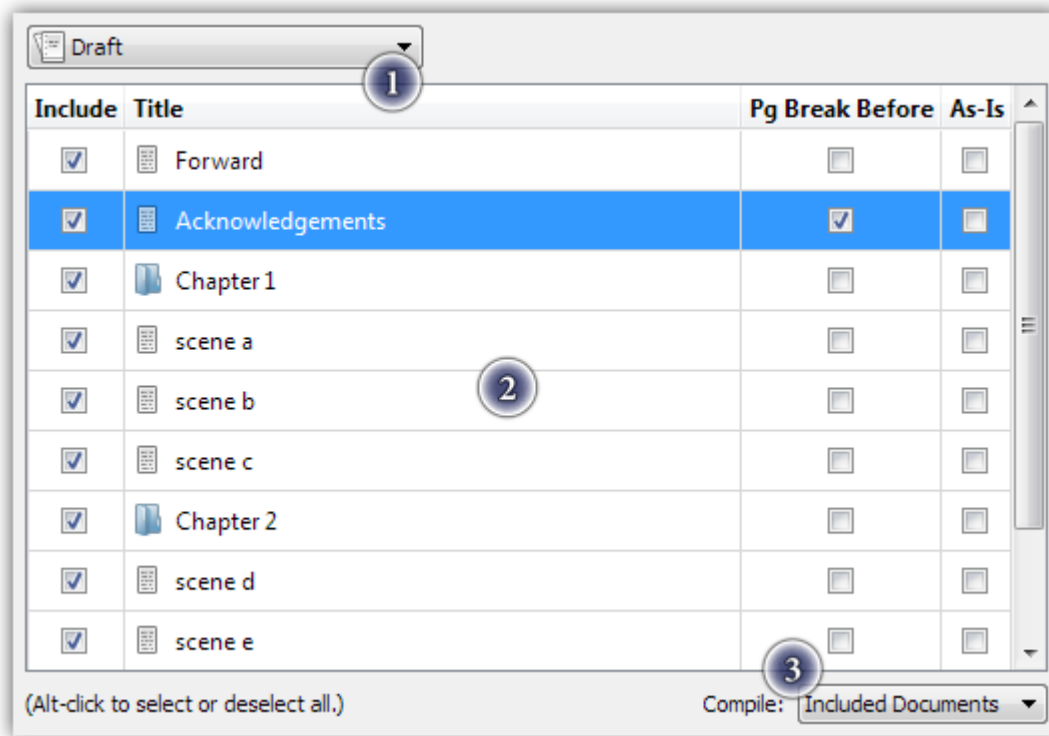


Figure 22.1: Compile—Contents Pane

clicking **OK** instead of **Cancel**). The easiest way to apply a preset you already have installed is to use the “Format As” drop-down menu. The available buttons in the Load interface are:

- *Import*: brings up a file browser. Locate a saved compile preset on your drive to install it into Scrivener
- *Export*: the selected preset in the left list will be saved as a file on your disk which you can then use to carry to another machine, send to a colleague, or back up for safe keeping
- *Delete*: removes the selected preset from your disk. It will be placed into your system Trash folder

## 22.5 Contents

This option pane is a core feature available to all export formats.

The Contents pane is where you define which parts of the Draft Scrivener will use to create your exported file. To accomplish this, it provides several tools of varying scope.

The method with the most immediate impact is the drop-down menu at the very top of the interface, marked as 1 (Figure 22.1), which sets the “Compile Group”. Ordinarily this will have the Draft selected (which might be called something else if you have renamed it, or started with a template).

In cases where you wish to proof only a portion of the book, or are working in a compound project that includes several editions or articles located in the Draft, you can use this drop-down to select only a portion of the draft folder. Whatever container you select will include everything from that point downward, or in outliner terminology, its descendants. You could, for example, choose the folder called “Chapter 1”. All of the scene files included in the folder will be used in the Compile, including the chapter folder itself, but nothing outside of that folder will be used.

### 22.5.1 Content Item List

The item list is the large table in the middle, displaying the contents of the current compile group. In the case of the illustrated example, marked as 2 (Figure 22.1), the Draft has been selected, and so the entire Draft contents are revealed in the list below it.

There are four columns in this list. It is usually a good idea to initially scan these columns to make sure everything will act in the manner you expect it to. On any of the checkbox columns, you can Alt-click any of the checkboxes to toggle checkmarks for all visible items in that column.

**Include** Linked to the Inspector option, “Include in Compile”, if unchecked the item will not be used in the final product unless the compiler has specifically be instructed to use this checkbox in an unorthodox fashion (below). 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, you should use the compile group selector or filters.

**Title** The visible name of the item in the Binder. This might be altered in subsequent compile options, or updated with a counter if it includes a token in the name.

**Pg Break Before** Linked to the Inspector option, “Page Break Before”.

Inserts a page break control before the checked document. A common usage for this is to set special pages like title pages and table of contents. For items which create a repeating pattern where a page break would be needed, you should use Separators (section 22.6) instead of individually marking checkboxes for each item; and indeed most default settings include an automatic page break wherever a folder follows a file (often used to denote a new chapter). With some compile formats,

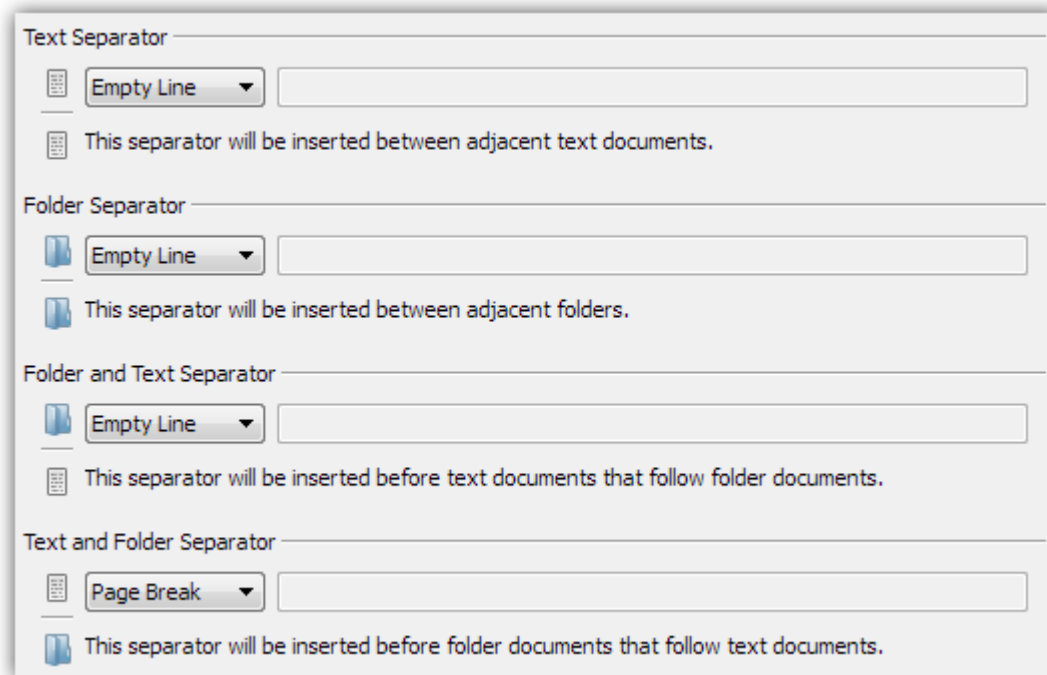


Figure 22.2: Compile—Separators Pane

page breaks can be important, particularly the e-book formats, which use section breaks to determine the automatically generated table of contents.

**As-Is** Linked to the Inspector option, “Compile As-Is”.

All formatting set in the formatting pane (section 22.8) will be ignored for the checked document. This also means no extra material will be added, including titles or synopsis. Text will always be included, even if it would otherwise be excluded.

In the lower-right corner (marked as 3) is an override which can reverse the settings of the “Include in Draft” checkmark. By default, the behaviour follows the logic of the document settings. You can however invert the logic and only export those items marked to *not* be ordinarily exported—or just ignore the flags altogether and export everything.

## 22.6 Separators

Separators are used to automatically insert space, breaks, or custom symbols between the various elements in the list which will be compiled. The important aspect to realise is that by the time the compiler gets to this point, your outline is being considered as a flat

list of files. It doesn't matter if a text file is four layers deep, if the next item in the list is a folder, it will still use the "Text and Folder Separator" rule.

Separators will not clean up any manually typed in separation that may exist between items in the editor text itself. For example, if you have items with a lot of empty lines in between them, the separators control will only establish if any *additional* separation should be added.

There are four primary types of possible combinations, each with its own individual rule. You can supply the same rule for all of them if you do not need any special differences between types. The types are:

- *Text separator*: inserted between any two adjacent text documents
- *Folder separator*: inserted between any two folders
- *Folder and text separator*: inserted at the top of any file that directly follows a folder
- *Text and folder separator*: inserted at the top of any folder that directly follows a text file

Each individual rule has four options, some of which change their behaviour depending on the compile format:

- *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", and is the most minimal option you can choose
- *Empty line*: two paragraph breaks will be inserted, causing a visible space between the items.
- *Page break*: a page break code will be inserted, causing the following item's text to move to the next page of the manuscript
- *Custom*: Anything entered into the adjacent text field will be placed between the two items. Note that the text or symbols entered here will be buffered by a single return on each side, forcing it to fall on its own line. The value will be centre aligned.

The following formats have different options and behaviours from those listed above:

**Plain-text Files** Fifteen empty lines will be inserted between the two documents, setting them apart from one another, since plain-text files cannot utilise page breaks, being composed of no pages themselves.

**HTML and XHTML** Empty paragraphs will be inserted between the two documents, spacing out the text. As with plain-text, single web pages do not themselves have pagination within them.

**ePub and Kindle** Since page breaks have no meaning in an e-book, the “Page Break” option will be changed to “Section Break”. This will not only cause the e-reader to switch to a new virtual page (if it chooses to display things in such a way), but will also insert the name of the item with the section break into the automatically generated Table of Contents.

**MultiMarkdown, and MMD HTML formats** In most cases the “empty line” option should be used to retain clean paragraph breaks between blocks. The “single line” type will cause the last paragraph of one section to flow directly into the next paragraph of the next section.

Page breaks are replaced by “section breaks”, which emit a series of hyphens on their own line. This will be converted to a horizontal rule using the HTML converter.

**MultiMarkdown LaTeX format** As with the other MMD formats, use of the “empty line” type will produce the best result in most cases.

The LaTeX code for a page break, `\pagebreak`, will be inserted for the “page break” option. Note that this will be enclosed in an HTML comment, which is the preferred method for passing LaTeX codes straight through the MMD system. This ability is disabled by default, and will need to be enabled by hand in the MMD sub-system ([chapter 21](#)).

## 22.7 Cover

Available only to the ePub and Kindle exporters, the Cover pane sets the cover and presentation material which will give your e-book a professional finish. The cover image will be placed at the very beginning of the e-book (before the table of contents). Most e-book readers automatically use the cover image itself to provide their thumbnails.

Very likely you’ve been provided with images from a designer that was assigned to illustrate your work. They may not provide you with the file sizes necessary for e-books, since this is still a new market. Graphics for cover art should be in a standard RGB raster format, such as JPEG<sup>1</sup>, and not a vector format, such as EPS, nor CMYK colour

---

<sup>1</sup>At the time of this writing, JPG is the safest format to use. Not only is it the most compact, some reader devices do not support the PNG format.

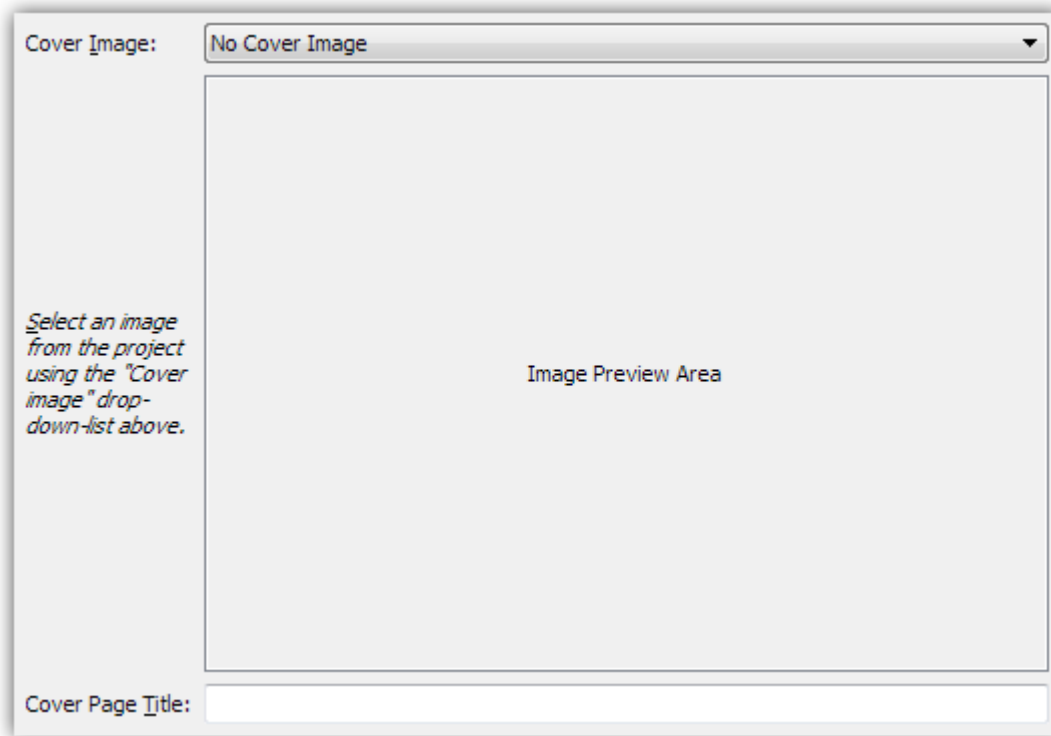


Figure 22.3: Compile—Cover & ToC

space which is designed for printing. For dimensions, there is no firm rule, but a size of roughly 800 pixels tall by 600 pixels wide at 72 DPI will display crisply on nearly all reading devices and computer screens with unnecessarily bloating the size of the e-book. 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.

In particular, watch out for very large files that were originally designed for full colour press. These will unnecessarily enlarge the size of your e-book, make it difficult to send out proof copies via e-mail, and could even cause e-book reader devices to run out of memory.

**Cover image** Cover images must be selected from resources within the current project.

If you have a cover that you wish to use with your book, make sure you have imported it into the binder first. All images will be listed in this drop-down menu, and will be displayed in the grey preview area in the middle of the pane once selected.

**Cover page title [ePub only]** You should not supply the name of your book here. This



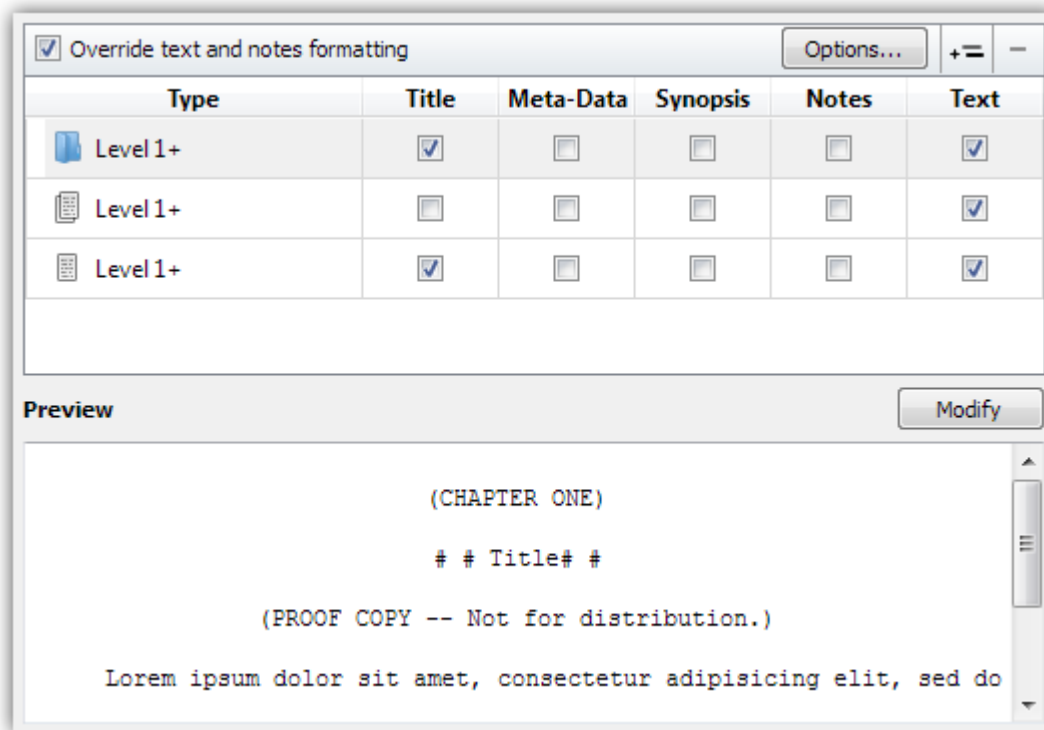


Figure 22.4: Compile—Formatting Pane

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.

## 22.8 Formatting

The Formatting pane is where you will design the look and structure of your document. It is separated into two main areas, both of which we will gradually introduce:

1. *Structure and Content Table*: A table for assigning content to the various Binder item types available, and the creation of level specific content and formatting rules.
2. *Formatting Editor*: A format editing and live preview area, which is linked with and displays the formatting rules for the selected row, above.

In the supplied example (Figure 22.4), the manuscript will print all folder titles (including any text within that folder item itself), and only text for file groups and files. It will also override the formatting to Courier 12pt, and adds extra information to the folder titles, one of which warns readers that they are holding a proof copy. All of this might

seem like magic right now, even if it gives you a taste of what the compiler is capable of, so let's take a look at the provided features. By the time you are done with this section, you should be able to see how all of this fits together, and in no time you'll be designing your own documents.

### 22.8.1 Structure and Content Table

The structure table initially has three rows which correspond to the three binder basic item types that can be found in a draft folder: folders, file groups (text files that have subdocuments), and text files<sup>2</sup>. For a moment ignore the “Level 1+” labels, we'll get to that in a moment. In the columns to the right of each type you will find a checkbox for each content element that can be optionally included by the compiler. Think of these as optional material you can include from your project, for *each* item in the Draft. As Scrivener proceeds down the list of items it has been instructed to compile, it will consult this table for instructions on how to handle it. If that item is a folder, it will look at the folder row and if “Title” and “Text” have been checked, will add the binder title for that item into the manuscript, followed by any text that is in that folder file. If the next item is a text file, it checks the text file row. If that row says only “Text” is exported, then it will print just the text content of that file into the manuscript.

