

# Small App - FASTA File Editor

This chapter illustrates how to use the Small Application Classes framework to construct an editor for a DNA FASTA sequence file. The FASTA file format is one of several file formats used in bioinformatics to store DNA sequence information. Here is an example taken from the National Center for Biotechnology Information (NCBI at <http://www.ncbi.nlm.gov>)

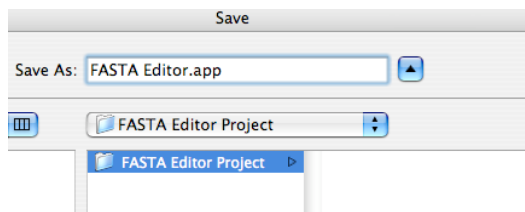
```
>gi|73672115|gb|DQ154325.1| Soil bacterium RFS-I82 16S ribosomal RNA gene, partial sequence
AGAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGCGTGCCTAATACATGCAAGTCGAGCGAACAGATAA
GGAGCTTGCTCCTTTGACGTTAGCGGCGGACGGGTGAGTAACACGTGGATAACCTACCTATAAGACTGGG
ATAACTTCGGGAAACCGGAGCTAATACCGGATAAGATTTTGAACCGCATGGTTCAATAGTAAAGACGGC
CTTGCTGTCACTTATAGATGGATCCGCGCGTATTAGCTAGTTGTAAGGTAACGGTTACCAAGGCAAC
GATACGTAGCCGACCTGAGAGGGTGATCGGCCACACTGGAAGTGAACACGGTCCAGACTCCTACGGGAG
GCAGCAGTAGGGAATCTTCCGCAATGGGCGAAAGCTGACGGAGCAACGCCCGTGAGTGATGAAGGCTCT
TCGGATCGTAAAACCTGTTATCAGGGAAGAACAAACGTAAAGTAACTGTGCACGCTTTCGACGGTACCT
GATCAGAAAGCCACGGCTAACTACGTGCCAGCAGCCGCGTAAT
```

The file format is straightforward. The initial character is always ">" followed by the identifier. The first space character ends the identifier and starts the "annotation" which in turn is ended by a line feed. On the next line the DNA sequence is started and it may contain additional line feeds for readability. It continues until the next ">" or until the file ends.

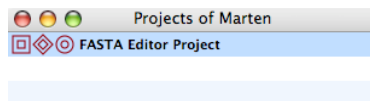
What we intend to do is create a document-centric application that can open a FASTA file in an editor window that contains three separate text edit boxes, one each for the ID, Annotation, and Sequence. It can only handle files that contain just one sequence.

## Creating the project

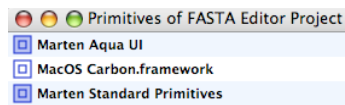
Launch the Marten IDE and create a new project. Name the project application "FASTA Editor" and save it in the project folder.



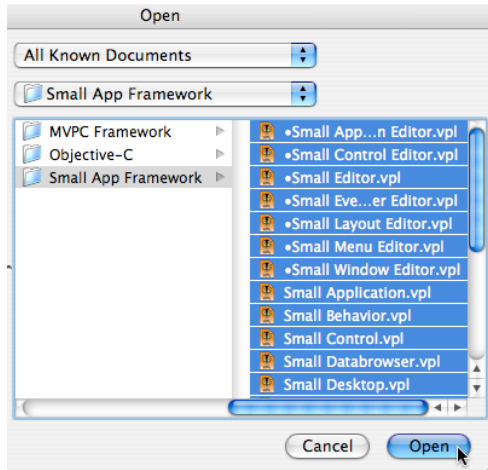
Name the project file "FASTA Editor Project" and save it in the project folder.



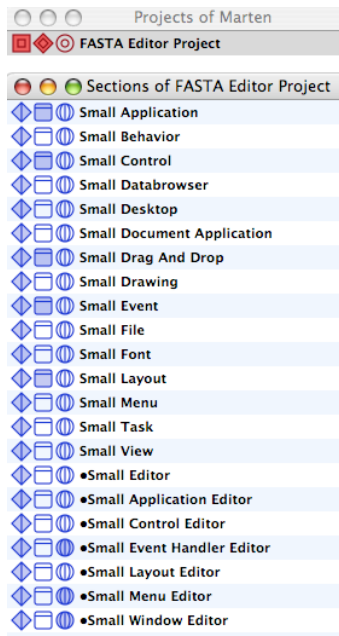
We need the standard libraries, so navigate to the appropriate Frameworks folder and add the MartenMacOSCarbon, MartenStandard, and MartenUIAqua libraries to the project.



