

compile that looks different from what you've been writing, but in most cases there will be no visible change. Naturally, if you *require* a script format that doesn't conform to the standard screenplay, make this option unchecked, or you will lose all of your custom formatting in the compile (Scrivener's copy will of course remain untouched).

Break dialogue and action at sentences Use this option to adhere to the standard of keeping action and dialogue sentences together, rather than breaking them up between pages. If a sentence would have ordinarily been split, it will instead be moved entirely to the following page.

Include revision colors from Preferences Your preferred revision colours, which can be set in the Editing: Revisions preference tab ([subsection B.3.3](#)), will be supplied to Final Draft's revision palette, maintaining a consistent revision system between the two applications.

Summaries and Script Notes Fonts The final two options allow you to set fonts for the indicated features within Final Draft. Do note that if you intend to share the FDX file with Windows colleagues, you may wish to change the default Summaries font to Helvetica, Arial, or something else that is commonly available.

[Return to chapter](#) ↗

24.4 Separators

In real-world terms, separators represent an easy and formulaic way of inserting separation between important elements in your draft. A few common examples would be a page break between chapters in a PDF, a section break in an eBook, or even a simple “#” or “* * *” between scenes in a novel. Separators in Scrivener can be inserted either broadly as a default (all folders should have a page break inserted, for example) or very specifically as part of the *role* or function of a section layout (such as an asterism between text for a “Scene” layout).

The separators panel is broken up into three main parts ([Figure 24.8](#)):

- a) **Default Separators:** these two settings broadly adjust how separators will be inserted in the draft. They will be used for section types that are not mapped to any layouts, and for all layouts by default. Defaults are how we want to treat items in general, perhaps even regardless of what type of item it is in the binder (chapter, scene, preface, etc.).
- b) **Section Layouts:** each layout, as defined in the Section Layouts compile format pane ([section 24.2](#)), can be set to override the defaults and do their own thing. This capability is useful when the layout is meant to serve a specific role that includes separation—like a Part break in a larger book.

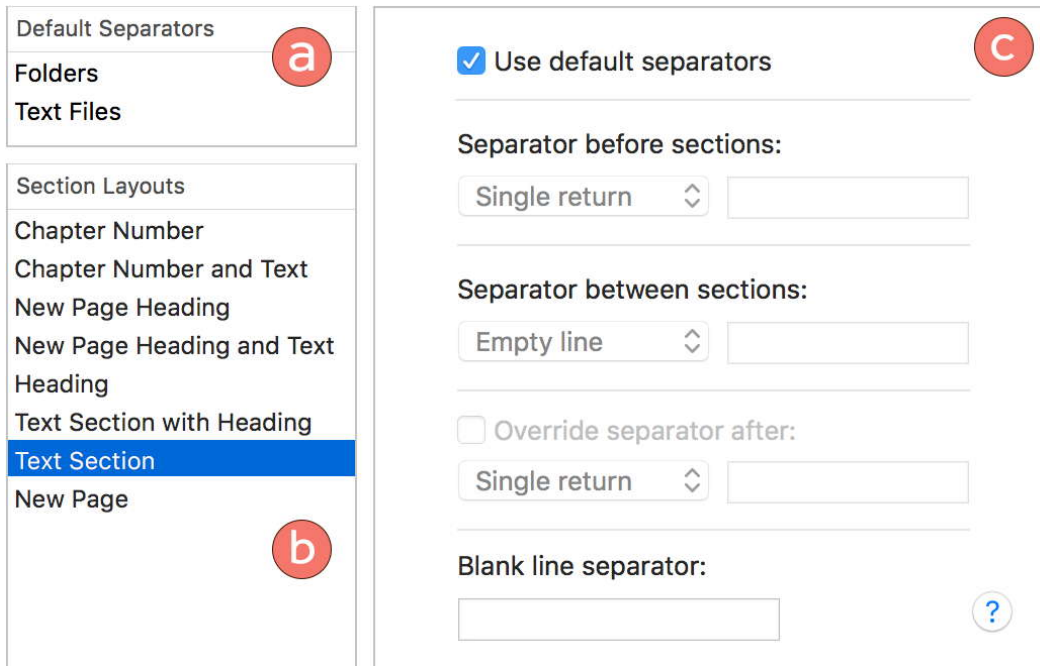


Figure 24.8: Separators can be defined by section layouts or as broad defaults.

- c) *Separation Settings*: when clicking on any of the entries in the above two lists, their applicable settings will be loaded into this area on the right.