Creative use of this feature can produce outlines (Titles only for all types), a document with only your notes and no manuscript text, chapter containers that export only their title, and so forth. Note that as you toggle elements on and off, you'll see a live preview of how they will be inserted into the compile, in the Formatting Editor below the table. We'll discuss how to change the look and feel of this in the next section.

Things get more interesting when these rules are applied to levels of hierarchy. This ability is best demonstrated with an example. Try selecting the Folder row in the table, and then click the “Add formatting level” button, indicated with an arrow (Figure 22.5). The label for the row that was previously selected will now be titled, “Level 1”, and the new indented folder row will be called “Level 2+”. As you might guess, this means you now have *two* rules for handling folders. The first rule will apply to level one folders only, the second rule will apply to level two folders, and anything deeper in the hierarchy than that. To start, all configuration options from the original row will be carried over into the new row to help you get started in customising it. This holds true for formatting as well as the options we've already discussed.

What do these levels correspond with? They are matched with outliner depth in your binder. An example binder structure has been inset in the example, with the items named

---

<sup>2</sup>Some of the provided project templates might configure a number of formatting rows beyond the basic default three. A new blank project, however, will always just have three rows, one for each type. You might want to create a new blank project in order to follow along with this explanation.

Type	Title	Meta-Data	Synopsis	Notes	Text
Level 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Level 2+	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Level 1+	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Level 1+	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

- Level 1
- Level 2+
- Level 1
- Level 2
- Level 3
- Level 2
- Level 1

← Example Binder

Figure 22.5: Example multi-level formatting rule.

to indicate how they correlate to “levels” in the Formatting pane. Items directly beneath the Draft are considered to be “Level 1”, a child item beneath one of those is “Level 2” and so on. The way Scrivener’s formatting engine works is by applying a format to the maximum depth in your draft outline. Inserting rules will thus only apply to those levels which it states in the label. In the provided example, I’ve added a checkbox to insert Meta-Data (creation and modification dates, labels, and so forth) to any folders at level two or deeper. Folders directly descendant from the Draft will only print their titles in the compiled document, but all folders at level two or greater will also print their meta-data.

To remove a level-based row, you can select it and click the - button to the right of the “Add formatting level” button.

There are other buttons and options in this table, but before we take a look at them, let’s examine the formatting editor.

### 22.8.2 Formatting Editor

The formatting editor view is linked with the structure table above it. When you select a row in that table, you are also looking at the format for that row as well. Thus, each *type* of binder item can have its own formatting rules, and further *levels of depth* in the binder can have their own formatting rules. A very simple example of this would be different heading sizes depending on the level.

While added components, such as Title and Synopsis, can be formatted to your choos-

ing, the main text area is special. To make any changes to the main text body formatting, the first thing you need to do is enable the formatting override. In the header for the structure table, click the checkbox labelled, **Override text and notes formatting**. When this option is engaged, your entire manuscript will be given a uniform appearance which can, if you choose, be radically different from how it looks in Scrivener.

To edit the formatting, you will need to click the **Modify** button on the right-hand side above the preview area. You will find a condensed Format Bar, a Ruler, some special options at the top, and a large text preview area. This cannot be typed in, it shows you what elements have been selected in the structure table, and how they will be formatted in the manuscript. You can otherwise treat it just like an ordinary editor. If you click in the “Title” area, you should see the formatting controls adjust to represent whatever formatting has been applied to the title. Try clicking the **Underline** button. Unlike with standard text, you are editing “areas” of text, so there is no need to select the entire title, you can just place your cursor anywhere within the title and click **Underline**. Refer to the documentation on using the The Format Bar ([subsection 15.4.2](#)) if you are unsure of what these various buttons do.

**E-books and Fonts:** Unless you have purchased distribution rights for the font you’ve selected, fonts cannot be legally embedded into the e-book (and Scrivener is not capable of performing this action; it would need to be done after compiling). Consequently, the precise font family that you choose in this pane will probably not be the font you see in the e-book. In most cases, this is precisely what you want. The e-book reader hardware or software will provide fonts for the reader, and most will want to use their preferred font, size and styling if given the choice. The exception to this is that the font family *will* be declared in the formatting meta-data, so if that font is available on the device or computer, then it will in some cases be used. Again, this is up to the software itself as to whether or not it should override the reader’s preferences with your font choice.

In addition to the standard controls detailed above, additional elements will be provided along the header bar of the formatting editor, if relevant to the particular format in use.

**Page Padding** Located on the right hand side of the formatting editor header bar, this controls how much padding will be added to the top of the page, in lines, when it happens to fall on a page break. Use this to offset your titles from the top of the printable margins if you prefer that look. This option will not be previewed in the Formatting pane, you will need to compile in order to see the results.

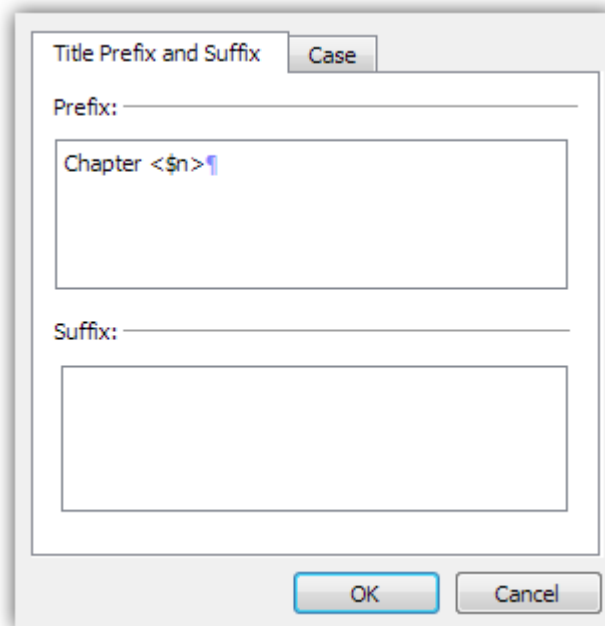


Figure 22.6: Title Prefix and Suffix tab

When using the plain-text output format, blank lines will be inserted wherever a page break would appear. Thus this setting can be used to visually space apart sections, even though in most text editors, page breaks are not possible. This option is not available to the MultiMarkdown formats, as this engine does not heed extraneous spacing.

### Section Layout

The “Section Layout...” button provides options for adjusting the content boilerplate text around headings, heading formatting and letter case adjustments. These adjustments will only appear when you compile (though some will be previewed for you in the mock editor below the button). They will not be added to the binder items’ titles or adjust the base text itself, and are thus useful for keeping clutter or awkward style choices (like all caps) out of the binder and editor. Rather than putting “Chapter <\$n>:” into every single chapter folder in the binder, you can simply add that to the Prefix and be done with it, never having to worry about renumbering after an extensive restructuring again.

The Title Prefix and Suffix tab (Figure 22.6) handles the insertion of arbitrary text around the title of the binder item itself. If you typed “Test - ” into the prefix field for folders, and compiled a draft with a folder named “Compiling with Prefixes”, then the final output would be “Test - Compiling with Prefixes”. In fact every folder included

for compile would have “Test - ” in front of its binder name. The suffix acts in a similar manner, only adding text to the end of the title. Changes made to the prefix and suffix fields can be viewed in the preview area, after dismissing the Section Layout window.

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*. This can prove very useful in some cases. It could for example let you use casual titles in the binder, and standard generic numbered titles in the final output. These are set specifically for each formatting rule established in the structure table. This means a folder on level 1 could use **Part <\$R>**, while a folder on level 2 could use **Chapter <\$n>**, and folders of level three and higher would use the hierarchical numbering provided by the **??** tag. Meanwhile text items could use something else entirely, or nothing at all.

Once added, formatting for the prefix and suffix can be handled independently from the title. If you wish to maintain a uniform look between the three components, use the mouse to select all of the visible title elements before styling. If the entire heading will be output on a single line, then only character formatting may differ between the three. If however components are located on a new line (you can add carriage returns to the prefix and suffix box, to accomplish this), then each can have differing paragraph and ruler settings, too.

**Using Distinct Formatting for Numbers** A common typesetting technique for chapter titles is to use a larger numeral. If you place the counter tag in the same field as the rest of the boilerplate text, you will be unable to size the number differently. Instead, place the boilerplate text (such as “Chapter “) into the prefix and the numeral tag in the suffix. Now you can control the font characteristics of the two components independently.

Tab characters and carriage returns can be added to these fields as well, and they have been set to show invisible characters in order to facilitate designing more complex or multi-line title adornments.

In both cases, no spaces will be inserted around the prefix or suffix. You will need to establish all spacing, if any, to be used in setting them apart from the main binder title.

**Prefix** Anything typed into this box will be printed before the relevant binder item’s title, on that same line. 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.

**Suffix** The contents of this box will be printed after the binder item’s title, or if the binder item is not being printed, directly after 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.

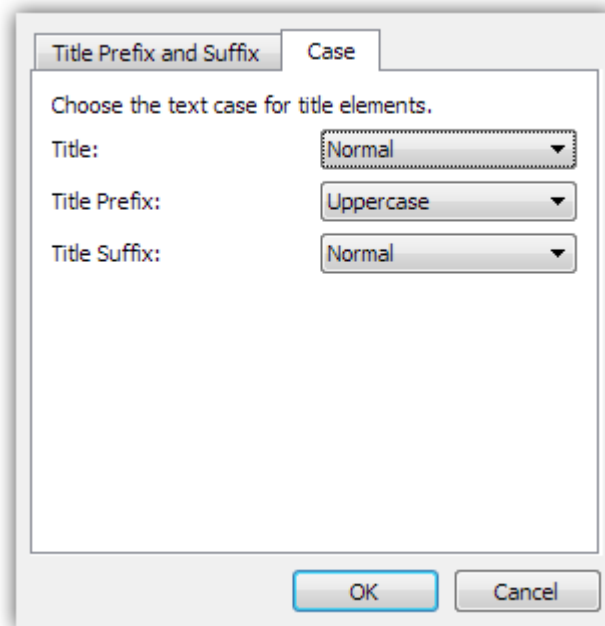


Figure 22.7: Title Appearance tab

**MultiMarkdown and Titles** The title prefix and suffix will ordinarily be placed within the hash marks that Scrivener generates to indicate title depth. If you prefer, you can disable the “Place prefix/suffix inside hashes” options to allow text entry outside of the header line itself. To allow compatibility with the other compile formats, you will need to manually type in carriage returns to separate lines. Placing characters before or after the hashmarks on a title line will at best cause it to no longer render as a title correctly.

### Title Appearance

In the [Section Layout...](#) dialogue, the Case tab ([Figure 22.7](#)) has three options for transforming the Title, Title Prefix, and Title Suffix, independently. These transformers adjust the letter-case appearance of the text while compiling, so you don’t have to use uppercase titles in your Binder.

### 22.8.3 Special Formatting Options

Returning to the structure table, the [Options...](#) button in the table header bar will open a sheet which lets you set a few options for how elements should be output. These settings are global to the entire compilation, and cannot be set specifically per formatting rule.

**Selective Opt-Out:** You can always set specific parts of the Draft to ignore all formatting, no matter what settings you choose in the compiler here, by using the **Compile As-Is** Inspector option; which also can be strategically administrated from the Contents compile options pane. This will even ignore selected elements from the structure table, such as whether Title or Notes are checked off. Only the *Text* of such items will be output (even if it otherwise would not output text), and it will be output precisely as shown in the main editor. Absolutely no options from the Formatting pane will be used when outputting these sections.

**Insert subtitles between text elements** When either Synopsis or Notes elements are enabled, Scrivener will insert a subtitle to help set apart these sections from the main text, which will itself also be titled. These titles can be formatted like everything else in the formatting editor in a unified fashion. You only need to edit the formatting of one subtitle to impact them all.

**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.

## 22.9 Script Settings

This set of options lets you configure a few details which impact how a script will be formatted when using the Final Draft 8 (FDX) format.

**Include footnotes as script notes** Inline and linked footnotes will be converted to Final Draft's script notes feature. 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 Final Draft's script notes feature. All comments will be removed from the compiled document if this option is not enabled.

## 22.10 Transformations

The Transformations pane provides a number of useful text and format conversions (where applicable), as well as a few presentation options. Some of the options listed will not pertain to every possible export format. For instance, the ability to convert italics to underlines is meaningless in a plain-text export. They will simply be ignored if they are not relevant.



### 22.10.1 Plain Text Conversions

These are useful conversion tools for translating special characters and some ruler styles to plain-text variants. These options are made available to the rich text formats as well.

**Straighten smart quotes** Typographic quotes will be substituted with straightened variants.

**Convert em-dashes to double hyphens** The ‘—’ character will be converted to -.

**Convert ellipses to triple periods** The ‘...’ character will be converted to three full stops.

### 22.10.2 Rich Text Conversions

This section provides options for automatically converting a few rich text conventions from one standard to another. They are only made available to the compile formats which support styled text.

**Convert italics to underlines** If the submission process requires underscoring to be used instead of italics, this feature will let you write in italics but produce a properly underscored manuscript.

This option is not available to any formats which produce a plain-text document.

**Convert underlines to italics** Use when you have produced a document with underlines, but need an italic version for compile.

This option is not available to any formats which produce a plain-text document.

## 22.11 Replacements

Replacements are a way for you to set up your own substitutions, which work in a manner very similar to Search and Replace, though without changing the underlying documents in Scrivener.

A table with four columns has been provided:

- *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.
- *Case-Sensitive*: when checked, the letter case in the *Replace* column must precisely match, otherwise it will be ignored by the compiler.

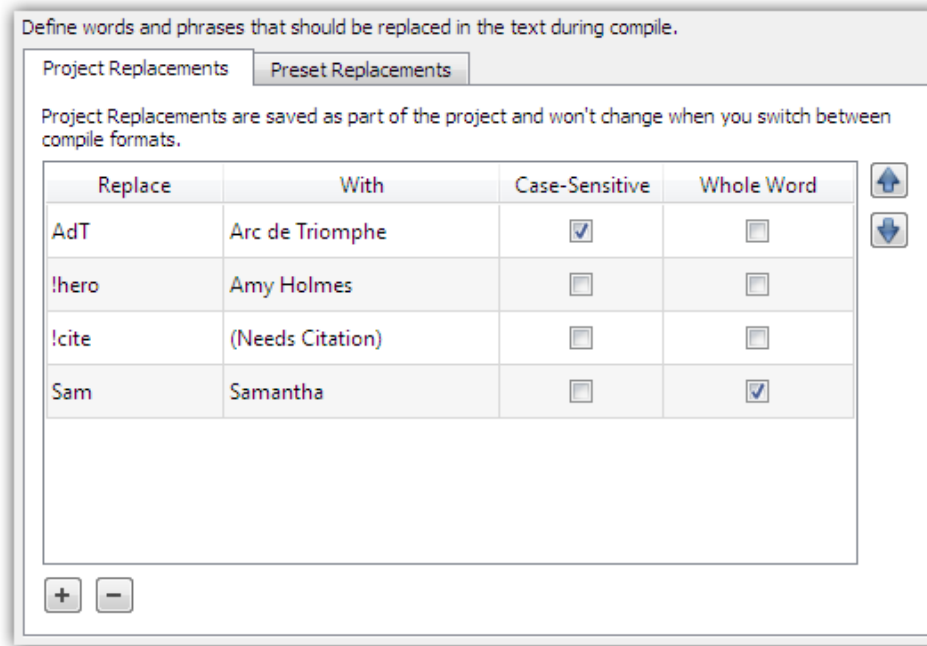


Figure 22.8: Replacements compile options pane

- *Whole Word*: restricts the match to only those incidences where it has a clear space around the phrase. “Sam” will not match “Sammy” when this option is active.

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.

Two tabs are presented to you, “Project Replacements” and “Preset Replacements”. It is important to know that while it may sound otherwise, the Replacements in the Preset tab are initially stored in your project as well. Everything in the compile pane is stored in your project. The fundamental difference between these two tabs is what happens when you work with other presets. Preset Replacements will be saved into custom presets when you create your own or update an existing one, but Project Replacements never will. Likewise, Project Replacements are totally protected from any changes made to your compile preset. If you switch from the template settings to an outlining preset, for instance, the Preset Replacements will be reset (usually to blank), but the Project Replacements will be safe. So in summary:

- *Project Replacements*: always stick even if you change presets, and are never saved into a preset. These are best for things that are highly specific to a particular work,

like name substitutions or expansions specific to the project.

- *Preset Replacements*: are saved into your project file as well, but are volatile in that changing a preset will reset the list to whatever that preset defines (including nothing, if that be the case). These are best for things which suit an entire category of works.

An example of how Preset Replacements work:

1. You have a project you created using a built-in template. This template comes with a few replacements built-in, but you decided you want to add your own. At this point the project no longer using the preset, it has now “forked” into its own unique set of settings, and therefore the “Preset Replacements” tab is disconnected from the original preset
2. You remove one of the built-in replacements you never use, and add a new preset replacement you’d like to use in future projects
3. To make these changes available in the future (or to other projects you are currently working on), use the **Save Preset...** button below the list of option panes ([subsection 22.4.1](#)). Let’s say you call this “Research Article”. Now your project is once again synced with a format preset on the disk, and that preset will be available to other projects.

At this point you could make a new project and select that preset, acquiring all of the preset replacements available from it. If you ever wished to update it, however, you would not only need to add the items to the Preset Replacements tab, but *update* the preset on the disk as well. Any project using that preset will benefit from the new presets, but those that *once* used it and no longer are (because you’ve customised the settings in one way or another) will need to be refreshed from the newly updated preset, or updated manually by copying and pasting Replacements between the compile panes of two open projects.

Replacements are evaluated in the order they are defined. This means you can use earlier replacements in your subsequent replacements, if you are feeling brave.