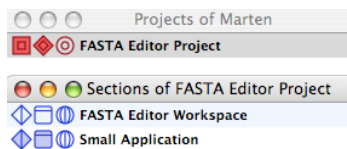
Since this is going to be an application based on the Small Application Classes framework, navigate to the Marten folder and open the entire contents of the Small App Framework folder.



Rearrange the opened sections into a more useful alphabetical order.



Next, create a workspace section that will contain your FASTA Editor specific code. Name it "FASTA Editor Workspace" and save it to the project folder.

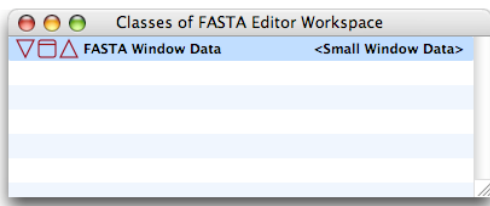


Create a FASTA file by copying the example text presented above and saving it as a plain text file named "sequence.fasta". The contents of the project folder should now be:

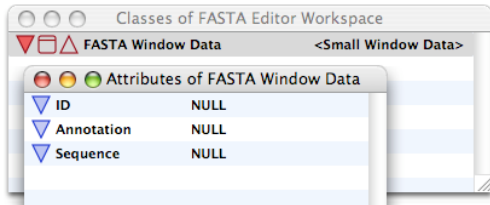


## Constructing the FASTA Window Data class

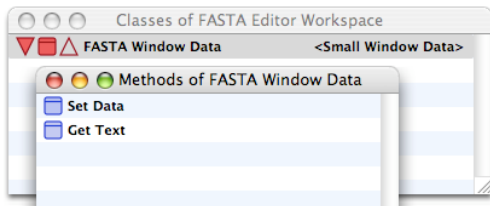
Before editing your application, you should create the single additional class needed to complete your code base. This is the class that represents the FASTA information and provides methods for transforming it. In a document-centric application this subclass of Small Window Data is the intermediary between the contents of the file and the contents of the editor window. The contents of the file must be transformed into an instance of this data class and then the instance must drive populating the contents of the editor window. Proceed by creating a new class called "FASTA Window Data" and install the Small Window Data class as the parent.



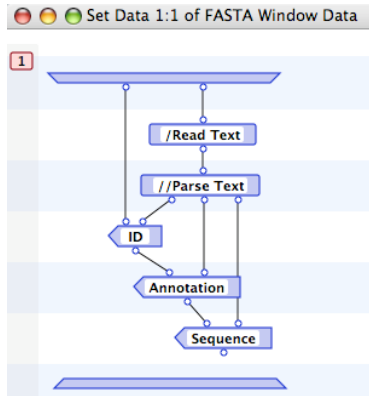
An instance of the class must be able to represent the information contained in a FASTA file, so create three attributes, one each for the ID, Annotation, and Sequence.



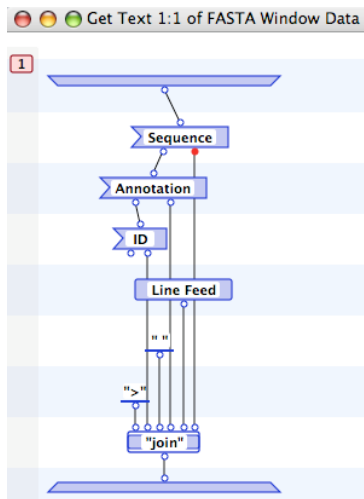
The instance must be able to populate its attributes with data from the file. This is the responsibility of the Set Data method. Copy the version of the Text Window Data class and paste it into the methods of FASTA Window Data. Similarly, when a file is created with the contents of the editor window, it requires the text data. This is provided by the Get Text method, so copy the Text Window Data version and paste it into the FASTA Window Data methods as well.



Modify the Set Data method to look like the following. Don't worry about creating the Parse Text method, you will code up that method while actually running the application.



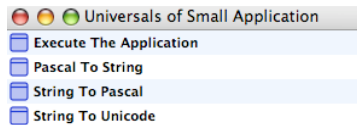
The text required for the file is easily constructed from the three attribute values. You just prefix the ID with a ">", follow it with a space. Then append the annotation followed by a line feed character. Finally the sequence is appended and the required text is constructed for writing to the file.



