

Change request log

1 Team

Brent Staab

2 Change Request

Change Request: je-1

Modify the status bar to show

- The word offset of the caret from the beginning of the file
- The number of words in the file

The new caret display options should be added to the Status Bar preference

3 Concept Location

Step #	Description	Rationale
1	Run the system	
2	Verify current functionality described in je-1. <ul style="list-style-type: none"> - Add text to edit window - Verify status bar content 	Make sure program behaves as advertised
3	Explore status bar using mouse <ul style="list-style-type: none"> - Click/double-click: No functionality - Hover over status bar: This displays the format of the content <ul style="list-style-type: none"> o <code>line,column[-virtual] (offset/length)</code> 	See if there is any information provided by the GUI that will help identifying the location in code to start looking.
4	Search for 'line,column' in IDE's search tool <ul style="list-style-type: none"> - Found string in <code>org/jedit/localization/jedit_en.props</code> - This is the English version. There are other languages supported, but I do not plan on changing those 	Try to identify where the corresponding functionality exists in code
5	Modify the string found in step 3, compile and rerun	Verify that I found the correct location
6	Open 'Plugins->Plugins Options->Global Options->Status Bar" <ul style="list-style-type: none"> - Found the strings associated with enabling/disabling each of the options displayed in the status bar - The one I focused on was 'Show length of file' 	Try to identify something that I could search for in the code base to find the functionality of enable/disable the options
7	Search for 'Show length of file' in IDE's search tool <ul style="list-style-type: none"> - Found string in <code>org/gjt/sp/jedit/options/StatusBarOptionPane.java</code> 	Try to identify where the corresponding functionality exists in code for enabling/disabling the options
8	Run jEdit in Eclipse debug mode with breakpoint in the function found in previous step.	Verify the code in question is executed when options are enabled/disabled
9	Search for 'show-caret-bufferlength' in IDE's search tool <ul style="list-style-type: none"> - Found string in <code>org/gjt/sp/jedit/gui/StatusBar.java</code> 	Try to identify where the corresponding functionality exists in code where the options are consumed
10	In <code>org/gjt/sp/jedit/gui/StatusBar.java</code> , right click on 'getLength()' portion of 'int bufferLength = buffer.getLength();' and select 'Open Declaration' <ul style="list-style-type: none"> - Functionality defined in <code>org/gjt/sp/jedit/buffer/JEditBuffer.java</code> 	Try to identify where the corresponding functionality exists in code for populating the 'length' portion of the status message described in step 3.
11	Task finished	I believe I found the location of all major functionality needed to implement the change request

Time spent (in minutes): 45

4 Impact Analysis

Step #	Description	Rationale
1	<i>It was determined that all the changes listed above are required to fully implement the change request</i>	<i>Try to determine if anything from the Concept Location phase could be eliminated</i>
2	<i>For the 'hint string' I determined that updating it to include (word offset/total words) is very low risk.</i>	<i>The hint string is already being printed and adding a few more characters would</i> <ul style="list-style-type: none"> - <i>Memory needed to store the extra characters</i> - <i>Increase the displayed string length</i> - <i>Virtually no extra processing cycles</i>
3	<i>For the Options menu to enable/disable the new word options, the risk is very low</i>	<i>The logic is expected to be executed very rarely so the exposure is very small. Also, the logic is very simple, essentially just storing/retrieving flags.</i>
4	<i>For the word count functionality, I determined the risk to be low to medium.</i>	<i>If the corresponding options are enabled, the logic will be executed every time the buffer is updated. This includes adding/deleting a character or cut/paste operations. The larger the buffer, the more time it will take to calculate the values.</i>

Time spent (in minutes): 30

5 Prefactoring (optional)

Step #	Description	Rationale
1	<i>For the hit string, there is no need for extra variables or logic.</i>	<i>The existing string could be updated to include the new information</i>
2	<i>For the Options menu, add new options using the same logic</i>	<i>The logic in question appears to work as expected, extending to include the new parameters is very straight forward</i>
3	<i>For the word count logic, I found that I could use existing methods that have been well tested.</i>	<i>We can reduce unexpected/undesirable behavior by using well tested existing functionality</i>