A few good examples of where and why a Layout might override default separators can be found in the default blank “New Format” settings, as depicted in the figure. We have the “Text Section” layout selected, which acts in accordance with the global defaults—but you may notice that if you disabled the **Use default separators** checkbox at the top, it would insert an empty line between other “Text Section” items—in effect acting like a scene in a typical novel. The “New Page” layout directly below that one in the list is set up to override defaults by default (whew), in that it’s entire purpose is for inserting a page break separator.

24.4.1 Managing Layouts from the Separators Pane

You may at times find you need to create a new layout on the fly to accommodate a special form of separation. In the lower left hand corner of the Section Layout list are a pair of + and – buttons. Section Layouts can be managed from this pane:

- When adding new layouts it works in the same fashion as the Section Layouts pane would—select the layout you want to duplicate and then click the +. You may of course need to further tweak it in the Section Layouts panel after adjusting its separators.
- Use of the – will fully delete the layout from the compile format. Use with care.

- Layouts can be renamed right in this list as well, by double-clicking on their names. Click elsewhere, or press **Esc** to confirm your changes once you've edited the title.

The Separators pane is available to all formats except FCF and FDX.

24.4.2 Separator Types

The simplest way to use and think of separators is of having them inserted above the section layout that uses them, thus placing it in between the preceding chunk of text and the current one. This is the usage we referred to before, where a common (and default) behaviour is to insert a page break before all folders. If you add a folder and put some files into it in your draft, it will automatically act like a major section break when compiled.

There are three other options for where or how separation should be handled. It would be easiest to describe how they work together with a few examples. If you wish to play along with the what we will be looking at in the screenshots, you will find a demonstration project called “8-compile_separators_demonstration.scriv” in the Extras Pack ([Appendix F](#)), and consult the help file at the top of the binder if you require any further explanation of how the project is set up. For the sake of visual clarity, we'll use custom separators, which allow us to insert arbitrary text between sections.

In [Figure 24.9](#) we see two different forms of separation being employed, as well as a third option that transforms blank lines in the text editor to match the form of separation we prefer between sections otherwise. The options that are in use:

Separator before sections As referred to before, this is the simplest form of separation to think of. The orange chunk of text has been set to “Page break” as its separation type. We don't see the effects of that here however because it is the very first item to be compiled. This is an important exception to keep in mind for this separator option: it must be *separating* the item from something in order to insert a separator.

For the sections coloured in lavender, we have added a visible custom separator. Since there is a space between the first lavender chunk of text and the orange chunk of text, we see that separator inserted there.

Separator between sections When the section type of the item preceding the current item is the same then this separator option will be used. We see the asterism symbol inserted between the two lavender chunks of text because they are of a like kind. If one of these chunks of text had been a different *type* of document (like the orange chunk), then the “- Before Section -” separator would have been used instead.

Blank line separator Available only to formal section types, rather than as a global default to files or folders, this setting will *transform* blank lines found