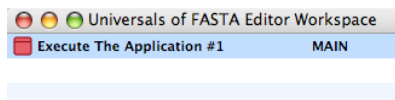
That is all the coding required before editing an instance of the application.

## Editing the application

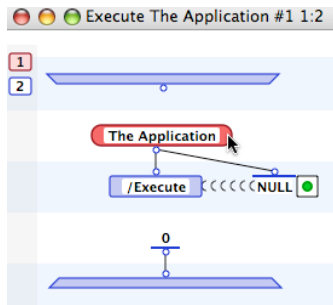
Remember, a Marten application must have a MAIN method which is the initial method that is executed by the application. An example of such a method is supplied as the first universal method of the Small Application section.



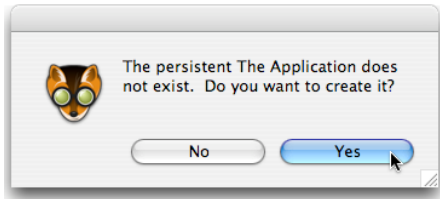
Copy the universal and paste it into the universals of the FASTA Editor Workspace. Install it as the MAIN method.



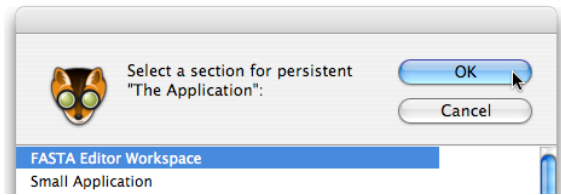
If you examine the initial case of this method, you see that it requires a persistent named "The Application". When the method is executed, it obtains the value of this persistent (which should be an instance of the Small Application class) and calls the appropriate Execute class method. The persistent doesn't exist so you need to create it. Double click the persistent operation in the case window.



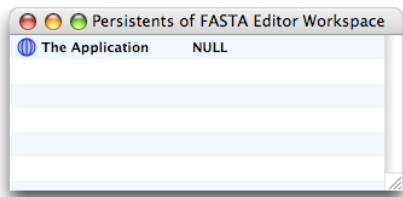
A dialog window opens and asks if you want to create this persistent. Click the Yes button.



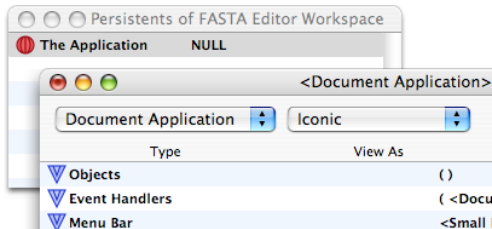
The persistent needs to be stored in some section file. The only file that should be modified is your FASTA Editor Workspace section file, so select that section and click OK.



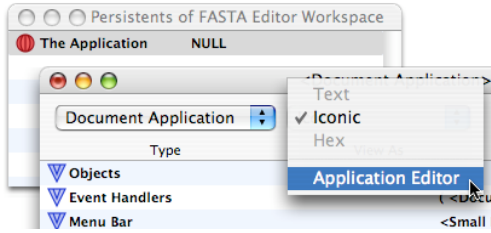
A window will open displaying the persistents of the FASTA Editor Workspace. The persistent "The Application" should appear on the list with the value NULL.



Double click the persistent to open up a NULL Value Window editor. Change the Type popup menu to "Document Application". The Document Application editor window should appear and look something like:

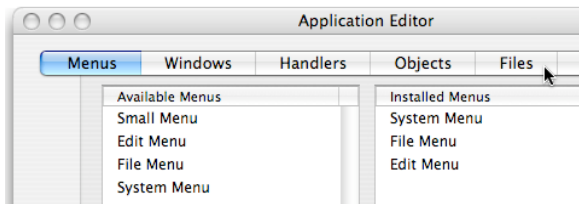


Now it's time to use an object editor to change the attributes of this Document Application instance. Under the "View As" popup menu, select "Application Editor".