### 22.11.1 Advanced Replacements Usage

Replacements can also take a special tag which will be used to match everything *within* the replacement match, that does not otherwise match the “structure” of the replacement tag. This is much more easily explained with an example. The character sequence `$@` can be inserted into the first column anywhere within the replacement field. An example

might be, `^$@^`. Now in your base text, if the replacement field encounters a caret symbol, it will seek for a second caret symbol and then copy everything in between those two symbols into the `$@` tag. This can be placed into the “With” column, amidst your custom expansion code. A practical example that would be useful for those writing directly in HTML:

```
^$@^
```

Replaced with:

```
<span class="index">$@</span>
```

Will take the following example text from your compiled document:

```
This is a ^test^ of advanced replacements.
```

And convert it to:

```
This is a <span class="index">test</span> of advanced replacements.
```

The `$@` character may be placed anywhere within the “With” field, but must be completely contained within delimiters in the “Replace” field. It can also be used multiple times within the “With” field. This means you can capture some text and then use it multiple times in the output. Here is a possible example, using LaTeX:

```
^$@^
```

Replaced with:

```
\index{$@}$@
```

Using the above example text, the output would be:

```
This is a \index{test}test of advanced replacements.
```

## 22.12 Footnotes/Comments

The Footnotes and Comments pane controls how these two forms of notation will be handled in the compile process, or whether they should even appear at all. Since there are two ways to add notes to documents (linked or inline notes), it is with some formats

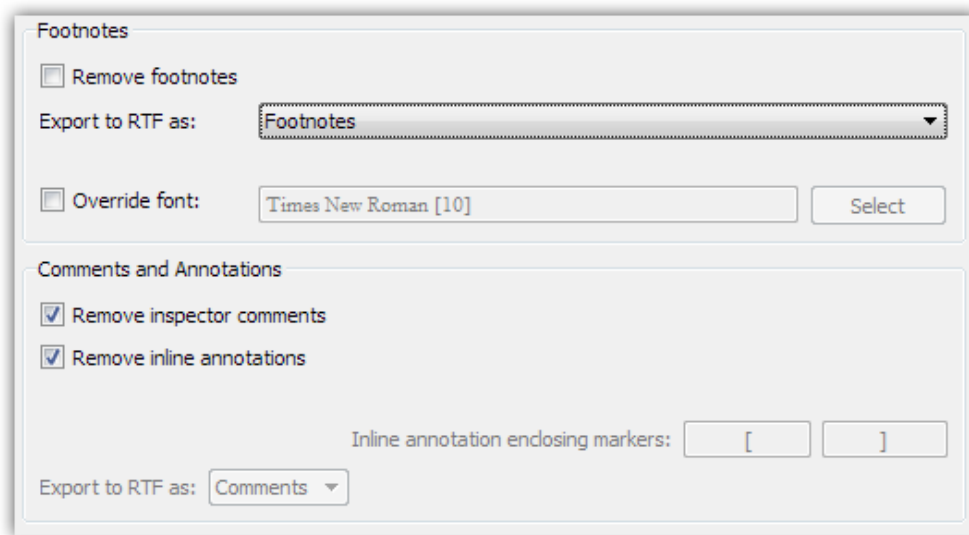


Figure 22.9: Compile—Footnotes/Comments Pane (RTF)

possible to treat them differently. As with other areas of the application and documentation, the usage of the term “footnote” is equally applicable to “endnote”, if the options to use endnotes have been applied.

The first section concerns itself with how footnotes should be arranged in the manuscript, and the second section deals with inline annotations and linked comments. Since not all of the available export formats support all available options, some variants of the pane will appear differently than shown (Figure 22.9).

This option pane is a core feature available to all export formats except for FDX, where footnotes are never used.

### 22.12.1 Footnote Options

Some formats do not support real footnotes or endnotes. When using these formats, Scrivener will create the appearance of endnotes by embedding look-alike text representations for both the marker and the footnote itself. Since most of these formats also do not have a concept of a “page”, it will be impossible for them to generate true footnotes. Consequently, they will nearly always be inserted at the very end of the document, becoming in effect, endnotes. When using these formats, the available options will reflect what is capable.

**Remove footnotes** The exported manuscript will have all footnotes (inline or linked) stripped from it. When this option is enabled, the rest of the footnote section will

be disabled.

**Override font** If you wish to use a separate font to print footnotes, this can be set here. Note that many word processors have their own special stylesheets for handling footnote appearance, so this option may not always be necessary. This format will be used consistently for all footnotes, even if they are featured in sections which have substantially different base formatting than the rest. Ordinarily, footnotes pick up the paragraph settings of the paragraphs they fall after.

**Group footnotes...** This feature allows you to gather footnotes together into a determined place, for formats that do not otherwise support true footnotes. It will toggle the default behaviour whereby all footnotes are attached to the very end of the manuscript, and will instead insert them at the nearest subsequent page break, becoming in most cases, chapter-based endnotes.

Available to Print, PDF, DOCX, ODT, (X)HTML, PS and TXT.

**Footnote Separator** Lets you choose what form of separation you would like between footnotes and the main text body. The available options are: single return, empty line, page break and a horizontal line.

**Footnote page title** When “Page break” is chosen as the separator (above), the contents of this text field will be used to title that section.

### 22.12.2 Comments and Annotations Options

As with footnotes, it is possible to use both inline annotations and linked comments in the same project, and selectively handle them during compile. You may wish, for example, to use inline annotations for simple editing notes, and linked comments for references to source material. It might then be useful for collaborators to see the source material references, but not the simple inline editing comments to yourself.

Most of the compile formats have the same options available, but will handle the export of annotations and comments in different ways, depending on what is supported within that format.

- *RTF/DOC*: Provide additional options for utilising margin comments. Unfortunately this feature is not widely supported by other programs. If comments refuse to show up in your preferred word processor, try using one of the other methods listed below.
- *Other rich text and word processor formats*: In all of the formats that support the ability to set font colours, annotations and comments will be inserted as actual

text in the document, enclosed in markers, and then coloured according to the annotation text colour, or linked comment background colour, in the project. For the purposes of this definition, HTML and the e-book formats are considered “rich text”.

- *Text based formats:* For formats like plain-text and MultiMarkdown (when the HTML method is disabled), annotations and comments will be simply placed into the text where they originally appeared in the project. You can control what, if anything, is used to enclose the comment in the text (square brackets are used by default).

The following options are available:

**Remove inspector comments** Only inspector comments will be stripped from the resulting manuscript.

**Remove inline annotations** Only inline annotations will be stripped from the text. If you wish for all comments to be removed from the manuscript, make sure both this and the above options are checked.

**Export to RTF as...** Choose between standard “Footnotes” (at the bottom of every page), “Endnotes” (at the end of the work), or “Comments”, which will convert them into margin notes in supporting word processors. This option is also applicable to the DOC format.

**Inline annotation enclosing markers** If annotations are to be left in place, you can set the two markers that will be used to delineate them from the rest of the text. Someone who is publishing in HTML might for instance wish to use the open and close syntax for HTML comments. Those using MMD to create LaTeX files might do the same, as HTML comments allow one to write raw LaTeX into the editor.

### 22.12.3 MultiMarkdown Options

The MultiMarkdown compile formats handle footnotes uniquely from the other formatters. Scrivener will, by default, automatically generate and ID the necessary footnote syntax for you. An MMD marker will be inserted wherever the footnote has been placed in the document, and the appropriately matching footnote text syntax will be inserted elsewhere in the compiled MMD file.

**Convert to HTML-styled text** Since MultiMarkdown has no good analogue syntax for comments, by default Scrivener will insert a styled HTML container around the

comment. Both inline annotations and linked comments will be treated identically in this fashion. Since MMD will pass HTML straight through its engine, the result will be a range of text coloured with whatever colour the annotation text or comment background was set to. The appearance, sans the bubble around the text, will thus be identical.

When this option is disabled, annotations and linked comments will be exported in the same fashion they are for plain-text files; enclosed in the defined markers. This latter option should be used with MultiMarkdown version 3, which is no longer XHTML based when using non-HTML target formats. Using HTML styled inline annotations in MMD3 will cause them to disappear from the output. Use the enclosing marker fields to instead inject code of your choosing. You could for instance use an open and close marker pair of: `<!-- \marginpar{` and `} -->`, to convert your inline annotations to L<sup>A</sup>T<sub>E</sub>X margin notes.

**Pro Tip:** In cases where an annotation falls entirely on its own paragraph, Scrivener will use a styled div instead of a styled 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.

## 22.13 Page Settings

Various print and page settings can be set up with the Page Settings pane, such as margins, paper size, headers and footers. Only formats that support page settings will use this pane, and amongst those not all support headers and footers.

**Paper Size** Here you can set the physical size of the page. Common paper sizes are provided, or you can supply your own custom sizes.

**Margins** Margins set how far from the edge of the paper stock the text container will extend to. This measurement does not factor in the non-printable area, which is printer specific, so when using very low values it is a good idea to check how close to the edge of the paper your printer can go, to avoid truncating the text.

### 22.13.1 Headers and Footers

These settings will impact the entire document, though you can opt out of using them on the first page with the “Not on page 1” checkbox. Further, you can also choose to



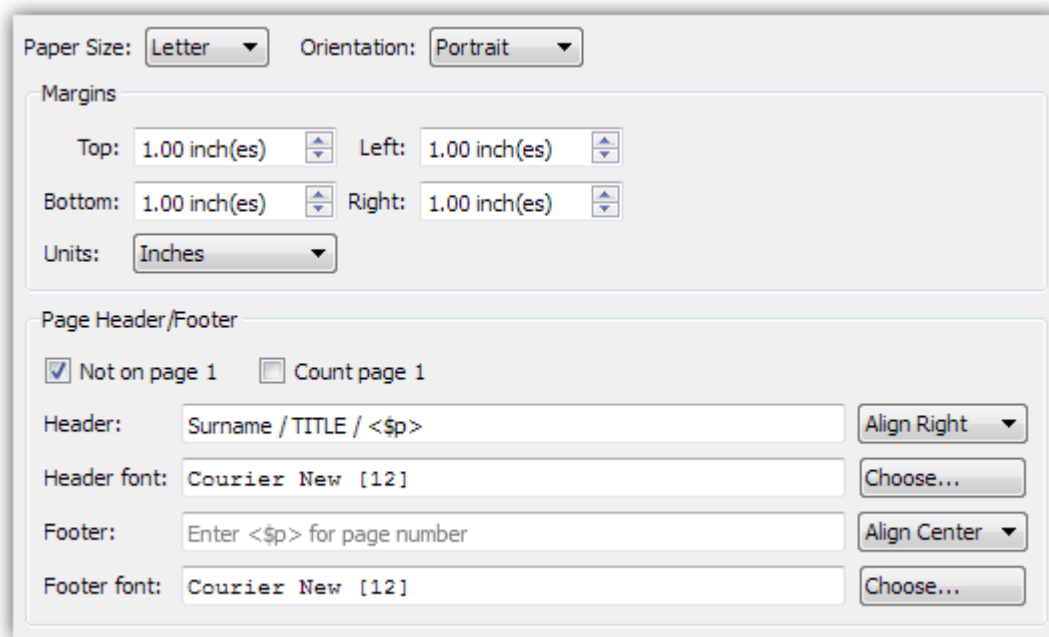


Figure 22.10: Compile—Page Settings (RTF)

not have the first page counted with the page count token by leaving “Count page 1” disabled. This will accommodate the use of a title page.

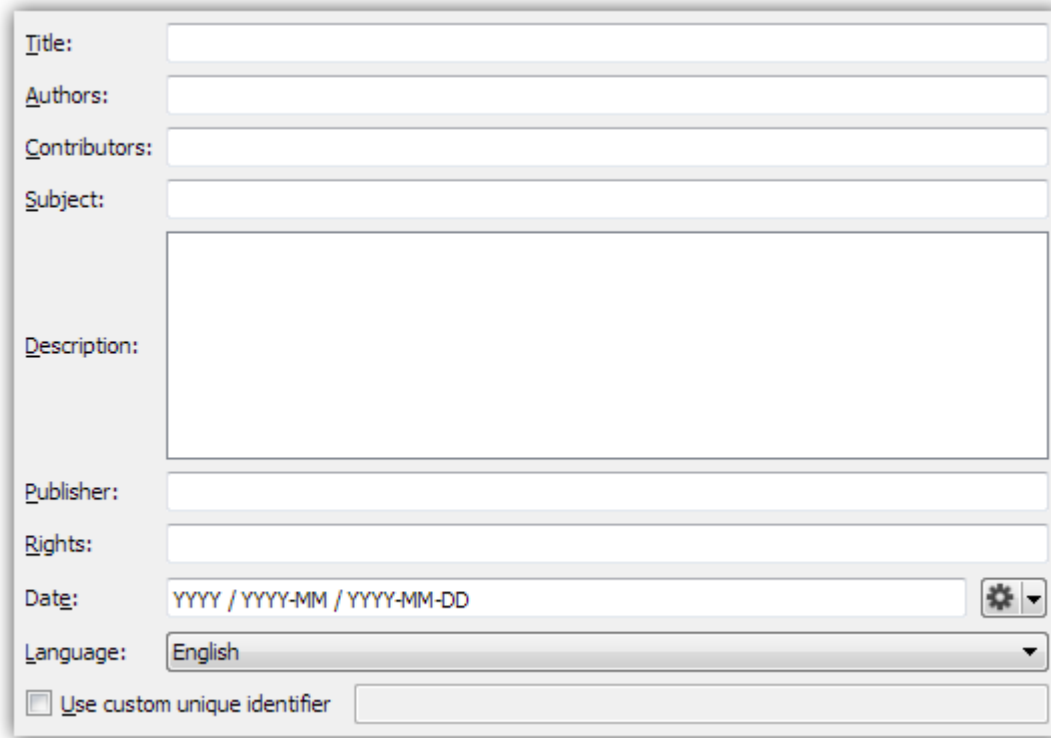
For both the header and footer, you will find the following controls:

**Header/Footer** These two fields respectively set the *content* that will appear on the page.

You may use the `<$p>` token here to specify where the page number should be printed. A common example from submission standards could be, “Surname / Book Title / `<$p>`”.

**Alignment** To the right of the content field, select the appropriate alignment for the header; whether it should be printed along the left or right side of the page, or centred.

**Header/Footer font** The font which will be used to print the header or footer. A preview will be displayed in a text field. To change the font, use the **Choose...** button to the right.



The image shows a dialog box titled "Compile—Meta-Data (ePub)". It contains the following fields and controls:

- Title:** A single-line text input field.
- Authors:** A single-line text input field.
- Contributors:** A single-line text input field.
- Subject:** A single-line text input field.
- Description:** A large multi-line text area.
- Publisher:** A single-line text input field.
- Rights:** A single-line text input field.
- Date:** A date input field with the placeholder text "YYYY / YYYY-MM / YYYY-MM-DD" and a gear icon to its right.
- Language:** A dropdown menu currently showing "English".
- Use custom unique identifier:** A checkbox followed by a single-line text input field.

Figure 22.11: Compile—Meta-Data (ePub)

## 22.14 Meta-Data

Meta-data is used to add properties to your manuscript, which many applications will display in a document properties viewer, and with some formats, will be used to add searchable criteria to the Spotlight database in the document’s properties. There are two primary meta-data interfaces available to different formats. The two meta-data panes will only appear along with the specific formats they have been tailored for, MultiMarkdown and ePub/Mobi. Despite the differences, the various panes are referred to by the same name in the option pane list.

Since, with the exception of MultiMarkdown, meta-data is not meant to appear in the content of the manuscript itself, none of the plain-text formats, or any formats that do not support auxiliary properties, will show the Meta-Data pane.

### 22.14.1 E-books

Most of the fields in the e-book meta-data pane are free-form text fields. They will be inserted into the appropriate book description fields, which will be available to e-readers to display in whatever manner they provide, and used to automatically sort the book into

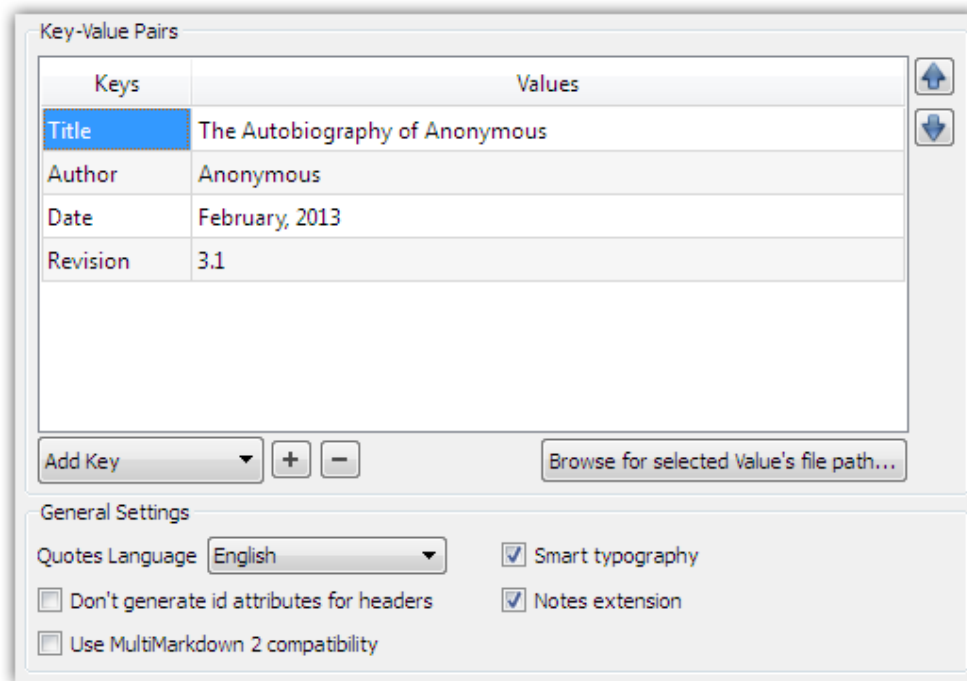


Figure 22.12: Compile—Meta-Data (MMD)

electronic catalogues. Most of the simpler e-readers, such as those embedded into specific purpose devices, will only use Title and Author, but many desktop readers and dedicated devices such as Adobe Digital Editions and Calibre will make use of as much meta-data as you can throw at it.

The three fields which you will want to be more careful with are Language Code, the custom unique identifier, and if you have multiple contributors, the Authors and Contributors fields. If you do not wish to provide a custom identifier, Scrivener will generate one for you based on the author, title, and date. If your publisher has provided a unique identifier for you, then you can check the box and type in your own.

For multiple authors and contributors, separate each author with a semi-colon to ensure that they all get registered as separate individuals in the e-book. The ePub format allows for multiple authors so that book searches can be done on each author independently. Amazon's KindleGen utility does not recognise multiple authors, so only the first name 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.

### 22.14.2 MMD

The Meta-Data pane for MultiMarkdown compilations provides a front-end to the preliminary meta-data block at the top of most MMD files. Note that what you type into the text area below the field will be printed into the compiled document verbatim. Thus, all rules pertaining to the meta-data syntax should be followed. Empty lines in this area will cause the meta-data block to prematurely abort.

Individual rows can be moved around with the arrow buttons to the right of the table. The order of meta-data fields only matters when using MultiMarkdown version 3.

This interface, being project-based, is useful for creating a general meta-data block. If your project is actually comprised of multiple documents, or if you simply prefer to type in the meta-data block by hand, you can use documents titled “Meta-Data” in the binder. Whenever a document so titled is the first document in the compile group, it will be appended to the meta-data provided in this pane. Duplication should be avoided where possible. A good strategy for multi-file projects is to use the project compile settings for general meta-data which does not change (such as your name), and then the keys specific to each output file in the “Meta-Data” documents.

For further information on the available MultiMarkdown meta-data fields, refer to the [online documentation](#)<sup>3</sup>.

To make inserting new keys easier, most of the standard meta-data keys used by the stock MultiMarkdown engine have been included in a drop-down menu below the table. Simply pick the one you wish to add, and then type in the value in the “Values” column. You can also add your own custom keys. Click the + button to insert an empty row in the table that you can fill out.