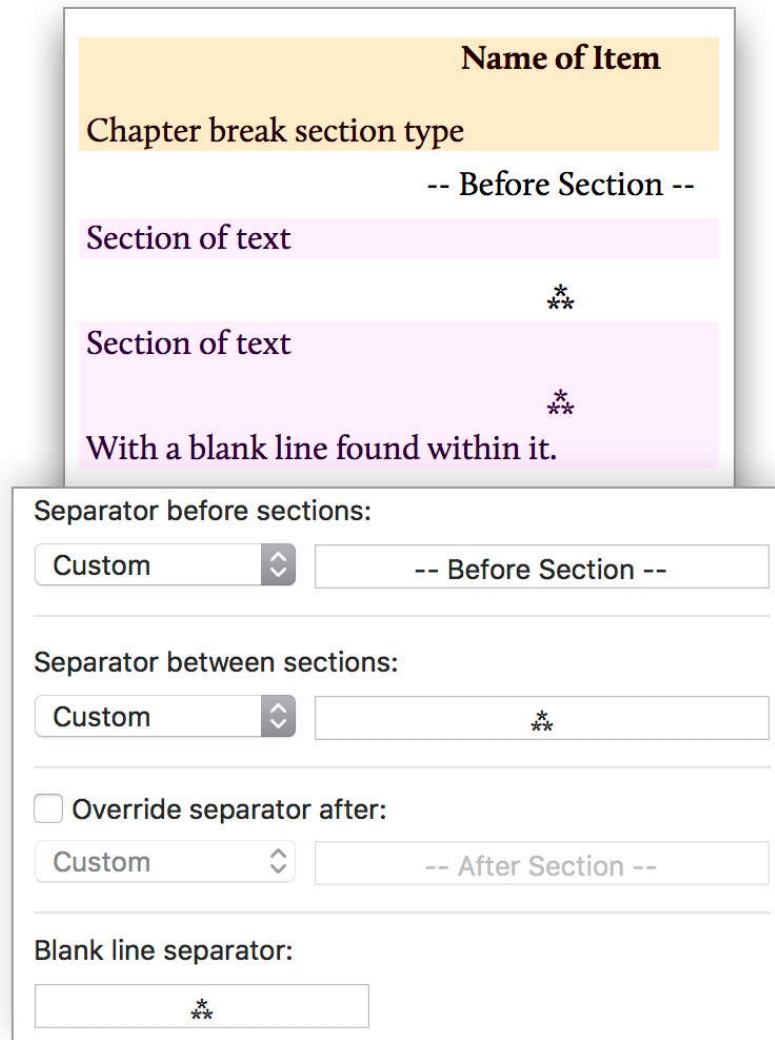


Figure 24.9: Separators can change depending on the items around the chunk of text that is inserting the separator. Compiled output colour-coded for clarity.

within individual chunks of text in the editor. The second lavender chunk of text has such a blank line within it, and we can see it transformed to match our preference of using an asterism between significant portions of text. You can thus *mix* how you write and use the draft outline with Scrivener. If it feels more appropriate to have a sequence of short scenes in one single outline item then you can feel free to do so.

Our next example ([Figure 24.10](#)) adds a new option into the mix, and in doing so modifies how these items in the draft work together. If you're following along with the demonstration project, open the compile overview screen and click the [Assign Section Layouts...](#) button, switching the “Section” type to the “Section Text (After)” layout.

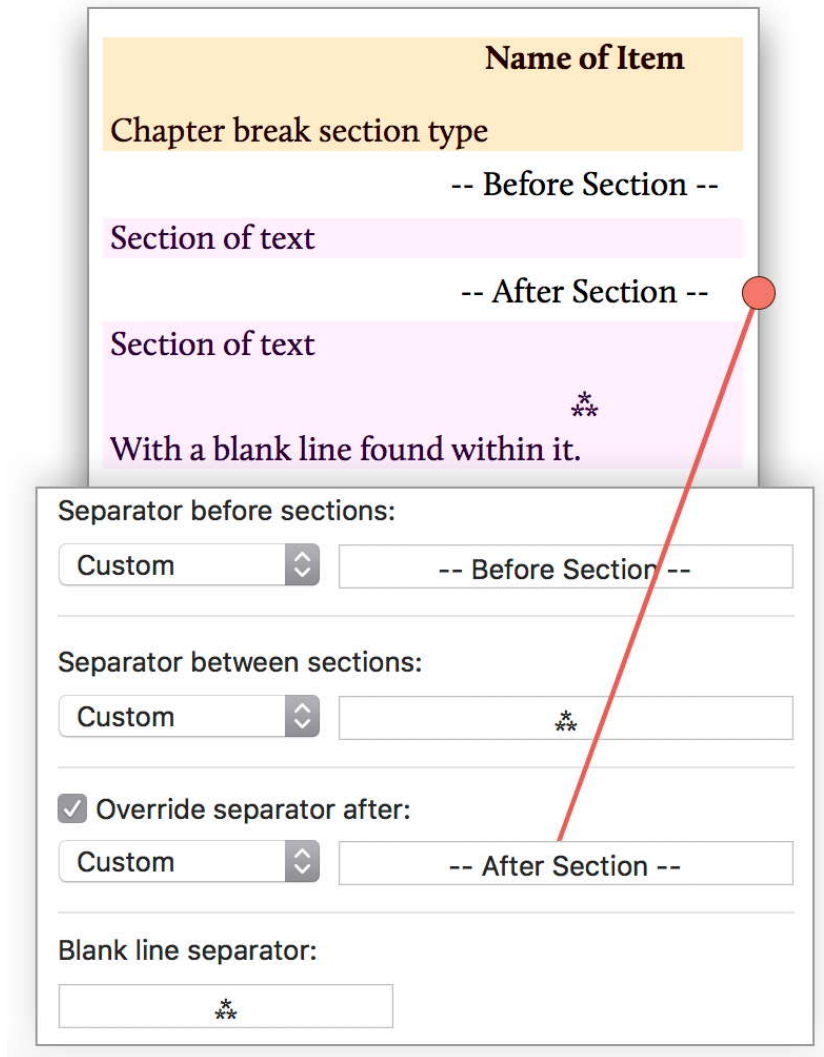


Figure 24.10: Overriding the separator following an item will modify any other separators that might have appeared in that slot, rather than adding an additional separator.