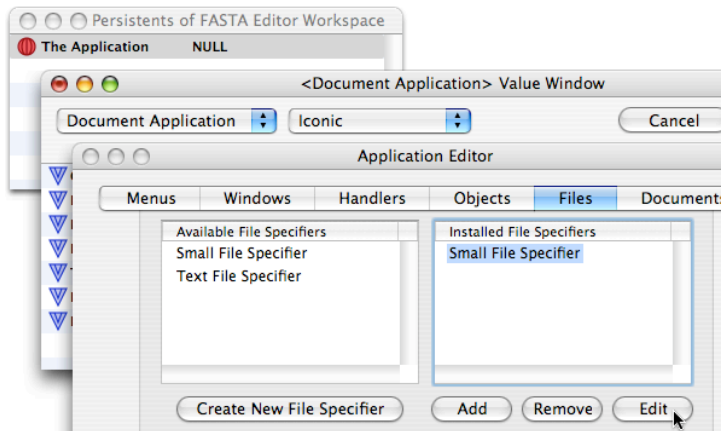


An Application Editor window opens. Notice that you can no longer click on the Document Application value window. That is because editors run in the project application process, not the Marten IDE process. When the project application process is frontmost, the Marten IDE is suspended.

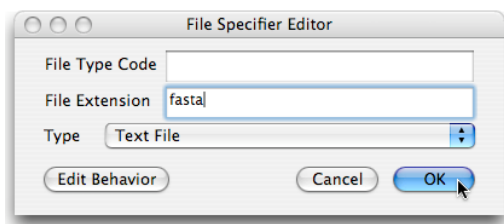
A Document Application has many attributes populated already. It has a menu bar containing a standard File menu and an Edit menu, it has a default event handler installed, and a default file specifier. The file specifier needs to be modified to accept just FASTA files, so click on the Files segment of the segmented view.



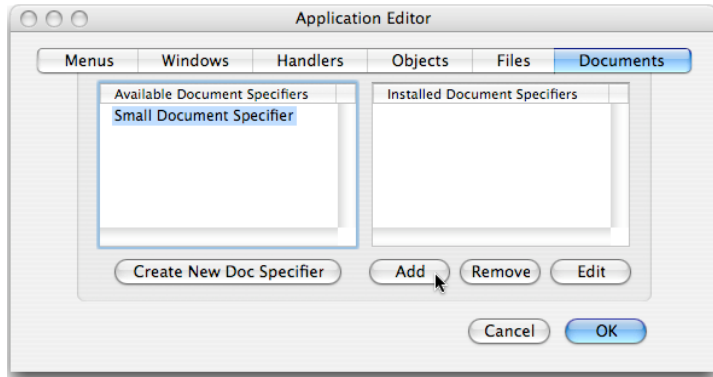
The view changes to display a list of the possible file specifiers and a list of the currently installed file specifiers. Now a file specifier is an object that specifies what types of files that application can open and what to do when such a file is opened. The default action when opening a file is described a bit later. Modify the specifier by selecting it in the Installed list and clicking the Edit button.



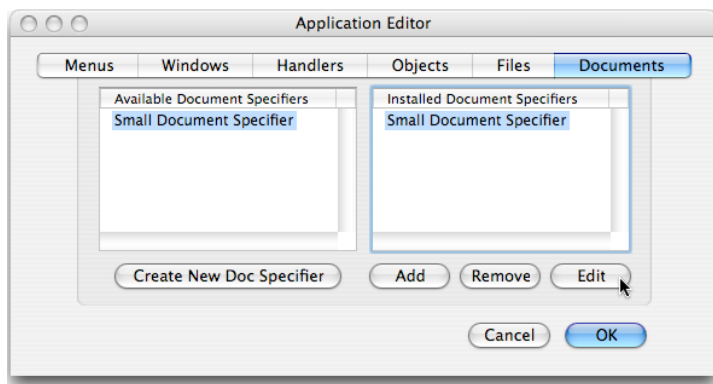
A File Specifier Editor opens with default values. Delete the File Type Code value and change the File Extension value to "fasta". This means that the Open dialog will only show files with a ".fasta" extension as acceptable to open. The Type popup menu states that such a file will be opened as a Text File. Click the OK button when finished with your modifications.



The default behavior for opening a file is to search a list of document specifiers for the proper mapping between a specific kind of file and a certain kind of window. You need to install the appropriate mapping so click on the Documents segment of the segmented view. Select a Small Document Specifier in the Available list and click the Add button to install it.

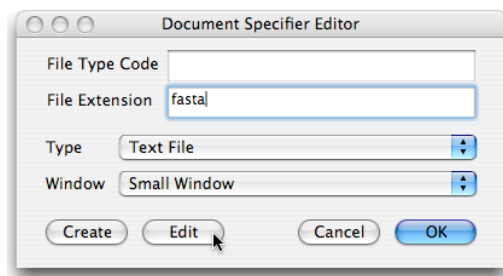


Now edit the newly installed document specifier by selecting it in the Installed list and clicking the Edit button.

