you will add you own menu items.

3. Change the `extensions` string array to list just the appropriate extension(s) for your new filetype.[2]

4.3   TextContents.java

This is conceptually part of `TextDocument`, so you could chose to merge the two if you consider it appropriate. You will need to modify the various I/O methods to use ones appropriate to your new filetype.

4.4   TextDocument.java

Use an appropriate Swing component instead of a `JTextArea` as the window component.

4.5   TextEditor.Java

1. Change the static strings that identify the editor.

2. Change references to TextXXX to something appropriately-named for your new filetype.

4.6   Other Files

You should feel free to split up your new Java files however seems appropriate, as long as the `XXXType`, `XXXDocument`, and `XXXEditor` classes remain.

You will not need the `Action` modules, except possibly a superclass serving the same purpose as `TextAction`). This would be a good idea if you want to centralize code to deal with things like translating Swing-specific information into specific data model elements.

5   Advice

Don't try to understand every detail of the existing software; there is too much of it for the time you have available. One of the values of software architecture is providing an appropriate level of abstraction that lets you comprehend the overall system without having to delve into details.

---

2   Some packages such as the Java image library can read several formats with different extensions.