Override separator after This option not only inserts a separator after the section, it will do so in *all* cases, overriding either of the above options if necessary to do so. In our example here, the **Separator between sections** is overridden, but if the third chunk of text was orange, it would in that case have suppressed the page break that otherwise would have been inserted by the “Heading” layout.

Also of note, this setting does not override the **Blank line separator** option, which doesn’t formally insert separators *between* chunks of text, but rather modifies how ad hoc separators you type into the editor are formatted.

Need a less adaptive behaviour?

As you may have seen by now, separators are by their nature contextual. They are meant to be inserted logically into your draft, and will not blindly duplicate separation between items or insert separation where there is nothing to separate. If you do need separation around an element *no matter what*, then the Section Layouts prefix and suffix settings (section 24.2.3) settings will do just that. The provided demonstration project also includes a section layout called “Text Section (Prefix/Suffix)” that shows the interaction between a section prefix & suffix with the separators around them.

24.4.3 Separator Settings

Each of type of separator that can be made use of in defaults or per layout will have the same four options available to them:

- *Single return*: a single paragraph break will be inserted, causing the final appearance to run from one document to the next with no visible “seam”. In essence this is the “no separator” option.

This can have an adverse effect in Markdown-based and Fountain formats, where a clear empty line is expected between all elements, including paragraphs.

- *Empty line*: two paragraph breaks will be inserted, causing a visible space between the items.
- *Page break* or *Section break*: a page break will be inserted, causing the following item’s text to move to the next page of the manuscript. Those formats that do not have a concept of paper, such as eBooks and web files, will refer to them as “section breaks”, and use the following behaviours:
 - Plain-text (TXT): the Unicode “Form Feed” control character (U+000C) will be inserted at the beginning of the line of the item that generated the separator with no carriage returns around it. Some text editors may handle this code in their display of the text (TextEdit in Page Wrap mode for instance). If the intention is to use the output of this document in processing this separator should in general not be used unless the processing engine is capable of handing a Form Feed character in some intelligent manner (many will just throw errors).
 - Web page (HTML): an `<hr/>` element will be inserted instead of a page break.
 - Fountain: the markup code for a page break (===) will be inserted.
 - eBook formats: all eBook formats will use a page break separator to indicate a formal cut to a new section. This is a significant cut in that

it will be the basis for generating the internal table of contents (as well as the HTML contents you can see in the reader). These cuts are also inserted into the navigation index for forward/backward by chapter movement in those readers that offer the capability.

- Markdown-based formats: the markup code for a section break (----) will be inserted. A special exception exists for MMD → LaTeX (.tex). The syntax for a page break will be inserted (\pagebreak), in such a way that MultiMarkdown will pass the code directly through to the final .tex file.
- *Custom*: Anything entered into the adjacent text field will be placed between the two items on its own line, using the paragraph attributes of the line preceding it with centre-alignment added.
 - Image Placeholder Tags (subsection 15.7.5) can be used here, providing a way of inserting custom separator graphics between sections.
 - If you require more spacing, you can insert your own carriage returns with the **Opt-Return** key combination.

There are a few additional options that may appear at the very bottom of this pane, depending on the document type selected:

Ignore indents when centering custom separators Available to file types that use formatting. Since separator lines inherit their base formatting from the preceding paragraph, this can often mean the separator will have a first-line indent like an ordinary block of text would. This indent pushes the calculation of what is “centre” over by the width of that indent. Normally you will want a separator to be aligned to the middle of the *page*.

You may want indents to be factored in if the section of text being separated is itself bulk-indented. For example, an extended block with separators in between sections of that quote would arguably look more coherent if the visible separator between them were indented accordingly.

Custom separator font Available to file types that use formatting and are capable of drawing upon system fonts (unlike eBooks). Tick the checkbox to enable the font selection tool. This can be particularly useful if you have a wing-ding style font that you want to make use of. This setting will not *not* be previewed in the settings above, so you will need to know which characters to type in to make use of such symbols.

[Return to chapter ↗](#)

24.5 Styles

Where it comes to compiling stylesheets from your project, Scrivener takes a somewhat unique approach to the problem in that every compile format can