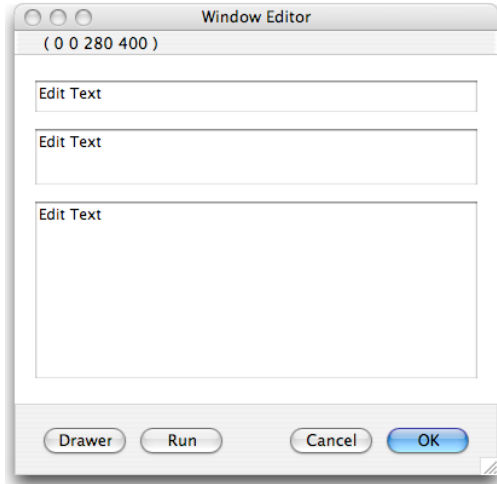


## Creating the FASTA file editor window

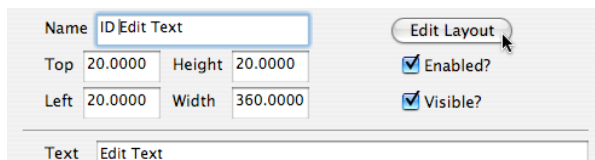
As stated earlier, a document specifier maps a certain type of file with a certain type of window. The Small Application Classes framework provides for a many to one mapping of file types to a window. For example, the application might be able to open three different kinds of text files ( plain-text, FASTA, etc. ) to a single type of window ( a window with a single text edit control ). However, in this case you will create a one-to-one mapping of a FASTA file to a FASTA Editor window. Consequently, delete any default File Type Code and modify the File Extension to "fasta". The Type of file should be a Text File (consequently when the New command is selected, the window will have an associated file) and the Type of window should be a generic Small Window. There is usually no need to create a subclass of Small Window but that option is available. The window certainly needs to be edited, so click the Edit button to open up a Small Window Editor.



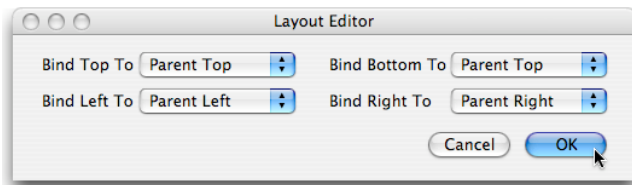
The editor opens with the current window bounds displayed in the upper left hand corner. Add three Edit Text controls and position them so that the window looks something like:



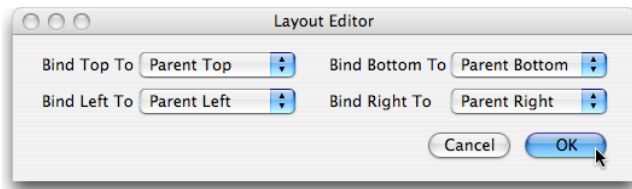
Double-click the top control and modify the name to be "ID Edit Text". Then edit the control layout by clicking the "Edit Layout" button.



You want the text edit box to keep the same distance from the top of the window but to stretch to the right if the window is resized. Modify the binding popup menus to look like:

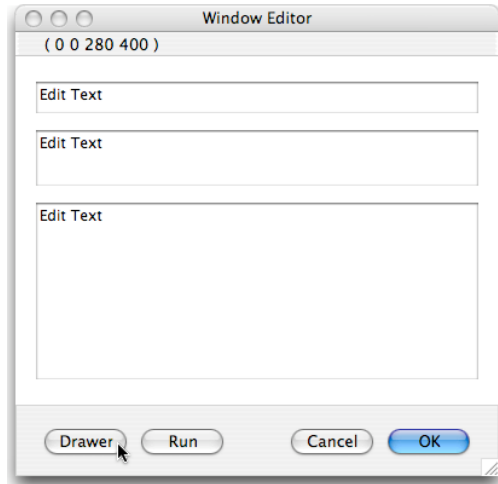


Click the OK button for the Layout Editor and then the OK button for the top control. Repeat this process for the next two Edit Text controls naming them "Annotation Edit Text" and "Sequence Edit Text" respectively. There is one difference though. The bottom control should stretch with the bottom of the window, so modify its layout as:



Close up shop by clicking the OK buttons and return to the Window Editor. The properties of the window itself now need to be set. Click the Drawer button to open a drawer with a variety of controls that determine the properties of the window.