If you need to reference a file on your disk but do not know the path, you can use the [Browser for selected value’s file path...](#) button. This will print the full text path of the selected file into the value for the current row you are editing. Note that MMD uses relative paths if it can. So in many cases it is easiest to reference support files (like CSS or bibliography files) simply by name, rather than supplying the full absolute path. You would then copy these support files into the output folder after compiling.

### 22.14.3 HTML and FODT Headers

When either MMD to HTML or MMD to FODT is selected as the output format, an additional meta-data field will be supplied. These match the “XHTML Header” and “ODF Header” keys, and are simply produced here with a larger value field for your convenience.

---

<sup>3</sup>[http://fletcherpenney.net/multimarkdown/users\\_guide/multimarkdown\\_syntax\\_guide/](http://fletcherpenney.net/multimarkdown/users_guide/multimarkdown_syntax_guide/)

Additional HTML options will be provided in this area as well which make adjustments to the MMD engine.

#### 22.14.4 General Settings

Below the meta-data table are a few general output settings, both for Scrivener and the MultiMarkdown engine itself.

**Quotes language** You can set this with a meta-data key yourself, but this drop-down menu will do so for you. A number of typesetting protocols are provided for.

**Smart typography** Disables MultiMarkdown’s smart typography, which will convert ASCII compliant punctuation to typographic punctuation, such as em-dashes, typographic quotes, and ellipses characters.

**Don’t generate id attributes for headers** Disables MMD’s automatic ID generation based on the name of the header. This option only impacts HTML. Do not enable this if you are using cross-references.

(Currently non-functional)

**Use MultiMarkdown 2 compatibility** For use with legacy MMD2 installations. This impacts how Scrivener will generate image syntax, and whether or not “Format: complete” is added to the meta-data block for you.

**Notes extension** (Currently non-functional).

# Exporting

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 \(chapter 22\)](#). 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.

## 23.1 Drag and Drop Between Projects

While it is not possible to drag items directly from the binder to other programs on your computer (including the Desktop), you can easily drag items between Scrivener projects. This will retain almost all of the document's meta-data, including its notes, snapshots, and keywords. Scrivener will attempt to match labels and status, but unless both projects came from the same un-altered source this will likely produce undesired results. Items should be proofed for their label and status after copying them into the target project.

## 23.2 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 **Ctrl-Shift-X**. This will by default also export any descendants of the selected items as well. This way you can select an entire chapter and export all of its section files at once.

When exporting more than one file, Scrivener creates a folder on the disk to hold all of the exported files. Enter the name you would like to be given to this folder in the “Save As” text field, and then choose where you would like this folder to be created. From the “Export text files as” pop-up menu, you can choose to export text documents as one of the following:

- Print
- Portable Document Format (PDF)

- Plain Text (TXT)
- Word Document (DOC)
- Open Document Format (ODT)
- Web Page (HTML)
- eXtensible Web Page (XHTML)
- PostScript (PS)
- Final Draft 8 (FDX)

All media files will be exported as they are. You can also choose to export comments and annotations, meta-data, and/or snapshots; and optionally choose to strip out all notation. 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), and the meta-data (including the synopsis, label, status and so on) will be exported into separate plain text (TXT) files. The structure of folders created on disk will reflect their structure in the binder<sup>1</sup>, so in this way you can export all of your files from Scrivener for use in another application if you so wish, or as an archival backup.

## 23.3 Exporting to an Outliner with OPML

Many outliners support a common format known as OPML. In a sense it is like RTF in that it is a way for many different programs to communicate with one another, and in this case specifically to communicate outlines. 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. There is an additional convention on top of this format which allows for some outlining programs to attach “notes” to outliner headings. Unfortunately this method of attaching notes is not a standard, and so support for it may vary in terms of quality and features. The good news is that many applications these days do support a “\_notes” field.

Scrivener supports the “\_notes” method for attaching additional data to outliner headers, both on import and export. This is optional, so if you are having difficulties loading

---

<sup>1</sup>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. Another caveat is that items will be sorted (typically alphabetically) rather than using the original order in the binder. If retaining the original order is important, producing a compiled version of the document alongside the full export would be a good idea.

the OPML file in the target software, try using “Titles only” as your export option. This mode will adhere to the standard OPML specifications.

Typical usage is to select (binder sidebar, corkboard, and outliner views are all acceptable places to make a selection) a container that you wish to export (all descendant items will be included), and use the **File ► Export ► OPML File...** menu command. You may also indicate that the entire Binder should be exported. Outliners use a “root” item concept. 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 of the top-level items within it (even the Trash and its trashed items).

There are two formats available for outline import and export:

1. OPML.
2. Mindmap, a standard format used predominately by Freemind.

When you indicate that some extra form of textual data should be added to the outline, beyond the title headings themselves, they will be attached as notes. Because of limitations in the design of this convention, it is not possible to transfer rich text information between outliners using the “\_notes” method. All material will be converted to plain-text.

### 23.3.1 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.

**Note attachment options** Using the loose convention of placing plain-text notes into a “\_notes” attribute for each outline heading, you can often communicate extra information to other outlining software. In all cases, the associated data for that option will be assigned to the same binder item it came from. So if an item named “Fruit” has a description written into its index card, and you select “Include Synopses”, then in your outliner, the “Fruit” heading will contain a note field with the description from the synopsis attached.



- *Titles Only*: if attempting this gives you issues or error messages, the first option “Titles only” conforms most closely to the published specifications for OPML and should be the safest format to use.
- *Title and Synopses*: contents of any index cards, or the synopsis field.
- *Title and Text*: in this case, “text” refers to the main body text for that item, or what you have typed into it using the text editor. This will be the content of the research material, or portions of your draft as you have written them. This only includes *text* from file or folder items in the Binder.
- *Title and Notes*: uses the auxiliary Inspector Notes field for each included item. As with the main text option, the results of this will be converted to plain-text.

### 23.3.2 Limitations

- This method is intended for exporting an indented outline, rather than exporting meta-data. If you wish to export tabular meta-data lists, try Exporting Meta-Data to a Spreadsheet ([section 23.4](#)).
- Only text items (including folders, naturally) in the binder can include their content in the “\_notes” field. Research items such as graphics or PDF files can export only their titles, synopses and document notes.

## 23.4 Exporting Meta-Data to a Spreadsheet

Outliner views can be exported to formats which are suitable for loading in spreadsheet software, like OpenOffice Spreadsheet or Microsoft Excel; many databases; and other miscellaneous programs that support compartmentalised data. If you wish to export a specific set of columns, or even all possible meta-data, use **File ► Export ► Outliner Contents as CSV...** This menu item will be disabled unless an Outliner is currently the active view. 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 23.3](#)).

**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 importing 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 deselected, Scrivener will export *all* meta-data columns in the order that they appear in the [View ► Outliner Columns ►](#) sub-menu.

# Printing

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 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. For all other forms of printing, either piecemeal or in groups, the standard print mechanisms 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 highlight meta-data and synopses.

In most cases (compile, notably excluded) all you need to do to print a selection is press **Ctrl-P**, or select **File ► Print...** from the menu. 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.

## 24.1 Document Printing

Individual documents and supported media can be printed one by one. When more than one text document has been selected in the binder 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).

### 24.1.1 Text Documents Print Settings

Page margins, paper orientation, and size can be set up with **File ► Page Setup...** or **Ctrl-Shift-P**.

**Paper** Sets up the type of paper that you intend to use in your printer (most common paper formats are supplied, and if a printer with multi-tray capability is attached to the computer, you can specify an optional tray in the Source field).

**Orientation** Portrait will print a document in the standard fashion, showing text in a tall column on the page. Landscape orientation treats your paper as flipped sideways, for a short but wide column of text.

**Margins** Set the four margins, each as an offset from the physical edge of the paper. So to offset the block of text one inch in from the left side, you would set Left to 25.4mm.

This tool will not take into account your printer's physical limitations. If you want to print very close to the edge of the page in order to minimise how much paper you use, you'll need to refer to your printer documentation to see what the minimum safe margin is.

## 24.2 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.

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 press **Ctrl-P**. Also, as with corkboard printing, this will not attempt to reproduce the precise appearance of an outliner, as this would lead to situations that would be difficult to print, with very wide outliners. Instead, the outliner printing tool generates an indented list with the title and synopsis for each item in the current outline view.

Each row in the outliner will be printed in a "block" on the page, and within this block there are three primary sections in this order:

1. Title Area
2. Synopsis Area (if applicable)

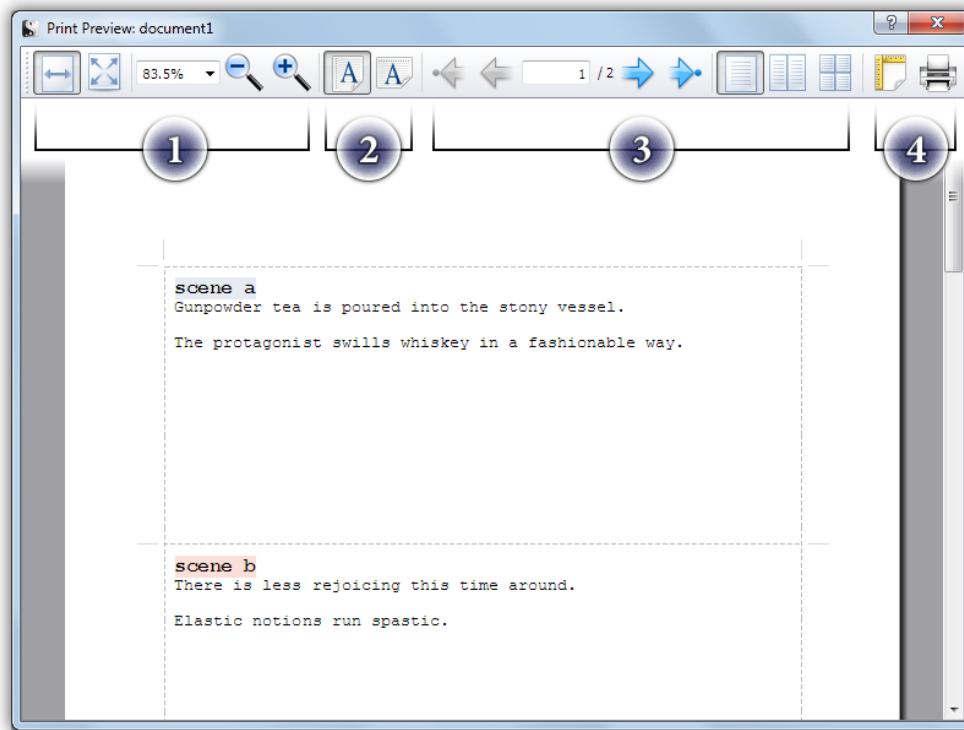


Figure 24.1: Print Preview Showing Index Card Output

If you find the outliner fails to provide the look you desire, you might consider using the compiler instead. Try starting with the provided preset, “Enumerated Outline” and customising that if necessary. Or, if you just want the data, try exporting the outliner to a spreadsheet compatible file ([section 23.4](#)).

## 24.3 Print Preview

The standard printing features discussed so far can all be previewed, prior to actually printing, by using the **File ► Print Preview...** menu command, or by pressing **Ctrl-G, Ctrl-R** on your keyboard. If you are looking for a way to preview the compiled draft, you should use the PDF compile format and view the PDF in a reader. You can then either print from the PDF reader if the result is desirable, or return to the compile interface to tweak your settings and try again.

The provided screenshot depicts a sample of what printing from a corkboard might look like, as simulated 3x5 index cards ([Figure 24.1](#)). Along the top of the window are various control buttons. Some of these will impact the print output itself, while others merely impact the display window for your convenience.

1. *Zoom display controls*: these do not impact the output, they only adjust the display settings within the window itself. The first two buttons will attempt to resize the pages to fit the window, either by width (as shown in the screenshot), or by both width and height (whichever edge is longer). You can also manually adjust the zoom percentage in the text box to the right of these buttons, or click the magnifying glass buttons to decrease and increase the zoom level respectively.
2. *Paper orientation*: select portrait or landscape orientation. This option will impact that printed output.
3. *Page display controls*: the page control features a text field you can use to jump directly to a specific page, as well as arrow control buttons for flipping through the document page by page. The two outer arrow buttons will skip to the very beginning or end of the document. The three buttons to the right of the page flippers control how many pages are displayed within the window at once. You can choose between a standard single-page view, simulated bookbinding view, and 4-page view. None of these controls will have any impact on the output.
4. *Print settings and output*: the final two buttons directly control print output. The left button will open the paper settings dialogue (which can also be access with **File ► Page Setup...**). The right-most button sends the current printout to the standard Windows print dialogue box.

To cancel the print preview window without printing, you can either click the window close button along the top, or tap the **Esc** key.

## 24.4 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 all of the options available to you during compilation will be functional. To print, open the compile interface with **File ► Compile (Ctrl-Shift-E)** and using the “Compile For:” drop-down menu, select “Print”, then click the Compile button.

Once Scrivener has finished processing your draft, you will be presented with the standard Windows print dialogue. From here, you can select from a number of software printers, designed to write a print job to a file, or send the print job to your physical printer. For example, if you have Microsoft Office installed on your computer, you’ll be able to print pages to OneNote. If you have a PDF writer installed you can print to PDF (though you might as well just use Scrivener’s PDF generator at that point).

See also:

- Read *Compiling the Draft* ([chapter 22](#)) for more details on how to compile.
- If you want to print a script, read *Printing or Exporting a Script* ([section 19.2](#)) for tips.

**Part V**  
**Appendices**



# Menus and Keyboard Shortcuts

This appendix will address each item in every menu, and will display the keyboard shortcut for it when applicable. 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 when necessary.

The menus in Scrivener are divided into various domains, in which we attempt to organise functions into them according to those domains. If you are looking for a particular feature and are not sure where it is located, then reference the table describing the menu sections ([Table A.1](#)). In this way you can try to track down a feature as you might in a topical thesaurus.

## A.1 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...** ([Ctrl-G](#), [Ctrl-N](#)) New Project creates a new Scrivener project. This will bring up the New Project and template selection window (see Quick Tour ([chapter 6](#))) which will walk you through creating a new project, which will be saved as a .scriv folder on disk and in the directory that you specify (such as Documents, or Desktop).

**Open...** ([Ctrl-O](#)) Open allows you to open a pre-existing .scrivx file on your hard disk. Note that Scrivener cannot open read-only files directly (such as those that have been backed up to a CD-ROM) and will give you the opportunity to save the project to an area where you have permission to save files.

**Recent Projects** ► Here you can select and open projects that have recently been opened in Scrivener (you can also specify in the General options tab ([section B.1](#)) whether Scrivener will reopen all projects that were open in a previous session when it is launched).

Table A.1: Menu Organisation

Menu Title	Purpose of Menu
File	Concerning the management of projects, including the transfer of information into and out of them, either manually with import & export, or automatically with backups.
Edit	If a function will work with the <i>content</i> 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 copying text, inserting an equation, checking spelling, or searching for all bold phrases in the text.
View	For adjusting how the project window looks and feels, as well as navigating (or changing what you are viewing) within the content of the project. The features here are not strictly aesthetic, and many work directly with project content itself, such as the <a href="#">View ► Go To ►</a> sub-menu. Other features are simply concerned with the control of window features, such as <a href="#">View ► Media ► Rewind</a> , or the appearance of things, as in <a href="#">View ► Corkboard Options ► Cards Across ►</a> .
Project	Settings and features specific to the project are gathered in this menu. This is also where content can be created, and is distinct from the File menu in that the File menu is concerned with creating new content in the project based off of existing data on your disk (such as importing a picture). The Project menu is for creating brand new content.
Documents	This menu concerns the management of existing documents, as well as direct navigation based off of a selection of documents. It is distinct from View menu navigation, which is concerned with more abstract navigation, “Go to this location here...”, instead of, “Open this document I have selected...”. Various aspects of a document’s meta-data and organisation can be set in this menu.
Format	Where the Edit menu concerns itself with the <i>content</i> of an editor session, the Format menu concerns itself with the <i>presentation</i> of content. In simple word processing terms, this includes such minutia as setting a range of text to bold or italic, but it also includes settings of larger impact, and the ability to add footnotes or comments to the text.
Tools	Various helpful tools (such as the name generator), and application settings are collected into this menu.
Help	Quite simply put, where you got this PDF file from most likely, amongst other resources. Additionally, setting up registration information, and other application information can be accessed from here.

---

**Close Project (Ctrl-F4)** Closes the current project and all of its associated windows.

**Close All** All open projects will be closed. If you have the start panel set to show when no other projects are open (section B.1), it will appear; otherwise Scrivener will fully close.

**Save (Ctrl-S)** Scrivener auto-saves your work so that you never have to worry about losing your writings. Projects are also saved automatically whenever they are closed. However, you can use Save to force an immediate save whenever you want.

**Save As... (Ctrl-Shift-S)** Will prompt you for a new project name and/or location. When you submit this dialogue box, Scrivener will immediately start working off of the **new** copy. If instead you want to generate a backup copy and keep working off of the original, use **File ► Backup Project To...** instead.

For more information on managing your project files, see Project Management (chapter 7).

---

**Import ►** Features 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 Gathering Material (section 11.1).

**Files... (Ctrl-Shift-J)** Import files from the file system into Scrivener. Only text 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.

**Web Page... (Ctrl-Shift-W)** Import and archive a web page from the Internet by supplying Scrivener with its URL.

---

**OPML or Mindmap File...** Convert an .opml or .mm (Freemind) file into a full outline in the binder, optionally including any note text that may be saved into the file. Note content can be imported into synopsis, text, or document notes—or stripped altogether so that only the outline structure is imported.

**Scrivener Project...** Import a selected Scrivener project into the current project's binder. All binder items will be imported, as well as some meta-data: keywords, and notes.

---

**Import and Split...** 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.

**Export** ► Provides tools for exporting elements of the binder to the file system. For more information on exporting your work, see Exporting ([chapter 23](#)).

**Files...** (**Ctrl-Shift-X**) Allows selected contents of a project to be exported as files and folders onto the hard drive. This feature should primarily be used to create a hard copy backup, or to share resources with someone not using Scrivener. Generally speaking, you will want to use Compile to export your book into one file, rather than many.

**Comments and Annotations...** Exports just the comments and annotations from the project into a single file. Optionally you can choose to export only the selected binder items.

**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 between programs.

See also: Exporting to an Outliner with OPML ([section 23.3](#)).

**Outliner Contents as CSV** When an Outliner view is active, this menu command will activate. It will export the current Outliner view in a format that is readily accessible to spreadsheet software and other programs capable of tabular display.

See also: Exporting Meta-Data to a Spreadsheet ([section 23.4](#)).

---

**Back Up** ► Functions for managing backup copies of the current project. For more information on back up strategies, see Backing Up Your Work ([section 7.8](#)).

**Back Up To...** Generates a 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 on network drives.

**Back Up Now** Triggers the automatic backup system to produce a backup immediately, using the preferences for backup location and rotation scheme. Customise how this works in the Backup preferences tab ([section B.9](#)).

