

Change request log

1. Concept Location

Step #	Description	Rationale
1	<i>We compiled the code and ran the system.</i>	
2	<i>We interacted with system: creating files, exploring different functions.</i>	<i>To get familiar with some of the features of the system, and identify the screens or graphical elements we had to change.</i>
3	<i>Utilized the IDE's search tool with regular expressions to locate instances of "horizontal scroll" within the codebase.</i>	<i>As the scroll bar will be horizontal and vertical.</i>
4	<i>From 5 matches in 4 files we clicked on TextArea.java .</i>	<i>As we wanted to know which class contains the functionality or implementation of scroll bars and so we can keep our toggle function inside it.</i>
5	<i>Utilized the IDE's search tool with regular expressions to locate instances of "toggle-statusbar" or similar toggle features within the codebase.</i>	<i>The search aimed to find existing toggle implementations, such as "toggle-statusbar" and "toggle-gutter", to understand how toggle functionality is typically implemented within the system. This would ensure consistency in user experience and code design. Searching for similar functionalities allows for reusing patterns or code, promoting maintainability and reducing the introduction of errors.</i>
6	<i>From 10 results across 10 files, we clicked on actions.xml.</i>	<i>As actions.xml was common link in between toggle-statusbar and other similar toggle and also actions.xml serves as a centralized location where all user-triggered actions are defined.</i>
7	<i>From 10 results across 10 files, we clicked on jedit_gui.props.</i>	<i>As it was also common among other similar functionalities and we found that jedit_gui.props holds configuration settings for the jEdit text editor's graphical user</i>
8	<i>From 10 results across 10 files, we clicked on jedit_en.props.</i>	<i>As it was also common among other similar functionalities and as name suggests it keeps language specific info and how to display the option.</i>
9	<i>Utilized the IDE's search tool with regular expressions to locate instances of "gutter" within the codebase.</i>	<i>As it is similar functionality feature</i>
10	<i>From 107 results across 20 files we clicked on jedit.props</i>	<i>As it had global variable(property name) upon which we can toggle scroll bar</i>
10	<i>We inspected TextArea.java for more confirmation</i>	<i>we found the updateScrollBar() which updates states of scroll bars and also we got to know about horizontal and vertical objects for each bar</i>
11	<i>We decided to create toggleScrollBar method in the TextArea.java class</i>	<i>Due to Encapsulation, Reduction of dependencies, Ease of maintenance, Direct access to components we decided to put this method here</i>

Time spent (in minutes): 90

Classes and methods inspected:

- *org/jedit/localization/jedit_en.props*
- *org/gjt/sp/jedit/jedit.props*

- `org/gjt/sp/jedit/jedit_gui.props`
- `org/gjt/sp/jedit/textarea/TextArea.java`
 - `updateScrollBar()`
- `org/gjt/sp/jedit/actions.xml`
 - `<ACTION NAME="toggle-gutter">`
 - `<ACTION NAME="toggle-statusbar">`

2. Impact Analysis

Step #	Description	Rationale
1	<i>We made a list of classes can be impacted by the change in TextArea Class</i>	<i>To track the classes that could be impacted by the change specifically where horizontal and vertical object is using different methods.</i>
2	<i>ScrollListener marked as "Inspected and unchanged"</i>	<i>We debugged by putting debugger points at <code>scrollVertically</code> and <code>scrollHorizontally</code>. We reached to this methods while searching for the term "vertical" and "horizontal" using IDE's search tool. While debugging we found that, the scrollbar's are not directly making calls to this methods.</i>
3	<i>The actions.xml marked as "Inspected and unchanged"</i>	<i>It is not impacted as it is just linking the action required for the toggle button</i>
4	<i>The <code>jedit.props</code>, <code>jedit_en.props</code> and <code>jedit_gui.props</code> marked as "Inspected and unchanged"</i>	<i>As they are property dedicated for this change and only are the global variables used by the method <code>toggleScrollbar()</code> and <code>toggle-scrollbar</code> action.</i>

Time spent (in minutes): 40

Classes and methods inspected:

- `org/gjt/sp/jedit/jedit_gui.props`
- `org/jedit/localization/jedit_en.props`
- `org/gjt/sp/jedit/jedit.props`
- `org/gjt/sp/jedit/actions.xml`
 - `<ACTION NAME="toggle-scrollbar">`
- `org/gjt/sp/jedit/textarea/TextArea.java`
 - `void fireScrollEvent()`
 - `toggleScrollBar()`
- `org/gjt/sp/jedit/textarea/ScrollListener.java`
 - `scrolledVertically()`
 - `scrolledHorizontally()`

3. Actualization

Step #	Description	Rationale
1	<i>We first created <code>view.gutter.scrollBar.enabled</code> property</i>	<i>Because this is the property which helps in toggling the scrollbar. Because it was crucial in all the functionalities, so we defaulted it to true</i>
2	<i>We then implemented <code>togglescrollBar()</code> method in TextArea class</i>	<i>As it required to check if toggle is enabled or not then checking for vertical and horizontal scroll bar objects if they are visible or not</i>

3	Then we implemented action named toggle-scrollbar in Actions.xml	For this, we will toggle scrollbar using togglescrollbar method and then set newly updated toggle value to view.gutter.scrollBar.enabled property as toggling depends on it. Then we will repaint and revalidate to reflect state changes
4	We implemented jedit_gui.props	To integrate changes to UI
5	We implemented jedit_en.props	To make it aligned with existing menu items.
6	We performed functional testing and manual testing.	As it was necessary to test the code such that everything works.

Time spent (in minutes): 90

Inspected:

- org/gjt/sp/jedit/textarea/TextArea.java
 - void fireScrollEvent()
- org/gjt/sp/jedit/textarea/ScrollListener.java
 - scrolledVertically()
 - scrolledHorizontally()

Inspected and Changed:

- org/gjt/sp/jedit/jedit_gui.props
- org/jedit/localization/jedit_en.props
- org/gjt/sp/jedit/jedit.props
- org/gjt/sp/jedit/actions.xml
 - <ACTION NAME="toggle-scrollbar">
- org/gjt/sp/jedit/textarea/TextArea.java
 - toggleScrollBar()

4. Validation

Step #	Description	Rationale
1	Test case defined: The scrollbars should toggle by clicking toggle scroll bar in view menu. Expected output: The correct output was there	This is the regular expected behavior. The test passed.
2	Test case defined: Entering characters so that both horizontal and vertical scroll bars become visible and then toggling the scroll bars with Toggle Scroll Bars button inside view menu. Expected output: correct output of appearing and disappearing should be there of both scroll bars	This is the regular expected behavior. The test passed.
3	Test case defined: Entering characters so that only horizontal scroll bar is active. Expected output: correct output of appearing and disappearing for only horizontal scroll bar	This is the regular expected behavior. The test passed.
4	Test case defined: Entering characters so that only vertical scroll bar is active. Expected output: correct output of appearing and disappearing for only vertical scroll bar	This is the regular expected behavior. The test passed.
5	Test case defined: Checking default behavior of toggle scroll bar option in view menu.	This is the regular expected behavior. The test passed

	<i>Expected output: It should be checked (as in visible)</i>	
6	<i>Test case defined: Toggle Button should work well with multiple clicks. Expected output: Correct output was there</i>	<i>This is the regular expected behavior. The test passed</i>

Time spent (in minutes): 20

5. Summary of the change request

Phase	Time (minutes)	No. of classes inspected	No. of classes changed	No. of methods inspected	No. of methods changes
Concept location	90	5	0	3	0
Impact Analysis	40	6	0	4	0
Actualization	90	7	5	5	2
Verification	20	5	0	8	0
Total	240	23	5	20	2

6. Conclusions

For this change, concept location was pretty difficult as well as we do need to see different features with same functionality and determine the changes. Impact analysis was even harder as we do need to see all the impact of the change but as we had made separate method of toggle scrollbar, the impact was lesser. The actualization was simpler one. The verification was also easier but somewhat time consuming. Overall, this change was at difficulty in between moderate to hard.