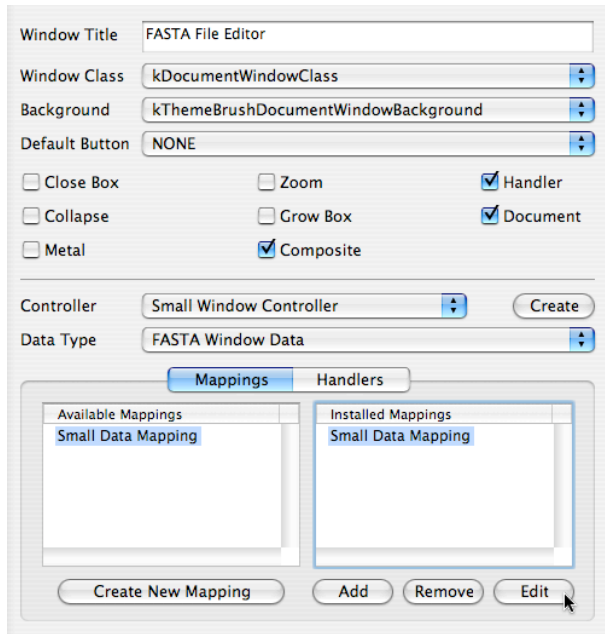
The drawer opens revealing further controls. The window may need to be resized to reveal all the controls but since you've already set the resize properties of the text edit boxes, everything will resize properly.

Change the title of the window to FASTA File Editor. This will not actually be the displayed window title (which will be the name of the file) but serves to illustrate how to modify the title. There is no push button to set as the default, so leave that popup menu alone. The background could be changed to be different if you want. Currently it is just a standard white document background. The window should always be a composite window with the standard window handler and the standard document window "buttons" (Close, Minimize, Zoom, and Grow).

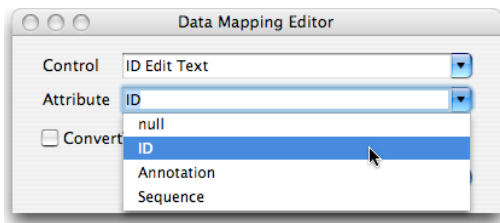
The key things to modify are what controller to use and what data to drive the population of the window contents. The Small Application Classes framework uses a Model-View-Controller design which encourages you to separate out the "actions" taken by the window in response to user interface interactions into an abstract class called a "controller". Basically this will be the class that contains methods that are invoked when a user clicks a button or slides a scroll bar for example. For example, if you were to implement an Edit Text control that contained a bit of sequence to "find" as a substring within the larger DNA sequence, you would probably place a push button on the window. In this case, you would want to create a custom controller class (say "FASTA Window Controller") to contain a method that would be invoked when the push button was clicked. However, because you don't have any buttons to click and only rely on the standard behavior of the Edit Text control, you can use an instance of the generic Small Window Controller as the window controller. Select that class from the Controller popup menu.

It is also time to make use of that data class you created at the very beginning. The FASTA Window Data class is meant to drive the population of the contents of the window. The controller that handles this by using data mappings from the class to the window controls. Select FASTA Window Data as the controller Data Type.

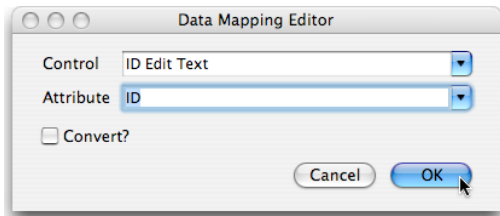
The mappings must be installed, one for each control. Install the first by selecting Small Data Mapping from the Available list and clicking the Add button. Then edit it by selecting it in the Installed list and clicking the Edit button.



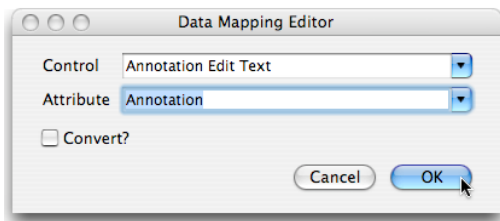
A Data Mapping Editor opens with combo boxes allowing you to map the window control to the data class attribute. You have set things up so that this is very straightforward. The ID Edit Text control should be mapped to the ID attribute of the FASTA Window Data class.