---

**Exclude From Automatic Backups** Removes the current project from the automated backup system. This is useful when the current project is very large, and is causing you to have to wait for long intervals while the automatic system performs backups. Note that when this option is engaged, Scrivener's backup protection will be completely ignored for this project. Take care to manually back up excluded projects from time to time.

**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. For more information on creating templates, read Templates ([section 7.7](#)).

---

For full documentation of the printing process, read Printing ([chapter 24](#)).

**Page Setup...** (**Ctrl-Shift-P**) Accesses the page layout setup window. **Note**, these settings do not adjust Compile printing, only current document printing, though Compile can be optionally configured to use your Page Setup paper dimensions.

**Print Preview** (**Ctrl-G, Ctrl-R**) Displays the preview interface, showing you what will print if you use **File ▶ Print Current Document...** This will not be suitable for previewing the final compiled version of the draft, as no processing and formatting will be done to the text. Read more about Print Preview ([section 24.3](#)).

**Print Current Document...** (**Ctrl-P**) Prints the current text editor selection. If more than one text item is selected, they will be merged together into a single printout, much like Scrivenings mode in your editor.

---

**Compile...** (**Ctrl-Shift-E**) Compile is the standard method for producing a manuscript out of all the pieces in the Draft. This feature provides an immense degree of flexibility, and is fully documented in Compiling the Draft ([chapter 22](#)).

**Exit** Fully shuts down Scrivener. This will not present the New Project window like the **File ▶ Close All** command will. Closing Scrivener in this fashion will remember any projects you had left open, and re-open them for you, if you have "Open recent projects on program launch" active, in the General options tab ([section B.1](#)).

## A.2 Edit Menu

The Edit menu contains options related to editing. Many of the standard functions you'll find in most Edit menus can be found here, including cut, copy, paste and find, along with a large complement of Scrivener-specific features.

**Undo & Redo (Ctrl-Z & Ctrl-Y)** 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 easily go back to another document and revert changes independently of other documents.

**Cut, Copy, and Paste (Ctrl-X, Ctrl-C, and Ctrl-V)** Cut, copy and paste act exactly as they do in other applications. Note that they only operate in a text editing context, and not on binder documents themselves. See Documents Menu ([section A.5](#)) for document management commands.

**Copy Special ►** This menu allows you to copy text or binder items in a variety of ways which can apply or remove data depending on the specific menu command.

- *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 as Basic HTML:* Applies very basic, HTML 4.01 compliant code. When constructing HTML e-mails, or working with an older web site, this is what you will want to use.
- *Copy as Bulletin Board Code:* Translates most basic formatting into BBCode, suitable for forum posts. Because BBCode does not support as many formatting features as the text editor, some elements will be lost.
- *Copying Without Annotations or Footnotes:* will strip out all annotation and footnote text, while retaining all other formatting. Most often this is useful for producing “clean” snapshots ([section 15.6](#)) after an editing session. This command will remove all notation, including linked notation.

**Paste and Match Style (Ctrl-Shift-V)** This pastes the contents of the clipboard without any of its existing fonts and styles. 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.

**Select All (Ctrl-A)** Selects all of the content of the current text, lists (like the binder and the keywords lists), outliner view, table view or corkboard.

---

**Complete (Alt-=)** Manually calls up the word-completion service, regardless of word auto-completion settings. Can optionally be set to provide suggestions automatically as you type, in the Auto-Correction option tab ([section B.6](#)).

**Complete Document Title (Ctrl-G, Ctrl-M)** If you start typing in a document title from the same Scrivener project, you can use this to cause Scrivener to search your project for matching titles, based on what you have typed thus far, and suggest alternatives.

**Add Selection to Auto-Complete List (Ctrl-G, Ctrl-A)** 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.

---

**Insert ►** The Insert sub-menu provides a wide variety of elements that you can insert into text fields. These will be placed at the current cursor position. Note that some of these insert invisible characters which control the flow of your document, you may not see anything happen unless you have [View ► Show Invisibles](#) enabled.

**Image From File... (Ctrl-G, Ctrl-J)** Allows you to select an image file from disk to insert into the current text document. See [Inline Images \(section 15.5\)](#) for further detail on working with graphics embedded in the text.

Image Linked to File...

**MathType Equation** Inserts a MathType image equation at the current cursor location. Requires the active focus to be in a text editor. When inserting an equation on a line, it will attempt to match the baseline of the current text. Requires a separate installation of MathType.

See also: [Using Equations with MathType \(section 20.5\)](#).

**MathML Equation** Inserts a MathML equation at the current cursor location, which will be rendered and displayed as a graphic in the editor, at the cursor position from whence the menu command was run.

See also: [Using Equations with MathML \(section 20.6\)](#)

---

**Line Break (Ctrl-G, Ctrl-Y)** Inserts a soft-break instead of a full paragraph break. An example use: when you need to create a list within a single paragraph.

**Page Break (Ctrl-G, Ctrl-Z)** Inserts a page break within the current text. Page breaks are invisible and will only show up when printing or exporting to a fully-featured word processor, or if you have “Show Invisibles” enabled. Typically, it is best to let Scrivener handle page breaks by section, using the “Page Break Before” option in the inspector, or in the Separators compile option pane (section 22.6).

**Non-Breaking Space (Ctrl-G, Ctrl-W)** Inserts an space character which will prohibit word-wrap from dividing two or more words from each other. For the purposes of word-wrap, it will consider a sequence of words separated with non-breaking spaces as a single word.

---

**Auto-number ►** Scrivener has a number of internal counters which are represented as tokens in the text. These will be counted and printed in the desired format when compiling.

For full documentation on how to use the various counters, see Placeholder Tags (Appendix D).

**Draft Word Count ►** Inserts a token which will be calculated during Compile which indicates the word count for the entire manuscript. There are a number of rounding options available, for cases where precise counts are undesired.

**Draft Character Count ►** As with “Draft Word Count”, inserts a token for the total character count. Again, rounding options are available.

---

**Endnote Marker** This special placeholder will collect all of the endnotes in a project and place them at the location of the marker. 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 multiple streams to compile both footnotes and endnotes, this marker will not impact footnotes.

This requires endnote grouping to be enabled in the compiler. See Footnotes/Comments (section 22.12) for further detail. This method does not work with .rtf and .doc formats.

---

**Current Date & Time (Ctrl-F6)** Inserts a plain-text date-stamp at the current cursor position.

**Append Selection to Document ►** Provides a sub-menu populated with all of the documents in your project. By selecting one of these, the text currently selected will



be added to the end of that document (this can be a folder or a file). If it is your intention to move the text instead of copying it, after issuing this command, you can simply press the **Delete** key.

Use the **New...** sub-menu choice to create a new document containing the selected text. You will be asked where to place this new document in a subsequent sheet.

---

**Link...** When adding a link, this tool works for hyperlinks of all types. To create a new Scrivener Link, see below. If the cursor is not placed within a link, you can use this command to make a new one. It will be inserted into the current document at the cursor position, automatically selecting the nearest word if necessary. The sheet will provide you with several common prefix options, or optionally no prefix.

If the cursor is currently placed inside of a link (including Scrivener Links), this command will let you edit the destination for that link.

**Scrivener Link** ► Creates a new Scrivener Link to a chosen document at the current cursor position, using the linked-to document name, or if you have pre-selected text, the original text will be turned into a link and no name will be inserted into the document.

Read more about Linking Documents Together ([section 9.4](#)).

**Unlink** If the cursor is placed in the text of a link, this command will destroy it. To delete both the text and the link together, select the linked text and delete it as you would normal text.

---

**Find** As with many applications, you will find familiar tools for finding things both in individual document. This list briefly explains the various functions. For full documentation on how to use Scrivener's extensive searching features, see Searching and Replacing ([section 20.1](#)).

**Find...** (**Ctrl-F**) The standard find and replace panel. This panel only works within a single editing window. Most often this means one document, but in the case of a combined text view, it might mean many documents.

**Find Next / Previous** (**F3** / **Shift-F3**) Jumps to the next or previous matching text based on the criteria supplied in the Find panel, relative from your current cursor position. Note these can be used even if the find panel is closed.

---

**Project Replace...** Shows the Project Replace sheet, which allows you to replace text throughout the whole project. This operation cannot be undone, use with care.

**Project Search...** (**Ctrl-G**, **Ctrl-S**) Provides a shortcut for the toolbar Project Find tool. If the toolbar has been hidden, a separate window will be opened up for your convenience.

Both of these tools are quite powerful and have a wide range of options. Please read Searching and Replacing (section 20.1) for more information on how to use them.

---

**Find by Formatting...** (**Ctrl-F3**) Opens the Find by Formatting panel. This tool is quite powerful and has a wide range of options, please see Find by Format (subsection 20.1.4) for full documentation.

**Find Next / Previous Formatting** (**Ctrl-Shift-F3** / **Ctrl-Alt-F3**) As with Find Next and Find Previous, will let you jump from match to match within a single document, even without the “Find by Formatting” panel open.

**Character Map...** Loads the standard character map utility which allows full access to the Unicode character set. Use this to insert characters that are not found on your keyboard.

## A.3 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 a setting in the main options, chances are it is in the View menu.

View Modes are various ways in which you can view the presentation of multiple items in the editor. For an overview of how view modes work, see View Modes (section 5.2). When a container is selected, the following menu command names will be prefixed by “Groups as”, indicating that this will not only change the view mode, but set it for future use as well.

**Document/Scrivenings** (**Ctrl-1**) If multiple items are selected in the editor, displays the current editor as a Scrivenings text editing session, showing the text of each document as if they were a single file. Also sets your preferred view mode to Scrivenings. See also, Editing Multiple Documents (section 15.8). Note this option is only useful

if the editor is displaying documents, media will not be mixed into an Scrivenings session.

If a single item is selected in the editor, this menu item will be labeled **Document**, and switch the editor focus to its text or media view.

**Corkboard (Ctrl-2)** Display the current editor as a corkboard. If the selection is a container, its immediate children will be shown as index cards. Also sets your preferred view mode to Corkboard. See also, The Corkboard (section 12.1).

**Outline (Ctrl-3)** Display the current editor as an outline. Also sets your preferred view mode to Outliner. See also, The Outliner (section 12.2).

---

**Go To ►** Offers an alternate way to jump to a specific binder item without using the binder. If the binder is hidden, or a collection is selected, for instance, will let you navigate to other areas of the project without having to alter your work environment. The contents of this menu will be populated by the binder, or by the current Scrivenings session. Also provided are some contextual navigation and focus functions which will let you move around in the binder, from the editor itself.

- *Previous Document:* (**Alt-Shift-UpArrow**) Jumps to the previous document in the binder, according to strict outline order (ignore hierarchy).
- *Next Document:* (**Alt-Shift-DownArrow**) As with Previous Document, but selects the next document in the strict outline order.
- *Enclosing Group:* (**Alt-Shift-LeftArrow**) Will display the currently edited item in context with its siblings using the current view mode. 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, for your convenience, or expand your current scrivenings session to include the siblings and parent of the current text item.
- *Editor Selection:* (**Alt-Shift-RightArrow**) 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 item(s) from Corkboard or Outliner as text (even if they are containers). In other words, if the selection is a container, these commands will load that item in a view. **Alt-Shift-RightArrow** will *always* load the selected item as a text file. When more than one item is selected, they will be loaded as a composite Scrivening in Multiple Selection mode.

**Collections** ► Provides commands for setting visibility and options of the Collections feature. For documentation on how to use Collections, see Using Collections (section 8.4).

**Show (Hide) Collections (Ctrl-Shift-9)** Reveals or hides the collection tab interface which will appear above the binder. Note the sidebar will need to be visible for this menu option have any immediate affect, but the option will be changed in the background for you, so the next time you enable the binder sidebar, Collections will be visible.

---

**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. To freeze a search result list without converting the original collection, simply select the entire list of items and add a new collection. The selected items from the search result will be added automatically to this new collection for you.

---

**Next Collection** Select the next Collection tab in the stack

**Previous Collection** Select the previous Collection tab in the stack

The remainder of this menu will display the available collections, allowing you to switch amongst them without the tab interface open.

---

**Zoom** ► Provides shortcuts to the zoom menu, located in the footer bar for text documents. Temporarily zooms the display of the font up or down to increase legibility, without adjusting the underlying font size. This will also increase or decrease the size of a viewed PDF.

• *Zoom In:* (Ctrl=) makes the text larger.

• *Zoom Out:* (Ctrl-Hyphen) makes the text smaller.

**Enter Full Screen (F11)** Enters the full screen writing environment, switching to text editing mode if necessary. If the editor is displaying a Scrivenings session, the entire session will be opened in full screen. Note this option is not available to media and web pages, or any other view modes.

**Layout** ► The commands in the Layout sub-menu affect the main window elements.

**Show (Hide) Binder (Ctrl-Shift-B)** Toggles whether or not the binder ([subsection 5.1.2](#)) sidebar 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) Inspector (Ctrl-Shift-I)** Toggles whether or not the Inspector ([subsection 5.1.5](#)) is visible on the right of the main window. The inspector displays all meta-data for the current document, including synopsis and notes. The inspector is hidden by default.

---

**No Split (Ctrl-'); Split Horizontally ('Ctrl-+); Split Vertically (Ctrl-Shift-')** These commands allow you to choose how to view the main editor in the main window. For more information on using splits, see [Splitting the Editor \(section 14.4\)](#).

**Swap Editors** Only available when Split Horizontally or Split Vertically is selected (see above). This commands swaps the position of the two split panes. For example, so that the view on the top will become the view on the bottom and vice versa.

---

**Show (Hide) Header & Footer View (Alt-UpArrow / Alt-DownArrow)** Toggles the visibility of the header and footer bar. Both are elements of the editor and can be toggled independently for each split. See [The Editor \(chapter 14\)](#) for more information on what these do.

---

**Manage Layouts** Toggles the Layouts panel, which allows you to manage view settings by saving the current settings, or retrieving other settings you've saved in the past. See [Layouts Panel \(section 9.5\)](#).

The **All Layouts** sub-menu displays all saved Layouts so you can quickly switch between them without using the management window.

**Editor ►** Options and commands for controlling the main editor.

**Lock in Place (Ctrl-Shift-L)** Locks the editor (or split) so that no binder clicks will affect it. When an editor is locked, its header bar will turn a shade of red.

---

**Forward/Backward in Document History (Ctrl-( and Ctrl-))** Like a Web browser, the editor keeps track of everything you've visited. Using these commands you can navigate back and forth in the history. Note you can also

use the Forward and Backward buttons in the header bar. Each split keeps its own history.

When selecting a number of documents in the binder prior to entering composition mode, the history queue will be pre-stocked with these selected items, so long as the main editor is not set up in Scrivenings mode.

**Other Editor** ► Provides controls to manipulate whichever split is not currently active.

- The Forward and Backward in History commands operate just like the above commands, only in the opposing split without moving focus there.

---

**Clear Document History** Use this command to wipe out the document history for the active split.

**Media** ► Commands for controlling the playback of audio/visual media in the editor. These menus and shortcut keys make it possible to control the playback of media while working in the opposing split.

- *Play/Pause* (**Ctrl-Return**)
- *Fast Forward* (**Shift-Ctrl-}**)
- *Rewind* (**Shift-Ctrl-**{****)

**PDF Display** ► This sub-menu controls how PDFs are displayed in the editor. Most of these options should be familiar as they are common to many PDF viewers.

- *Actual Size*: Zooms the PDF to 100%, even if it is too large to display all at once.
- *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 view.

---

**Corkboard** ► Visual options for the corkboard. The settings in this menu are project specific. For more information on using the corkboard, see [The Corkboard \(section 12.1\)](#). 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.

- *Cards Across*: sets how many cards will be drawn along each row, before wrapping to a new row. The Auto-Fit option will automatically wrap cards based on whether the next will fit, much like word wrap works, and is the default.
- *Show (Hide) Pins (F9)*: toggles whether the coloured markers are displayed on the index cards. Markers take on the colours associated with the labels that have been assigned to the documents represented by the index cards. The visual style of pins can be adjusted in the corkboard option tab (section B.3).
- *Show (Hide) Status Stamp (F10)*: 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 Corkboard tab (section B.3), and the Appearance tab (section B.2).
- *Show (Hide) Keyword Colors (Ctrl-F12)*: keywords assigned to a document can be visually indicated along the right-hand side of the index card as coloured pieces of tape. The number of keywords that can be shown at once can be changed using the corkboard display options button. For more information on using keywords, see Setting Up Meta-Data (chapter 10).

**Outliner Columns** ► Displays a list of toggles which will reveal or hide the corresponding columns in the active Outliner view. These settings are stored by individual editor split, letting you have custom columns for the two views, and are saved in the project file. You can also manage columns using the arrow button above the scrollbar in the outliner itself.

See also: Managing Columns (subsection 12.2.1).

**Use Label Colour In** ► A document’s label colour (see Setting Up Meta-Data (chapter 10) for more information on labels) can be applied as a tint to various interface elements throughout Scrivener. This menu contains a list of toggles which will enable tinting to the corresponding elements. The intensity of all tinting can be universally adjusted in the Appearance option tab (section B.2).

- *Binder (F5)*: rows in the binder will have a background bar placed behind items with the associated label colour.
- *Icons (F6)*: Only the icons will be tinted throughout the project. Wherever the icon for a binder item shows up (such as in the Header bar, next to the title in Corkboard, in the binder, and so forth) it will be tinted using the colour of the assigned label.

- *Index Cards (F7)*: the entire background “paper” for index cards will be tinted using the assigned label colour. This includes the index card which appears at the top of the inspector.
- *Outliner Rows (F8)*: as with binder tinting, the background for the entire row will be filled in with the label colour.

---

**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 application 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.

- *Left/Right/Top/Bottom Editor & Binder: (Ctrl-Tab)* This menu item follows a chain of the three most common areas of desired focus, the primary split, secondary split, 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.
- *Binder*: Moves focus to the binder sidebar from anywhere in the interface, if visible.
- *Left/Bottom Editor*: Moves focus to the Left or Bottom editor, if the editor has been split. Otherwise selects the primary editor if there is no split.
- *Right/Top Editor*: Moves focus to the Right or Top editor, if the editor has been split. If it has not been split, this shortcut will not be available.

**Outline ►** Provides navigation and disclosure commands which work in both the outliner and in the binder, referred to below as the “active outliner”. Container is defined as a folder by default, but if “Treat all documents with subdocuments as folders” is enabled in Navigation preferences ([section B.4](#)), the definition of container will be expanded to include document stacks as well.

- *Expand All*: Expands all of the visible items recursively. When this command is used in the binder, it will expand the entire binder (including all non-Draft sections). When used in outliner, it only works within the context of the currently viewed portion of the binder.