Time spent (in minutes): 10

6 Actualization

Step #	Description	Rationale
1	<i>For the 'hint' string, update the existing string to include '(word offset/total words)'</i>	<i>This matches the format of the data being added to the status bar.</i>
2	<i>For the options menu, search for 'caret-offset'</i>	<i>Find all reference of this string, this tells me what needs to be added for the new options</i>
3	<i>Add 'caret-word-offset' and 'caret-bufferWordLength' versions of the 'caret-offset' references</i>	<i>We're duplicating the functionality associated with 'caret-offset', so it stands to reason there should be equivalent code for the new options</i>
	<i>In org/gjt/sp/jedit/buffer/JEditBuffer.java, search for the code to calculate the current status bar information</i>	<i>Since we're adding functionality to existing logic, it makes sense to keep all logic in the same area.</i>
4	<i>Add functions to calculate the word count of a buffer</i> <ul style="list-style-type: none"> <i>- Used Google to find an existing algorithm for counting words in a string</i> <i>- Using existing functionality to get the text buffer content.</i> <i>- Depending on the word count needed (from start or total), use the getText() method to get the desired information</i> 	<i>This functionality is primary function described in the change request. By using existing functionality, reduced the potential for negative side effects.</i>
5	<i>In org/gjt/sp/jedit/gui/StatusBar.java, I added logic to conditionally call the functions added in step 4 to determine the word counts. The logic was placed in functionality.</i>	<i>Since the amount of time it takes to execute the code increases with the size of the buffer, I wanted to avoid this calculation if the option is disabled.</i>

Time spent (in minutes): 120

7 Postfactoring (optional)

Step #	Description	Rationale
1	<i>N/A</i>	<i>N/A</i>

Time spent (in minutes): 0

8 Validation

Step #	Description	Rationale
1	<p><i>Test Case: Hint string</i> <i>Input: Hover over the lower left side of the status bar</i> <i>Expected Output:</i> <i>line,column[-virtual] (offset/length)(word offset/total words)</i></p>	<p><i>Make sure new content is described on hint string</i></p> <p><i>The test passed</i></p>
2	<p><i>Test Case: Verify 'word offset' and 'total words'</i> <i>Input:</i></p> <ol style="list-style-type: none"> 1) Add several words in the edit window 2) Delete characters/words from edit window 3) Paste text into the edit window 4) Cut text from edit window <p><i>Expected Output:</i> For all inputs, the word count matches content in edit window</p>	<p><i>Make sure the word count functionality works as expected in a variety of scenarios</i></p> <p><i>The test passed</i></p>
3	<p><i>Test Case: Verify enable/disable functionality</i> <i>Input:</i></p> <ol style="list-style-type: none"> 1) Add several words in the edit window 2) Disable 'caret offset in words from start' 3) Disable 'length of file in words' 4) Enable 'length of file in words' 5) Disable 'length of file in words' 6) Enable 'caret offset in words from start' 7) Enable 'length of file in words' 8) Disable 'Show length of file' 9) Disable 'caret offset from start of file' 10) Disable 'Show caret line number' 11) Disable 'caret offset from start of line' <p><i>Expected output:</i></p> <ol style="list-style-type: none"> 1) Full status bar is displayed and all content is correct line,column (offset/length)(word offset/total words) 2) Offset in words is removed from display line,column (offset/length)(total words) 3) Total words is removed from display line,column (offset/length) 4) Total words is displayed line,column (offset/length)(total words) 5) Offset in words is removed from display line,column (offset/length) 6) Offset in words from start is displayed line,column (offset/length)(word offset) 7) Total words is displayed line,column (offset/length)(word offset/total words) 8) Length of file is removed from display line,column (offset)(word offset/total words) 9) Offset from start of file is removed line,column (word offset/total words) 10) Caret line number is removed (word offset/total words) 	<p><i>Make sure the word count functionality works properly when enabled and disabled. Verify multiple combinations to ensure the correct content is displayed and in the correct format</i></p> <p><i>The test passed</i></p>