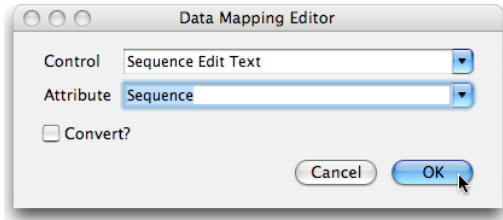


Be sure NOT to press the Return key until you have specified both the Control and Attribute (this is equivalent to clicking the OK button and the window will close prematurely). When you have editor looking like that below, click the OK button or hit the Return key.

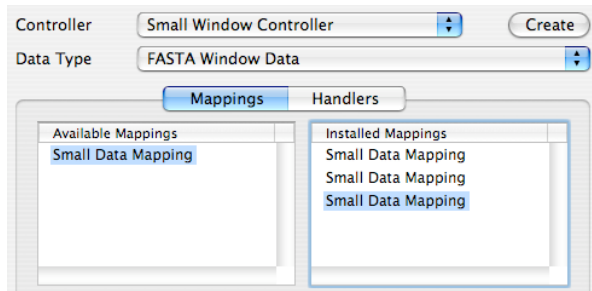


Continue in a similar vein with the other mappings. Install them and edit them as:

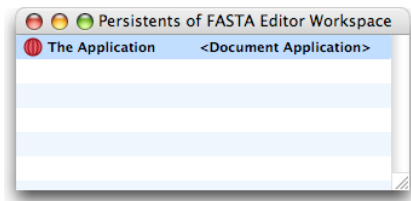




After editing the data mappings, the drawer should look like:

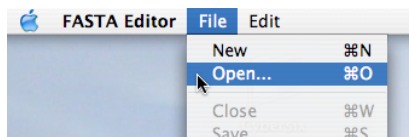


Click the editor OK buttons to close all the editors and return to the Marten IDE environment with your Document Application instance installed as the value of the "The Application" persistent.

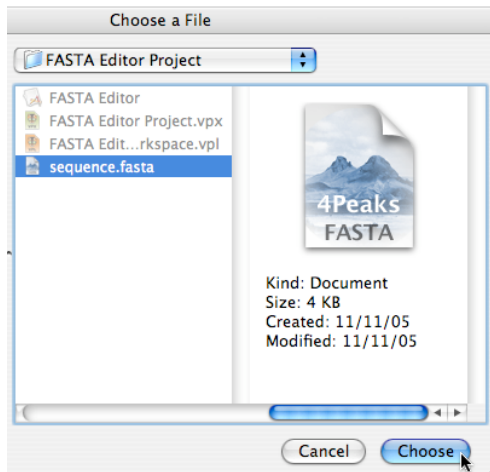


## Running the application and creating the parsing code

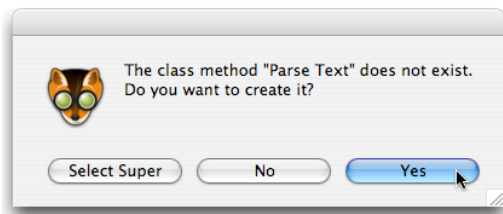
You are just about done, but one thing that remains is to create that "Parse Text" method. To do this, it is most convenient to simply run the application and force it to be created. Select the Run Application method from under the Run menu. Your FASTA Editor runs, displaying the standard document application menu bar. To force creation of the Parse Text method, select the Open command from under the File menu



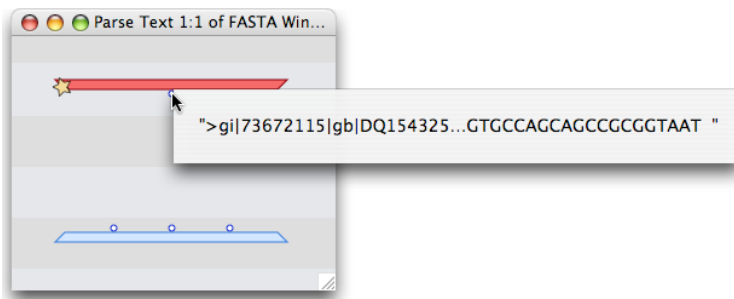
and then choose the FASTA file you created earlier from the example text.



The application will attempt to execute the FASTA Window Data/Parse Text method in the course of reading the file. Since there is no Parse Text class method, the project application halts and the Marten IDE then asks if you want to create it. Click the Yes button.