- *Collapse All*: As with *Expand All*, this works the other way, closing all visible open items recursively. Note that in both cases, you can use the Option key in conjunction with a mouse click to *selectively* perform this function on the arrow (and everything beneath it) clicked.
- *Collapse All to Current Level*: Working in a similar fashion to *Collapse All*, this command will collapse the entire outline only up to the currently selected item's level.
- *Previous Container (Ctrl-Shift-UpArrow)*: Will select the previous visible container in active outliner, no matter what the depth. In this case, previous means a container higher on the list than the current position. 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 **LeftArrow** key.
- *Next Container (Ctrl-Shift-DownArrow)*: Works the same as *Previous Container*, only selecting the next visible container downward from the current position.

---

**Reveal in Binder (Ctrl-Shift-8)** This command is available the editors and any collection view in the sidebar. It will show the location of the currently edited (or selected) file(s) in the binder, opening containers as necessary to do so. This is most useful when the method you used to arrive at the current document did not involve clicking in the binder.

**Binder Affects** ► Adjusts which split binder sidebar clicks will be opened in. By default all clicks will be sent to the active editor. 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.

- *Current Editor*: default behaviour as described above.
- *Alternate Editor*: whichever editing split is *not* current active.
- *First Editor*: the left/top editor no matter which editor is active.
- *Second Editor*: the right/bottom editor split, no matter which editor is active.

**Show (Hide) Toolbar (Alt-LeftArrow)** Toggle the visibility of the main application toolbar.

## A.4 Project Menu

The Project menu addresses commands and configuration options which are specific to the active project, such as document format over-rides, custom meta-data, statistics, and more. If more than one project is open, this menu will target the foremost project window.

**New Text (Ctrl-N)** Creates a new text document below the current selection. When a folder is selected, the new file will be created within the folder. Otherwise the new file will appear as a sibling below the current selection, or the location of the document that is being edited.

**New Folder (Ctrl-Shift-N)** Creates a new folder document at the current selection. Folders will always be created as a sibling to the current selection.

For more information on the differences between files and folders, see Project Planning ([chapter 12](#)).

---

**Add Collection** Creates a new collection, automatically placing any selected documents into the collection for you. This is analogous to clicking the + button in the top of the sidebar.

**Remove Collection** Deletes the currently selected collection tab. Since collections only store a list of references to your original items, this will not remove the items as well. This is analogous to clicking the - button in the top of the sidebar. This cannot be undone.

For tricks on how best to use text and project statistics, read Goals and Statistics Tracking Tools ([section 20.2](#)).

---

**Project Targets (Ctrl-.)** Toggles visibility of the targets floating panel, which lets you track a few simple metrics in real-time as you type and edit. This can be left open while you work.

**Project Statistics (Ctrl-.)** Opens a window displaying the statistics for the current project, including the word and character count for the draft and for the current binder selection.

**Text Statistics (Ctrl-/)** Opens a window displaying statistics for the current document (and thus only available when a text document or Scrivenings are focussed).

---

**Meta-Data Settings...** (**Ctrl-Shift-M**) Displays a sheet which lets you adjust the two main document meta-data fields (Label and Status by default, but these can be re-named), as well as the custom meta-data columns for this project.

See also: Setting Up Meta-Data ([chapter 10](#)).

**Auto-Complete List...** (**Ctrl-Shift-4**) Displays a sheet which lets you manually add and remove terms to the project's auto-complete list. Depending on your auto-completion settings, this will let you add custom names, places, and workflow indicators to assist in typing out repetitive items. Modify how aggressive auto-complete should behave in the Corrections preference tab ([section B.6](#)).

---

**Show (Hide) Project Keywords (Ctrl-Shift-O)** Displays or closes the Project Keywords ([section 10.2](#)), for managing all of the keywords in use by your project or assigning them to items.

**Empty Trash...** **Permanently** discards the contents of the project trash can. You will be warned before this is done. Once contents have been deleted, there is no way to undo that action. This can also be done by right-clicking on the Trash itself in the binder.

## A.5 Documents Menu

The Documents menu contains all commands relating directly to documents, such as duplicating and splitting documents, viewing them in combination and moving them, and hoisting the binder to view isolated parts of the draft.

**Open ►** The Open sub-menu provides you with a number of ways in which to open a selected item. These menu items are relevant from the binder or collection views, as well as corkboard and outliner views, but note that in most cases, a single click, or double-click on the icon will open a file all by itself. How it does that is dictated by the settings available in the **View ► Binder Affects...** sub-menu. The corkboard and outliner can be configured to immediately open clicked files too.

**in Editor (Ctrl-Shift-Return)** This command mirrors many of the basic interactions you can take to open an item in the currently active editor, such as clicking on an item in the binder or double-clicking on its icon in the Corkboard or Outliner views. This menu item is available to provide a keyboard shortcut for these actions.

**in Other Editor (Ctrl-Alt-Return)** This will open a new split if necessary, using the previous split orientation used.

---

**With All Subdocuments** ► If the selected item has subdocuments (this can be determined on the corkboard if the index card appears as a stack of cards; and in the outliner if it has an arrow beside its name) you can choose to open it as a corkboard either in the current editor or the other split. This command is essentially a macro for selecting the container and all of its children at once, producing a flat corkboard in Multiple Selection mode. While this view cannot be used to make edits to the structure (you'd want to use the Outliner for that), it is a handy way of viewing all cards downward from a certain point.

- *On Editor Corkboard:* Replaces the current editor view with the selected item's contents. This is similar to pressing **Shift-⌘-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 split editor to open the item, instead of replacing the current view, creating a split if necessary.
- 

**in External Editor (Ctrl-F5)** This command is available on any type of item that is not a text or folder document. It will open the file using the default editor for that type of file. An example would be opening a PDF in Adobe Reader. Opening items this way allows them to be edited in their native applications, and the results of those edits (if Scrivener can display the file type) will be visible upon saving that file.

**Original URL in Browser** When used from an archived web page that you have imported, this command will take you back to the original URL using your default web browser (thus leaving Scrivener).

---

**Snapshots** ► Provides commands to manage an item's snapshots, make new ones, and commit large-scale snapshot actions on selected items or even the entire project. For more information on how to use snapshots, see [Using Snapshots \(section 15.6\)](#). Note that in some cases these shortcuts will not be available. Generally this happens when the selection includes items which are not text items, or the text of the item(s) are empty.

**Take Snapshot (Ctrl-5)** Available when viewing any text or folder document, this command will take a snapshot of the current text and store it for later use.

When more than one item is selected, all selected items will be snapshot at once.

**Take Snapshot With Title (Ctrl-Shift-5)** As above, but will prompt you for a descriptive title prior to taking the snapshot, which will be displayed next to it for future reference in the snapshot table.

When more than one item is selected, the supplied title will be applied to each snapshot taken for all selected items.

---

**Show Snapshots** Will reveal the Snapshots pane in the Inspector, revealing it if necessary.

---

**Duplicate** ► There are two different methods you can use to duplicate the selected item(s).

- *with Subdocuments:* (Ctrl-D) This is the default method in most applications. The entire item (including all of its children if it has any) will be duplicated.
- *without Subdocuments:* (Ctrl-Shift-D) 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.

#### A.5.1 Set Selection as Title

**Set Selection as Title (F2)** Available when text has been selected in the main editor, the title for the current document will be replaced with whatever text has been selected.

**Move to Trash (Shift-Del)** As you no doubt guessed, this moves selected items to the project's trash. It will use active selections in the Outliner or Corkboard, as well as in the binder sidebar.

---

**Split** ► Provides a couple of methods for splitting the current document into two different documents. 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).

- *at Selection:* (**Ctrl-K**) simply create a new document with all text below the current caret position. Text selections will be ignored, and instead the whole paragraph starting from the beginning of the line containing the selection will be split into the second document.
- *with Selection as Title:* (**Ctrl-Shift-K**) works the same as above, but in this case the currently selected text will be used to title the new document that is created, rather than requesting a new document name after pressing the shortcut. The split will occur at the beginning of the selection, thus leaving the selected text in the new document.

**Merge (Ctrl-M)** The opposite of splitting, merge will take two or more documents and combine 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 meta-data 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 meta-data result.

---

**Group (Ctrl-G, Ctrl-U)** Takes two or more selected items and creates a new folder, placing the selection within that folder. You will be given the option to name that folder after invoking the command.

**Ungroup (Ctrl-Shift-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 destroy the original container, allowing you to move the ungrouped items back in, if it was made in error.

**Move To ►** Provides keyboard shortcuts for moving the item around in the outline in two dimensions, as well as completely displacing it to another spot in the binder.

The first portion of this menu will be populated with all of the items in the binder. The currently selected item(s) will be moved beneath the chosen item as subdocuments. Note that if you select a text item as the target, it will become a file group as it now contains children. Items that have been used recently will be displayed in the top-level of this list, beneath any favourites (if applicable). \* *Left/Right/Up/Down:* Use **Ctrl-** with the appropriate keyboard arrow key to move items around in the outline. Left and Right will promote or demote the item in the hierarchy; Up and Down will change its order in context with its siblings.

**Add to Collection** Displays a list of all standard collections in the project. The currently selected item(s) will all be added to the collection you choose. This command is also available in the binder contextual menu.

- *New Collection*: in a single move will create a new collection, opening the tab interface if necessary, and assign the currently selected document(s) to that new collection. This is equivalent to clicking the + button in the collection header bar, though more convenient if you use the menu and realise while there that what you thought was a collection has not yet been actually made.
- The remainder of this menu will be used to display existing arbitrary collections. Dynamic collections will not be displayed as they cannot be modified directly.

**Sort** ► Sort acts on the children of a selected container, and sorts items alphanumerically, either in *Ascending* or *Descending* order. Note that this a **permanent** change that cannot be undone. Additionally, you can choose to shuffle all of the children randomly.

---

**Convert** ► Provides tools for converting the document's type, writing mode, or format defaults.

- *Convert to Folder (File)*: Converts the selected item(s) to folders or files. Note that these two 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. This can also be done by right-clicking on an item in the binder.
- *Formatting to Default Text Style*: If the font and ruler styles of the document do not conform to the application or project defaults, you can use this command to use those to reset its appearance. Read more about [Resetting Formatting \(subsection 15.4.3\)](#).

## A.6 Format Menu

The Format menu contains all commands that deal with formatting the text of an individual document. It contains all of the commands for changing the styling, creating tables and lists and controlling the ruler, along with commands for highlighting and setting the current text colour. Additional Scrivener specific tools for formatting the text can be found here as well, including various forms of notation.

**Font** ► Displays commands for adjusting spans of text formatting; useful tools for controlling the character-level appearance of your text. Fonts family, variant, and size need to be set with the The Format Bar ([subsection 15.4.2](#)).

- *Show Fonts*: (**F4**) access the system-wide standard font palette, which will change the font you type in from that point forward, or alter the existing font of a selected range of text.
- *Bold/Italic/Underline/Strikethrough*: all of these operate in the same fashion by either modifying the future behaviour of typing if there is no selection, or toggling the selected text between the various styles. Note that not all fonts support these styles. If it appears that one of these is not working, check your font for proper variant support.
- *Outlined*: (**Ctrl-;**) changes the style of the font to outlined instead of filled.
- *Bigger*: (**Ctrl->**) increases the size of the font by one point.
- *Smaller*: (**Ctrl-<**) decreases the size of the font by one point.
- *Baseline*: (**Ctrl-G, Ctrl-UpArrow** and **Ctrl-G, Ctrl-DownArrow**) adjust the baseline height of the text, most commonly referred to as superscripting and subscripting.
- *Text Color*: provides several built-in text colour options, as well as access to the custom colour window, and the option to remove any colour assignments made on the selected text. These can also be accessed from the default Format Bar.
- *Copy/Paste Font*: (**Ctrl-Shift-C / Ctrl-Shift-Y**) copies the character attributes (anything in this menu) from the current cursor position. Use Paste Font to apply these settings to other selections of text.

**Text** ► Displays commands for controlling the paragraph-level formatting of your text. Paragraph alignment, as well as spacing (which is not address in this menu) can also be controlled with the The Format Bar ([subsection 15.4.2](#)).

- *Align Left/Center/Justify/Right*: controls the text alignment of the current paragraph, or all paragraphs falling within a selection range.
- *Writing Direction*: toggle between Left-to-Right and Right-to-Left writing styles.
- *Copy/Paste Ruler*: (**Ctrl-G, Ctrl-C** and **Ctrl-G, Ctrl-P**) copies the ruler and paragraph format settings (including indents and tabs) from the current paragraph. Use Paste Ruler to apply these settings to other selections of text.



- *Spacing...*: accesses the advanced paragraph and line spacing palette, which can also be found on the Format Bar, off of the line-spacing sub-menu.
- 

**Table** ► The initial command, “Insert Table...”, adds a default table into your document at the cursor position unless you are already in a table. In both cases it will also open a floating configuration window containing tools for adding and manipulating tabular data in the text editor. Tables can be set up with a given number of rows and columns, but using the options in this window, you can add or remove either to fit the constraints of your data later on.

This menu contains convenience commands for the management of cells, rows, and columns. They are also accessible via the Table sub-menu of the contextual menu.

**Lists** ► Provides basic list styles. These features are also supplied via the list button on The Format Bar ([subsection 15.4.2](#)).

---

**Scriptwriting** ► (**Ctrl-4**) Although Scrivener is not a dedicated script writing application such as Final Draft or CeltX, the Scriptwriting submenu does make it relatively straightforward to create first drafts which can be later fine-tuned in these applications. For full documentation on this, see Scriptwriting ([chapter 19](#)).

---

**Show (Hide) Ruler** (**Ctrl-Shift-R**) Toggles the visibility of the tab-stop and page margin ruler for the active editor. Note that all formatting tools have been moved from this ruler to the format bar.

**Show (Hide) Format Bar** (**Alt-RightArrow**) Toggles the visibility of the character and paragraph formatting bar, which provides quick access to many of the most common types of formatting tools.

---

**Highlight** ► (**Ctrl-Shift-H**) The highlight menu lets you place a background highlight behind the selected text, much like using a highlighter marker on paper. For more information on how to use highlights, see Text Colour and Highlights ([section 17.4](#)). These options can also be found on the Format Bar.

---

For more information on annotating and footnoting your text, see Annotations and Footnotes ([chapter 17](#)). Note that while the menu system refers to these as footnotes, it is possible to export them as endnotes during compile.

**Comment (Shift-F4)** Attaches a comment to the selected text (or previous word), making a new linked comment in the inspector pane. Useful for keeping notation text out of the editor.

**Footnote (Shift-F5)** As with Comments, adds a linked footnote style comment to the inspector pane. Useful for keeping footnote information out of the editor.

**Inline Annotation (Ctrl-Shift-A)** Converts the selected text into an inline annotation, or toggles the writing mode to annotation mode. Useful for placing notes right in the editor, alongside the relevant text.

**Inline Footnote (Ctrl-Shift-F)** Converts the selected text into a footnote, or toggles the writing mode to footnote mode.

---

**Convert ►** The convert submenu provides useful tools for the manipulation of text and notation features.

**To Uppercase** Converts the selected text to all uppercase characters.

**To Lowercase** Converts the selected text to all lowercase characters.

**To Title Case (Ctrl-6)** Converts the selected text to title case, which takes into account common English grammar conventions.

---

**Quotes to Smart/Straight Quotes (Ctrl-“ & Ctrl-Shift-\\)** Converts between typographic “curly” quotes and straight inch and foot marks, as are necessary for some plain-text formats and scriptwriting.

---

**Multiples Spaces to Space (Ctrl-Space)** Useful for cleaning up a document that has been typed up with multiple spaces in between sentences and so forth.

---

**Inline to Inspector and Inspector to Inline Conversions** These commands allow you to convert inline annotations to linked Inspector comments, vice versa, and footnotes. These commands work on the entire current document.

**Options** ► Provides options which impact the display or behaviour of the text editor.

- *Show (Hide) Invisibles*: (**Ctrl-Shift-\\**) Toggles visibility of hidden control characters, such as paragraph breaks, tabs, spaces, page breaks, and so on. A few control characters do not have associated special characters
- *Ghost Notes Mode*: (**Ctrl-Shift-7**) Ghost mode fades out the intensity of inline annotations and footnotes in the text editor, making it easier to read around them. When a note is clicked into, it will temporarily brighten to facilitate editing, returning to “ghosted mode” when you are done. Disable this option to have all notes appear at full intensity at all times.
- *Typewriter Scrolling*: (**Ctrl-G, Ctrl-T**) Toggle a feature which keeps the currently edited line of text in the middle of the screen, vertically centred. This setting is global, but will impact composition or the regular editors independently, depending upon the current mode.

**Bibliography/Citations...** (**Ctrl-G, Ctrl-O**) If a citation manager has been set up in the General tab (section B.1) of Scrivener’s options, this command will launch or activate the set manager, facilitating the process of selecting a citation and paste it back into Scrivener using whatever system that application provides.

## A.7 Tools Menu

**Spelling** (**Ctrl-G, Ctrl-Q**) Shows the main spell checking panel (subsection 15.2.2), which can be used to walk through the document looking for errors, allowing you to select from a list of potential replacements or edit the original, among other functions.

**Writing Tools** ► Access to handy Internet searches, based on the currently selected text phrase, as well as the The Name Generator (section 20.3).

---

**Scratch Pad** (**Ctrl-Shift-Zero**) Displays or closes the floating Scratch Pad Panel (section 11.3); useful for collecting information while using other applications.

---

**Customize Toolbars...** Edit the various toolbars in Scrivener. For details on using this, see Customising Toolbars (Appendix C).

---

**Options...** (F12) Accesses the main application options window. For a complete list of all available options, see Options (subsection B.2.1).

## A.8 Help Menu

The Help menu has useful tools and links for learning Scrivener, as well as a way to check for updates on our server, or submit bug reports.

**Scrivener Manual** A quick link to the PDF that you are likely reading. 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](#)<sup>1</sup>.

---

**Interactive Tutorial** If you have not yet gone through the tutorial (you should!) this menu command will walk you through the process of creating the tutorial Scrivener project. 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.

**Video Tutorials** A handy hyperlink to a ten minute introduction to Scrivener as a video presentation.

---

**Support** A handy hyperlink to our web site's support page. Here you can download the latest copy of the PDF in US Letter or A4 (or the project used to make it), find contact e-mail addresses, links to our forums and wiki, after-sales support from our vendor, eSellerate, and more.

**User Forums** A handy hyperlink 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.

**Literature & Latte Home** A handy hyperlink to our [home page](#)<sup>2</sup> which provides easy access to everything else we offer on our web site.

---

<sup>1</sup><http://www.literatureandlatte.com/support.php>

<sup>2</sup><http://www.literatureandlatte.com/>

**Scrivener Home** Link to the main [Scrivener page](#)<sup>3</sup>, where you will find useful download links for updates, case studies, links to share Scrivener with Twitter and Facebook, and more.

---

**Registration...** When you purchase the application, use this menu item to copy and paste your registration information from the e-mail that you will receive from Literature & Latte. In case you have lost the original e-mail 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. Not infrequently, invoices can end up in spam collection folders, so be sure to check your e-mail application and make sure it did not get tagged as spam, first.