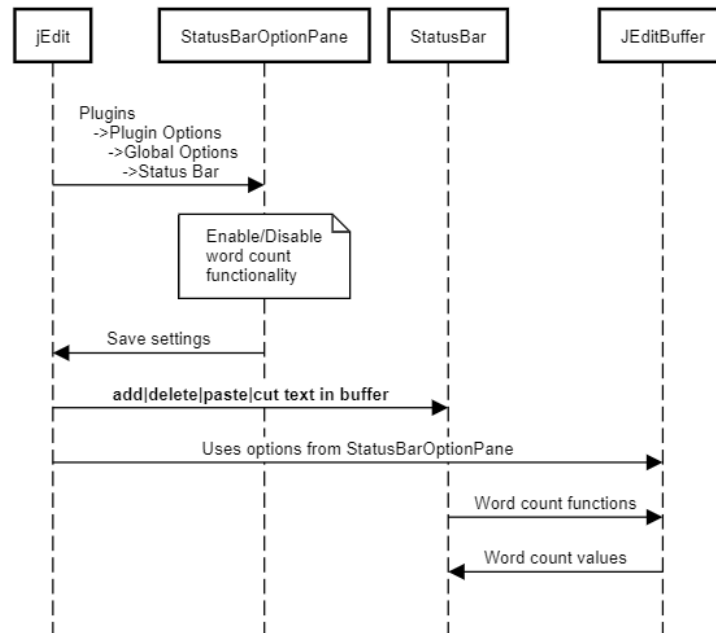
Time spent (in minutes): 90

9 Timing

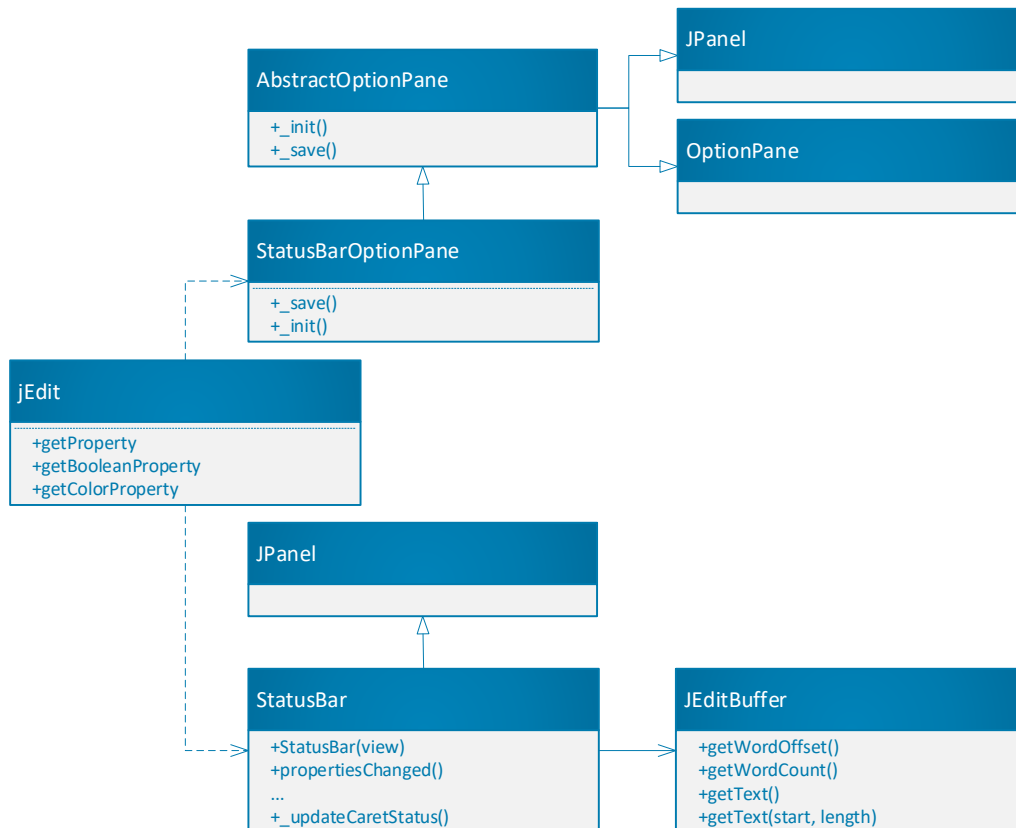
Phase Name	Time (in minutes)
Concept location	45
Impact Analysis	30
Prefactoring	10
Actualization	120
Postfactoring	0
Verification	90
Total	295

10 Reverse engineering

UML sequence diagram



Partial UML class diagram



11 Conclusions

For this change, concept location was relatively easy because the information found during experimentation on the live system was easily located using the Find functionality in Eclipse. Once the primary location was identified, additional locations were found using the 'Open Declaration' functionality. Impact analysis leveraged my experience and I made assumptions based on my experience in industry. As for actualization, I used the editor built into Eclipse which provided hints including auto completion, syntax and spelling. Testing was accomplished by manually testing the modified system. I tried to exercise as many scenarios as I could think of, though I'm sure there are some that I missed.

Classes and methods changed:

- *Classes*
 - *StatusBarOptionPane*
 - *org/gjt/sp/jedit/options/StatusBarOptionPane.java*
 - *StatusBar*
 - *org/gjt/sp/jedit/gui/StatusBar.java*
 - *JEditBuffer*
 - *org/gjt/sp/jedit/buffer/JEditBuffer.java*
- *Methods*
 - *StatusBarOptionsPane*
 - *_init()*
 - *_save()*
 - *StatusBar*
 - *updateCaretStatus()*
 - *JEditBuffer*
 - *ADD - getWordOffset(int caretPosition)*
 - *ADD – getWordCount()*
- *Other*
 - *org/jedit/localization/jedit_en.props*
 - *Adaptations specific to the English language*