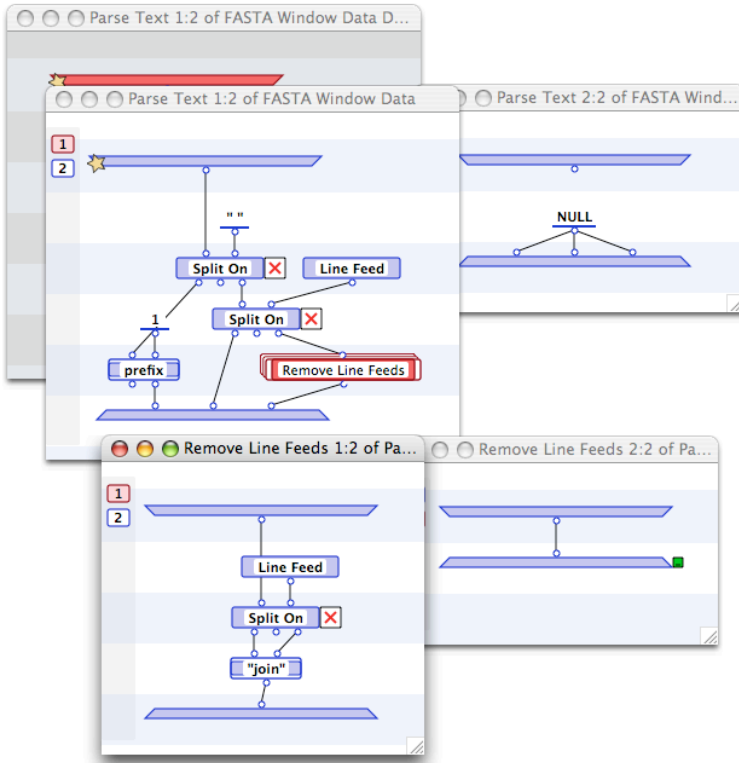


A debug window for the newly created method will open because of the Breakpoint set on the input bar. Hit the Return key to execute the input bar and click the mouse on the input bar root. A popup window opens displaying the value of the root. Now you can see the actual text that the method must parse into the ID, Annotation, and Sequence values.



It is time to finish writing the code for the application. Create an additional case and install operations as illustrated below. The "Split On" universal method is useful because it will take a string, find a substring within it, and then return the "pre-substring" string, the substring, and the "post-substring" string. It will fail if the substring cannot be found within the string.

After creating the code, be sure to step through the debug window and examine the various operation root values to see how it all works.



At the end of debugging the method, remove the input bar breakpoint and save the section. When the method finishes executing, your FASTA editor window opens displaying the identifier, annotation, and the sequence. You can modify any of the values and save the result either back to the file or to a new file.

```

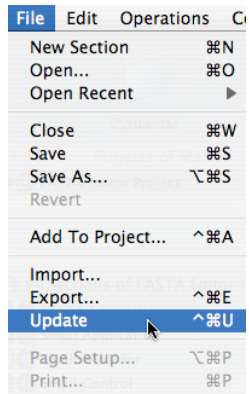
sequence.fasta
-----
gi|73672115|gb|DQ154325.1

Soil bacterium RFS-I82 16S ribosomal RNA gene, partial
sequence

AGAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGCGTGCCTA
ATACATGCAAGTCGAGCGAACAGATAAGGAGCTTGCTCCTTG
ACGTTAGCGGCGGACGGGTGAGTAACACGTGGATAACCTACCT
ATAAGACTGGGATAACTTCGGGAAACCGGAGCTAATACCGGAT
AAGATTTTGAACCGCATGGTTCAATAGTGAAGACGGCCTTGC
TGTCACCTTAGATGGATCCGCGCGCTATTAGCTAGTTGGTAAG
GTAACGGCTTACCAAGGCAACGATACGTAGCCGACCTGAGAGG
GTGATCGGCCACACTGGAAGTGAACGACGCGTCCAGACTCTAC
GGGAGGCAGCAGTAGGGAATCTTCCGCAATGGCGAAAGCCTG

```

The only thing that remains to do is to update your project application so that it can run as a standalone MacOS X executable. Select the Update command from under the File menu.



Once the Update process completes, save everything and quit the Marten IDE. Navigate to your project application in the project folder and double click it. It will launch and you can begin editing DNA FASTA files!