**Check for Updates...** You can use the Check for Updates menu item to check to see if there is a newer version of Scrivener available for download (all updates within a major version number are free). This tool lets you check for updates and then download and install them all without leaving the application.

---

**Purchase Scrivener...** This option will only appear if the software has not yet been registered. During the trial period, it will be available so you can easily get to the web site and secure a serial number for your copy.

**Keep Up to Date...** Presents a form which you can use to submit your e-mail 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 “literatureandlatte.com” in your spam filter, prior to submitting this form, as you will be sent a confirmation e-mail 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 e-mail address will never be shared with third-parties or sold to marketing lists.

---

**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 the red **X** button to close the window when you are done.

---

<sup>3</sup><http://www.literatureandlatte.com/scrivener.php?platform=win>

# Options

Like most applications, Scrivener installs in a “ready to use” state. The default options have all been carefully selected to present a cohesive and useful writing environment. Options should thus be considered a preferential, and perhaps even slightly more advanced way of using Scrivener. This manual has described the application in reference to its defaults, but there are a great many ways to tweak the interface and behaviour of the application to suit your preferred working style. This appendix will thoroughly document every single option item, as presented to you in the Options window itself and its tabs. This way, if you wish to seek clarification on a particular item, you can find it using the manual’s list of option tabs, and then within that section, sub-sections will separate groups of preferences in accordance with their grouping in the application.

Because Scrivener is a multi-project application, options are split up in different areas so that you can supply new projects with some starter settings. Once these projects have been created, they will maintain their own individual settings from that point forward. One step beyond that, there are a sub-set of options which apply at the document level, and each document will diverge from the project and application options.

Not all options work in this “cascading” fashion. The ones that do will be indicated as such in their notes. Many of the settings that are wholly project specific make no appearance in the application option tabs at all. A good example of this is whether or not labels are displayed on index cards. By default they are, but you can turn them off for a project—and it will remember that setting. There is, however, no way to change that behaviour globally<sup>1</sup>.

All of the project specific options are located in menus, and most of these are in either the Project menu or the View menu. This section will not cover these specifically, though some might be mentioned in passing where applicable. Instead, this section strictly focusses on the application options window and its tabs. For information on specific menu options, see the appendix, Menus and Keyboard Shortcuts ([Appendix A](#)).

To bring up the Options window, click on Options in the Tools menu (or press **F12**). Along the side you’ll find a sequence of icons. Each of these tabs leads to a sub-section

---

<sup>1</sup>Unless of course you consider project templates, which you should view as a form of “secondary” settings. Project templates are how you can adjust how Scrivener acts with a brand new project.

of the application options, and the rest of this appendix will deal with each of these subsections in turn. If you close and re-open the options window later on a single session, the last tab you were looking at will be preserved.

At the bottom of the window are buttons for setting the options, cancelling, and saving or loading your options from an external file. To save your changes and leave the Options window, click the **Ok** button. The **Apply** button will update Scrivener with your settings and save your options, but leave the window open so you can make further changes. The **Cancel** button, as you might suspect, leaves the Options window and discards any changes you have made up until the window was opened, or when you last pressed **Apply**.

The **Manage** drop-down menu is used to save your options to an external file, or later to load them back in. Use this to transfer your settings from one computer to another, or to periodically back up your options.

Finally, the **Defaults** button will do a “factory reset”, returning all options to their default states. This cannot be undone, but you will be warned after clicking the button.

## B.1 General

The General tab governs the application’s basic behaviour, its integration with other programs, and import and export of material.

### B.1.1 Startup Options

**Open recent projects on program launch** Reopens recently opened projects whenever Scrivener is launched—thus saving you the effort of opening what you were last working on.

**Show start panel when there are no projects open** Toggles the display of the start panel, which lets you create new projects, open existing projects on your drive, or select from a list of recently worked on projects. Note you can always completely quit the application by using the **File ▶ Exit** to avoid seeing this window.

**Show full project path in the title bar** When enabled, the full path (the precise location of the project, including its parent folders) will be displayed along with the name in the title bar at the top of the project window.

**Automatically check for updates** If this is checked, whenever Scrivener is launched it will 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 checked, you can still check for updates by choosing **Check for Updates...** from the Help menu.

The drop-down menu beside this option governs how frequently Scrivener will check for new versions. If you want to stay on top of the latest updates, “Daily”

is generally good enough, but if updating frequently is disruptive to your work habits, you can set this down to check by week or month.

### B.1.2 Saving

**Save after a period of inactivity** Sets how long Scrivener waits to save changes after you stop using it. Scrivener automatically saves changes made to projects. So that it is not constantly saving—and to avoid any slowdown this might cause—when any changes are made to a project, Scrivener takes note and waits for a suitable time to save them. It then waits until you stops interacting with the program for more than than two seconds (by default). So long as you do not resume working, Scrivener deems it a good time to auto-save the project without the save process interrupting what you are doing. You can change the period Scrivener waits to auto-save with this setting. If you increase the period significantly, be sure to use **File ► Save** to regularly save changes to the disk.

When you are first starting out with the program, you might want to keep an eye on the save indicator and make sure that your working style fits in with the defaults. If you find the that the project goes for long periods of time without saving, you might want to reduce the interval to one second. If it is saving in the moment you start back up, try increasing it to three.

**Automatically name untitled text documents upon saving** When enabled, text documents that have been left untitled will acquire the first few words from the document. This will not trigger until you have added a second line, or typed around one complete sentence into the first line. This way, your documents do not get titled until the wording has settled down a bit.

### B.1.3 Bibliography/Citations Manager

**Bibliography/Citations Manager** Scrivener can integrate with several popular bibliography management programs. Use this option to select your favourite from the Applications folder. The reset button will clear your choice.

**Pro Tip:** This merely links which application will launch with the **Format ► 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.1.4 Scratch Pad Notes Location

**Notes location** Your scratch pad notes are stored on the hard drive using RTF files. This option lets you choose where those files will be automatically saved. By default,



they will be stored in your user folder, under a special folder that Scrivener will create for them.

In secured working environments, you might wish to alter the location of this to reside in an encrypted area of the drive if necessary.

### B.1.5 Language

**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 for Scrivener, you can use this setting to manually adjust the language.

Use the **Reset** button to return the interface to the Windows system default.

## B.2 Appearance

Most options relating to the appearance of Scrivener can be modified here. In a few cases, things which technically modify appearance but are more related to the specific function they fall within, will be located in the dedicated tab for them. A good example are the Corkboard appearance controls.

### B.2.1 Options

General and miscellaneous appearance options.

**Show subdocument counts in the binder** Beside each container in the binder, a number will be displayed showing how many documents it contains. This number is recursive, meaning that it will not only count its immediate children, but any of the descendants beneath those children as well.

**Alternate outliner row colors** Draws alternating background colours behind rows in the outliner. When this is turned off, the background will be a solid shade. You can define the actual colours used in the section below.

### B.2.2 Colors

Custom interface colours are chosen using a selection tree sorted by category and then type. To alter the default colour for the desired interface element, open the tree for its major category, and then the item itself from the indented list. Click on the colour well in the right column to assign a new colour, or click **Use Default Color** to reset the option to the application default. As with the rest of the options window, when you click the **Apply** button, the program will adjust to show your settings, so it is often beneficial to set

up the program so that it exhibits the area you wish to customise, then adjust the colours while previewing how it will appear.

**Note:** Some categories have identical settings from other categories, such as both the Editor and Full Screen sections. Where duplication exists, refer to the original mention.

The Full Screen “background” colour options also allows a texture to be applied to it. When you see the button **Choose Texture...**, you can set this particular element of the interface to use a tiling graphic for its background instead of a colour. Many tiling graphics can be found on the Internet for free. It is best to choose textures that are relatively small, as larger textures can slow down the interface or cause the computer to run out of RAM. To clear a texture you no longer want to use, just click the **Use Default Color** button.

Both the “Editor” and “Full Screen” sections have a special function which can be accessed by clicking on the category instead of the individual interface item. A button called **Copy from Editor ► Full Screen** will appear. This button copies all matching items from one mode to the other.

For a full list of all options, refer to the Appearance Colour Settings ([Table B.1](#)) table.

### B.2.3 Fonts

Options for setting application fonts. If you wish to adjust the editor fonts, visit the Formatting tab ([subsection B.5.1](#)). This interface acts in a similar fashion to the colour selection interface already described. Instead of a colour well, each element will have its own font selector instead. A preview of the font will be displayed in a box, and below that the **Select Font** button opens up a font picker. Click the **Use Default Font** to reset the font for this element to the application default.

**General Settings**   • *Menus & Windows:* the interface font used to display the application.

• *Document Notes & Project Notes:* default font for the document and project notes, in the inspector pane. This is only a default, it will not change the font of text already entered into the note fields, so it is safe to use this if you have styled text in your notes.

• *Binder:* font used to display the names of items in the binder sidebar (including collections).

• *Outliner:* font used to display all textual information in the outliner.

• *Comments:* font used within the Inspector sidebar comments.

• *Inspector Footnotes:* as above, only with sidebar footnotes instead.

Table B.1: Appearance Colour Settings

Category	Interface Element	Description
General	Binder Background	Background colour for the Binder
	Outliner Background	Background colour for the Outliner
	Outliner Alternate Background	Background colour to use for alternating rows
	Snapshot Text Background	The “paper colour” for the snapshot text area. Note this will apply both to snapshots shown in the inspector and snapshots displayed in the main editor
	Document Notes Background	Document notes editing background, or “paper colour”
	Project Notes Background	As above, but for project notes so that the two can appear distinctive from one another
	Recent Search Background	Background colour for the search results collection and its tab
Editor / Full Screen	Text	Default display text colour; only displays text in this colour that does not already have a colour applied to it
	Page	Main editing background, or “paper colour”
	Links	Internal links will be coloured according to this setting
	Visited Links	Recolours visited hyperlinks (not Scrivener Links) after you click on them
	Invisible Characters	Text colour used to draw special symbols depicting invisible characters
	Text Selection Background	Text colour when it has been selected
	Annotation Text	Text will be recoloured to this when it is an annotation
	Ghost Annotation Text	When <b>View ► Ghost Notes Mode</b> is on, inactive annotations will take on this colour unless they are selected
	Footnotes Text	Text will be recoloured to this when it is a footnote
Footnotes Background	The shading used to fill in the footnote bubble	

Table B.2: Appearance Colour Settings Cont'd

Category	Interface Element	Description
(Cont'd)	Footnotes Background	Background shade which will be used to fill in the footnote marker
	Ghost Footnotes Text	Same as with Ghost Annotated Text above
	Ghost Footnotes Background	Alternating shading to be used when a footnote is ghosted
	Search Text Selection	Highlight colour used to illuminate which portions of the text matched a search
	Search Text Selection Border	The bubble border used in above highlight; set to the same to use a non-bordered look
	Find Text Selection	Highlight colour used to illuminate matching text when using <a href="#">Edit ► Find ► Find...</a>
	Find Alternate Text Selection	When set to a different colour, the highlight will pulsate between these two
	Find Text Selection Text	The text colour will be changed to this temporarily in the find match. Useful when using a dark highlight.
	Background	Only available for full screen, sets the colour for the area to the sides of the virtual paper.
Index Cards	Index Card Divider	The heavy border that separates the index card title from its synopsis
	Index Card Lines	The line rule colour used in the synopsis area; set to the background colour to eliminate them
	Index Card Status Stamp	Base colour which will be used to render the stamp text across the face of the card. The opacity of this can be adjusted in the Corkboard tab
	Index Card Border Color	Outline colour used to draw a border around the entire card in the Corkboard (does not apply to the index card in the Inspector)
	Index Card Background Color	Paper colour to use for the index cards. Has no effect when index cards are being tinted by labels
Scratch Pad	Editor Background	Paper colour for the notepad editor.
	Editor Text	The text colour that will be used in the notepad editor.

**Corkboard settings**   • *Index Title*: title font for index cards. This impacts both index cards in the corkboard and inspector.

- *Index Text*: synopsis text font. This impacts both index cards in the corkboard and inspector.
- *Index Text Small*: alternate font to be used when the “Small Text” option is enabled in the corkboard display settings.
- *Status Stamp*: used to render the status stamp, which can be optionally printed on the face of index cards in the corkboard behind the synopsis text.

**Scratch Pad** The *Scratch Pad Editor* font setting will adjust the display font for typing in new notes. Since notes are stored in rich text, this will only impact notes you create from this point on out. Existing notes’ formatting will remain untouched.

## B.3 Corkboard

Provides options and settings which will impact the way the corkboard functions and appears.

### B.3.1 Options

**Allow dropping dragged items onto cards** 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.

**Empty space double-click will** Adjust the behaviour of what should happen when double-clicking in a blank area of the corkboard.

- *Do nothing*: ignores all double-clicking on the corkboard background.
- *Create a new card*: follows a common behaviour amongst diagramming applications, where you can indicate the position of a new card by double-clicking with the mouse.
- *Opens the parent corkboard*: will 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. The keyboard shortcut for this is **Ctrl-⌘-R**, and will remain in effect even when this option is not selected.

### B.3.2 Appearance

**Corners** Choose between cards with rounded corners, or cards with square edges.

**Label indicator** Choose between an colour marking on the corner of the card, or a graphical pin, to indicate label colour. Both will be tinted according to the colour of the label in use for that index card.

**Corkboard background** Lets you customise the standard corkboard background. Select from “Corkboard pattern”, “Beige Graph Paper”, or “Slate Graph Paper” to use one of the provided background patterns; “Custom color” to choose a background colour using the colour well to the right of the pop-up button; or “Custom background...” to select your own image file to use to fill the background. For better performance, use smaller, tiling textures rather than large images.

**Status stamp opacity** Allows you to set the opacity of status stamps on the index cards within the corkboard. The stamp, unlike a real one, is drawn beneath the text, so even at full opacity it will not obscure synopsis text. This only impacts how boldly the stamp will be drawn.

**Index card shadows** Sets the intensity of the shadows around the cards and borders of the corkboard view. Slide all the way to the left to disable shadows.

**Label tint opacity** Define how much blending Scrivener should use between label colour and the underlying icon or background colour. Moving the slider right toward “High” will cause icons to be coloured more vividly; moving the slider left toward “Low” will cause the effect to be more subtle.

## B.4 Navigation

The navigation settings provide settings to alter how Scrivener reacts to the various actions you can take to navigate around in your project. There are numerous ways of doing so, from clicking directly on items, to using buttons to scroll through documents like a flip-book, to using menus to jump directly to a specific place in your book, and more.

In nearly all cases, unless otherwise indicated, navigation options are globally applied; that is, no matter how you select an item in the interface, these options will apply.

### B.4.1 Folders and Files

**Treat all documents with children as folders** When this is checked, any document that has sub-documents will act like a folder and will open in whatever view mode folders are set to use (the last group view mode selected). If this isn’t checked, normal text documents or media files that have sub-documents (any type of document can act as a “container” for other documents in Scrivener) will be opened in single text mode.

**Include enclosing group in combined scrivenings** When clicking on a folder (or file group, if “Treat all documents with subdocuments as folders” is enabled), 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, or document stack text, then disabling this will remove the empty entry at the top of the Scrivenings session.

**Show folder text when selected from search results** Overrides default behaviour of opening folders (and if “Treat all documents with subdocuments as folders” is set, file groups as well). When this is enabled, all container style search results will reveal their text when clicked on in the search results, rather than revealing their children in accordance with the selected view mode. This option is enabled by default, because 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 is because of its text.

#### B.4.2 Return Key Behavior

**Return ends editing synopsis in corkboard and outliner** By default, the **Return** key will submit the edits being made to the synopsis in index cards and the synopsis outliner field. When disabled, the Esc key can be used to stop editing, or clicking anywhere outside of the card with your mouse.

#### B.4.3 Media

**Rewind when paused by...** Use the slider to set how many seconds Scrivener should rewind the media stream back, when using Auto-Rewind ([subsection 14.3.3](#)). This preference is not used unless the media player has been set to automatically rewind on pause.

## B.5 Editor

The options for the main text editor control how text is displayed within the editor itself, including its formatting.

#### B.5.1 Default Main Text Attributes

These settings determine how *new* documents will be configured. They will not adjust the formatting of documents you have already created, in any projects, so you needn't worry about these settings destroying your existing formatting.

To reset a document to the application options, use the menu command, **Documents ► Convert ► Formatting to Default Text Style**.

The top half of the tab is dedicated to setting up the default settings for new text documents (files and folders). You will have access to the full standard ruler settings, as well as font controls via the **A** button to the far left. All changes you make here will be immediately previewed against the provided sample quotation. These settings will impact all of the new text documents you create, so use the special effects with care.

See the Appearance tab ([subsection B.2.2](#)) to change the colours used in the editor.

### B.5.2 Editing Options

**Use block insertion point** Adjust the width of the insertion point from its default of 1 pixel. This will dramatically increase the visibility of it, however with variable-width fonts (like Calibri), this setting can produce odd results as the cursor will most often be wider than the character following it. It is best used with a fixed-width font like Courier or Consolas.

**Use font hinting** Most fonts have advanced metrics for handling small characters, referred to as “hinting”. Hinting simplifies the shape of characters as necessary, to improve legibility. Most fonts will be handled correctly, but if you find your preferred font is illegible this setting might improve its appearance.

**Typewriter scrolling (and for full screen) in new projects** Enables this feature at the application level for either mode selected. Typewriter scrolling keeps the text you are currently editing in the middle of the window. There are two checkboxes, one for enabling it by default in the main editor, and the other for full screen.

**Ruler Units** Lets you choose the units (centimetres, inches, picas, or points) that should be used in the main editor’s ruler (available by hitting **Ctrl-Shift-R** when the editor has the focus or by selecting Show Ruler in the Edit menu.

**Ruler snap every** Use the scroller buttons to increase or decrease the step value of ruler adjustments. When this is set to anything but “No snap”, indent and tab markers will stick to increments at the level of resolution you supply here.

**Editor Margins** Adds spacing between the text editor and the surrounding view. This impacts both horizontal and vertical margins at once. Vertical margins will be drawn at the very top and bottom of the editor, so they might not be visible if you have scrolled into the middle of a long document.

### B.5.3 Default Zoom

Controls for setting the default zoom level of the three main text editing areas: Editor, Document Notes, and Project Notes. The editor setting is a preset for new projects.



Once a project has been created, you will need to adjust the zoom independently, using the footer bar in the editor. The settings for the two types of Notes will impact all projects immediately, and there is no other way to change the zoom setting for these fields.

To change the default zoom setting, use the first drop-down menu to select the area of the interface you wish to adjust, and then use the second drop-down to select a zoom size.

135% is the default zoom setting used by Microsoft Word and other popular word processors, and so will render your text at a familiar size. Zoom size will not impact your print or compile settings, only the visual display of fonts as you write.

## B.6 Corrections

The Auto-Correction tab contains optional typing aids, such as typographic enhancements (em dashes, smart quotes, etc), word completion, and control of the spelling engine.

### B.6.1 Spelling

**Check spelling as you type** Displays red underlining for words that are not found in the built-in dictionary, as you type. This setting immediately impacts projects.

**Correct spelling errors as you type** Will correct common spelling errors and typos automatically. For example, “teh” will be corrected to “the”.

**Dictionary** Click the **Select** button to choose your preferred dictionary. You can only load one dictionary at a time for spell checking. If you cannot find your language in the supplied list, click the **Download** button. Additional dictionaries are available, and can be downloaded and installed into Scrivener with a single click.

**View Personal Word List** As you add new words to the built-in dictionary, you can review your additions by clicking this button. A window will appear with a list of all the words you have added. Add new words with the **+** button, and remove mistakes with the **-** button.

### B.6.2 Auto-Capitalization

**Fix capitalization of sentences** Will fix letter case issues if a lowercase letter follows a period and a space.

**Capitalize ‘i’** Will automatically capitalise the letter ‘i’, if typed by itself.

### B.6.3 Word Auto-Completion

**Suggest completions as you type** Rather than waiting for you to use the **Edit ► Complete** menu command (**Alt-=**), words from the project's Auto-Completion ([section 15.7](#)) list 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 **Enter** 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.

### B.6.4 Substitutions

Enables a wide array of commonly used symbols and typographic conventions, by detecting when their use is appropriate. Not all text entry areas are capable of using substitutions, but they will always be available in the main text editors.

If you intend to use Scrivener as a *de facto* plain-text editor to compose syntax, like HTML, LaTeX, MultiMarkdown, or DocBook, you will probably want to disable all of these features as most syntax formats either handle this for you, or require specific instructions for typography.

Most of the options are self-explanatory. Those requiring explanation follow:

**Use smart quotes (” “)** Will convert inch and foot characters into typographic quotes (also referred to as “curly” quotes) as you type, according to your system language settings.

**Replace double hyphens with em-dashes (—)** To aid in typing in em-dashes, this option will substitute two hyphens (--) as you type.

**Replace triple periods with ellipses (...)** The special single-character ellipses will be used instead of three full stops, to improve readability.

**Disable smart quote, em-dash and ellipsis substitution in script mode** The three options above this one will be ignored in script mode.

**Enable Additional Substitutions** Turns on the custom substitution engine. A large number of common text-to-symbol substitutions have been added for your convenience. You can add your own, or remove ones you do not like, by clicking the **Edit Substitutions** button.

## B.7 Keyboard

The Keyboard options tab is where you modify application shortcuts, or hot keys, for individual menu items in Scrivener. If you find a built-in shortcut to be awkward to press on your particular keyboard, or you just don't like a particular arrangement, you can modify them all here.

To find a shortcut either navigate through the tree view to find the menu, sub-menu or even sub-sub menu wherein the command is found, or use the "Filter" search bar at the top to rapidly located a menu title, if you know its name. You can search for any of the text that you see in the table, so feel free to search for the descriptive name, actual menu name, or even its shortcut. This can thus serve as a convenient reverse-look up if you pressed a shortcut by accident and don't know what it did.

To change a shortcut, select its row in the table and note that the "Key Sequence" field toward the bottom will be filled with the currently assigned shortcut. Click into this field and then type in a key sequence to capture it. The capturing mechanism will continue to capture subsequent keystrokes, or "chords" as they are sometimes called. This way you can group like commands into a common cluster. This requires you to tap in the first part of the shortcut, and then release the keys and follow it up with subsequent rapid keystrokes to invoke the command. If the shortcut you used conflicts with another one in the list, it will be highlighted in red. If it is black, then you are good to go. Clicking the **Reset** button will return the shortcut to the application default.

The **Import...** and **Export...** buttons at the bottom of the table let you manage shortcuts collectively. The import button has a few templates available that you can choose from which will attempt, as best as possible, to match other applications. Those who work with the Macintosh version of Scrivener, for instance, could select a template that more closely matches the hotkey arrangement on the Mac (though of course since modifier keys differ, it is not possible to a complete match). Also available are templates for Microsoft Word and OpenOffice. You can load templates from a file, which is a useful way of restoring your settings from a backup, or copying your favourite shortcuts from one computer to another.

The export button has two options. One lets you export a shortcut template for use with the aforementioned import feature. The second option will create a human-readable text file which is suitable for printing out as a reference.

**Limitations:** At this time, not all of the menu commands have been supplied to you in this list. While you can modify most of the menu commands that already have a pre-defined shortcut, most that do not have one already will not be listed. This limitation will be addressed in future versions of Scrivener.

## B.8 Import and Export

**Import supported file types only** By default, Scrivener will allow you to drag file types into the binder<sup>2</sup> even if it cannot internally display them. You will still be able to load them using external applications. This option prohibits importing unsupported file types. If any have been imported into projects in the past, they will remain.

**Show warnings for import restrictions** If you are already familiar with Scrivener's import limitations involving various file types, then you can disable the warning that appears whenever importing them. This will let you turn that warning back on.

**Convert HTML files to text** When importing HTML pages from your drive, or saved web files from your disk, they will be converted to RTF. This provides the advantage of easy modification to the content, at the expense of losing the original web page layout (which in some cases can be of benefit, if all you want is the article, and not the advertisements, say).

**Use Microsoft Word or Open Office for doc and docx conversions** If one of these software packages are installed, enabling this option will result in a cleaner and more precise rendering of the document. It will also require external libraries to be used, and so is thus a bit slower than the standard converters. In most cases, the standard converters will be fine. Since leaving this option on has been known to cause unnecessary file usage, it is best to turn it on while needed and then disabled afterward.

**Default URL import type:** When importing web pages directly off of the Internet, using **File ▶ Import ▶ Web Page...**, the specified import method will be used by default. You can however override the default from within the import dialogue box on a per-import basis:

- *Dynamic Web (Embedded Browser):* this is a **non-archival** method. It will display the web page in an embedded browser, loading it from the Internet whenever you click on it. Useful for storing references to pages where changes are frequent and salient, or to operate as a built-in web browser.
- *HTML (Text Only):* imports the full original web site layout, only sans the images. This is the best format to use if you are interested only in the text, as it will save resources. This will archive the content locally so that even if the original web page disappears, you will always have access to it.

---

<sup>2</sup>You can only ever drag text files, or folders with text files, into the Draft

- *Image (Browser Quality)*: imports a snapshot of the full web page as a graphic.
- *PDF Document*: will convert the web page to a PDF files, which should retain much of its original layout and graphics.
- *Plain text*: no formatting of any kind will be retained, and the layout will be linearised into paragraphs. This is the best format to use when you just want to strip the article out of the page clutter for your research. Web pages are typically encoded in UTF-8, which is also safe to use for older web pages encoded in simple ASCII. It is very rare that the Windows system encoding will be used on a web site, since web sites must remain viewable on many different types of computers and devices.

**Encoding for plain text** When saving or importing plain-text files, the encoding (most frequently the concern is whether it is Unicode or not) can be an important difference between garbage or a clean file. If you unsure of what encoding your text files use, experiment with this option until you get the right result. Text files you've written in Notepad will most likely be saved in the Windows system encoding. UTF-8 is generally the best selection for non-2byte languages, that you've acquired off of the Internet or from modern Unicode enabled text editors. It is also backwards compatible with older text files in most cases, and is the new industry standard moving forward.

**Invalid character handling** If a character cannot be recognised by the importer, it will be handled according to the choice made here:

- *Replace invalid characters with ?*: the diamond query icon will be used to show where these invalid characters were in the original document. If you see a lot of these, you might want to try using another encoding.
- *Remove invalid characters*: only use this option if you are having problems importing a file under any encoding and do not feel that anything important is being communicated by the invalid characters.

## B.9 Backup

Scrivener has a fully automated backup system. The options in this tab will control the various aspects of it.

**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.

**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.

**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.

**Back up on each manual save** Will create a backup copy whenever you manually save a project with ⌘-S. Note this is not the same as Scrivener's auto-save feature, which by default is triggered after every two seconds of idling. Backups will only be triggered if you select the menu command manually.

---

**Compress automatic backups as zip files** If you are storing your backups on a network drive, or Internet synchronised service like DropBox, this option will not only save space 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.

**Use date in backup file names** In addition to a numbering token, which Scrivener uses to keep track of the rotation (defined below), the date and time of the backup will be included in the file name, making it easier to sort by recency in Windows Explorer. It also means each backup filename will be unique, which could be of use with an external backup system; old backups will not overwrite when using the OS's native back-up features.

**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 rotate through the copies, overwriting the oldest available. 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 at least three or four days of backups available. To disable the limiter entirely, select "Keep all backup files".

---

**Backup location** The default location stores your backups in your AppData folder, under Local\Scrivener\Scrivener\Backups. 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.

# Customising Toolbars

Both the main application toolbar and the Format Bar ([subsection 15.4.2](#)) can be customised by adding or removing buttons and organising them with separators. The window is divided into three sections. The left section contains all of the available tools that can be placed into the toolbars. On the upper right side, you can select between showing the list of tools for the Format Toolbar, or Main Toolbar. The list of selected tools are shown in the list below this selector.

Use the left and right blue arrow buttons in the centre column to move tools in and out of the toolbar. The up and down blue arrows will move the selected tool up or down in the list (which corresponds to left and right in the horizontal toolbar as shown on the screen).



---

# Placeholder Tags

Table D.1: Statistics Placeholder Tags List

Code	Description
Statistics	
<\$wc>	Gets replaced during the Compile process with the total word count of the text currently being compiled.
<\$wc50>	... Rounded to the nearest 50 words.
<\$wc100>	... Rounded to the nearest 100 words.
<\$wc500>	... Rounded to the nearest 500 words.
<\$wc1000>	... Rounded to the nearest 1000 words.
<\$cc>	Gets replaced during the Compile process with the total character count of the text currently being compiled.
<\$cc50>	... Rounded to the nearest 50 words.
<\$cc100>	... Rounded to the nearest 100 words.
<\$cc500>	... Rounded to the nearest 500 words.
<\$cc1000>	... Rounded to the nearest 1000 words.
Page Numbers	
<\$p>	When used in the header or footer, gets replaced with the current page number.

---

Table D.2: Auto-Number Placeholder Tags List

Code	Description
Auto-Numbering	
<\$n>	Gets replaced with Arabic numerals during the Compile process. The number is incremented each time a <\$n> tag is encountered in the text, so “<\$n>, <\$n>, <\$n>” would become “1, 2, 3” in the compiled text.
<\$sn>	The same as <\$n> but intended to be used for sub-numbering. The count restarts each time an <\$n> tag is encountered. Thus, “<\$n> (<\$sn>, <\$sn>), <\$n> (<\$sn>, <\$sn>)” would become “1 (1, 2), 2 (1, 2)” in the compiled text.
<\$r>	Gets replaced with lowercase Roman numerals during the Compile process. The number is incremented each time a <\$r> tag is encountered in the text, so “<\$r>, <\$r>, <\$r>” would become “i, ii, iii” in the compiled text.
<\$R>	Gets replaced with uppercase Roman numerals during the Compile process. The number is incremented each time a <\$R> tag is encountered in the text, so “<\$R>, <\$R>, <\$R>” would become “I, II, III” in the compiled text.
<\$w>	Gets replaced with numbers as lowercase words (using the current language settings) during the Compile process. The number is incremented each time a <\$w> tag is encountered in the text, so “<\$w>, <\$w>, <\$w>” would become “one, two, three” in the compiled text.
<\$t>	Gets replaced with numbers as title-case words (using the current language settings) during the Compile process. The number is incremented each time a <\$t> tag is encountered in the text, so “<\$t>, <\$t>, <\$t>” would become “One, Two, Three” in the compiled text.
<\$W>	Gets replaced with numbers as uppercase words (using the current language settings) during the Compile process. The number is incremented each time a <\$W> tag is encountered in the text, so “<\$W>, <\$W>, <\$W>” would become “ONE, TWO, THREE” in the compiled text.
<\$hn>	Gets replaced during the Compile process with hierarchical numbering based on the level of the document in which the tag occurs relative to the Draft folder or compile group (depending on the current compile settings). So occurrences of the <\$hn> tag in the second document in the Draft folder may get replaced with the number “2”; occurrences of the tag in the third subdocument of the eight document in the Draft folder may be replaced with “8.3”.
Restarting auto-numbering streams	
<\$rst>	Place <\$rst> immediately before any of the auto-numbering tags to restart the numbering. So, for instance, “<\$w>, <\$w>, <\$w>, <\$rst><\$w>, <\$w>” would become “one, two, three, one, two” in the compiled text.

# Tips and Tricks and Trouble

## E.1 Getting Support

As a registered user of Scrivener, you are eligible to receive technical support for any enquiries you may have, from problems to general questions. You are also invited to participate in our discussion forums, where a vibrant community of other Scrivener users can often offer advice and help. For all avenues of support, please visit our main [support web page](#)<sup>1</sup> for details on where to proceed and which e-mail address to use.

---

<sup>1</sup><http://www.literatureandlatte.com/support.php>

# Project Bundle Format

A Scrivener project is stored on your computer as a cohesive and integrated set of standard folders and files. This internal structure is designed to be logical and accessible, in the rare case that you need to salvage material out of your project when Scrivener is not otherwise available. On a Macintosh, the operating system will hide the fact that this is a folder from you, and display it as though it were a single file. Since this is not a feature of the Windows operating system, the contents of the project will always be visible to you. If you are using a Mac, you'll need to right-click on the project file and select "Show Package Contents". The remainder of this section will discuss the contents of the .scriv package as though it were a normal folder.

The entire folder structure is considered to be a part of the project, and should never be separated from the rest. Consequently, it is often best to use an archive method to back up your project, such as Zip compression. This increases the portability of the project in convenience and size, and protects it from accidental damage. This should be done by zipping the container .scriv folder (or package).

On the flip side, you shouldn't use the .scriv folder as a place to store material. While this theoretically won't hurt anything, we can't guarantee that it will remain safe there, as this entire folder and its contents are curated by the software. If you wish to organise external material along with your project, we recommend creating a folder to contain both the .scriv folder and whatever other documents you wish to store. This warning is naturally more pertinent to Windows users.

At the top-most level, the folder will be named according to the project name you chose, and have a ".scriv" extension added to it. Beneath that level, the .scrivx file is the master file which defines the structure of your project. Every single entry in the binder, your keywords, your print settings, all of these and more are saved into this XML file.

A new project will have anywhere from two to five folders at this top level, depending on which features have been used in the project.

- Files: where most of your actual data will be stored. Here you will find all the pieces of your draft, your research material, imported media, and so forth.

- **Icons:** if custom icons have been saved into the project, they will be stored here. This folder will be created as necessary.
- **QuickLook:** on a Mac, this provides information to the Finder's Quick Look preview function. It gives you a quick overview of the project's contents in an indented outline format. Windows does not use this folder, and unless you've opened the project on a Mac, you won't see it.
- **Settings:** project level settings and appearance settings files will be saved into this folder, as well as computer specific information used to lock the project while you are editing it.
- **Snapshots:** this optional folder will be used to store snapshots, and may not exist until snapshots have been created within the project.

## F.1 The Files Folder

This folder contains the following items:

- **binder.autosave:** a redundant backup of your primary .scrivx file, at the top level. If the primary file becomes corrupted or goes missing, this can be used in an attempt to recover the project. A new copy is created every time Scrivener auto-saves.
- **binder.backup:** a third level of backup protection, this file is only created after you successfully open a project, ensuring that it is a valid copy of the primary .scrivx file.
- **Docs:** this folder stores all of your binder data, from draft sections to imported material. If it is listed in the binder, it will be in this folder, named according to its internal ID number. You can cross-reference these ID numbers by reading the .scrivx file, if necessary.
- **ProjectNotes:** a folder which stores any project notes you have created. Notes will be stored in RTF files, and the index.xml file defines any custom properties they may have.
- **search.indexes:** this file is used to optimise the search speed of Scrivener. It contains a complete plain-text copy of nearly every string in the project.
- **user.lock:** will only appear while the project is open (or in the unfortunate case of a crash wherein the software never got a chance to remove the file). This stores a key that identifies your computer. If you try to open the project a second time on another computer, you will be warned that it appears to be open already.

- `version.txt`: a simple marker for letting Scrivener know what version of the project format this is, so that it can remain compatible with future versions of the software.

## F.2 The Settings Folder

This contains all of the settings which are specific to the project. Scriptwriting, compile, and overall general UI settings such as whether you had the interface split, and what they were viewing, all the way down to whether you're tinting icons with the label colour and so on. If the UI has become damaged for some reason and the project refuses to load, or loads in a malfunctioned state, it can sometimes help to reset the UI by removing the configuration file for it. This will return your project to a factory default state in terms of visual and workflow settings *only*. It will not harm any data. Make sure the project is closed, first. To reset the UI, delete the `ui.ini` file in this folder.

## F.3 The Snapshots Folder

The snapshots folder is where all of your old snapshots are stored. A new sub-folder will be created within it for each item in the binder that has snapshots. The folder contains an XML file with some information about the snapshot (such as its descriptive name and time stamp), as well as any link information that existed when the snapshot was taken. The snapshots themselves are standard RTF files, and so can be accessed in an emergency if necessary.

The presence of these folders are the one thing that Scrivener uses to determine if an item in the binder has snapshots. If you delete the folder, it will no longer have snapshots. Thus if you are intending to distribute the `.scriv` file for whatever reason, and wish to wipe out all of the old working data, it is safe to delete these folders on a one-by-one basis, or indeed the entire Snapshots folder itself to totally clean out the project of all old versions. You will probably only ever want to do that on a copy of the original project.

# Legal

Scrivener and this manual are copyright © 2006–2013 by Literature & Latte LLC. All rights reserved. Literature & Latte are not affiliated with, or endorsed by the following companies and their products or services:

- Microsoft Office (and all products therein); Microsoft Windows
- Apple Pages; Numbers; Mac OS X and all utilities mentioned therein
- Dropbox
- Simperium Simplenote
- DenVog Index Card for iPad
- Hog Bay Software; PlainText; WriteRoom
- OpenOffice.org (and derivatives NeoOffice & LibreOffice)
- Final Draft
- Movie Magic Screenwriter
- CeltX
- Mariner Montage
- EndNote
- Bookends
- Sony Reader
- Amazon; Kindle, and KindleGen tool
- Calibre
- Adobe Acrobat Pro; Adobe Reader

- RedleX Mellel
- Nisus Writer Pro
- Papyrus
- Aspose
- Twitter
- Fountain
- Freemind

All products above and otherwise mentioned in this text are trademarks of their respective owners.

Literature & Latte take no responsibility for any text lost through synchronisation. As with all synchronisation processes, you should ensure you back up regularly and double-check that the right documents are being synchronised.

Visit our web site for a copy of the [licence agreement](http://www.literatureandlatte.com/licence.php)<sup>1</sup>. By using this software you are agreeing to be bound by the terms of this licence.

---

<sup>1</sup><http://www.literatureandlatte.com/licence.php>