

## Change request log

### 1. Concept Location

| Step # | Description  | Rationale  |
|--------|--|--|
| 1      | We installed and set up jedit on our systems .   |  |
| 2      | Once jedit is up and running we opened a document in jedit   | To get to know how jedit works and to explore various options present on jedit   |
| 3      | After opening a document in jedit, we focussed more on the view menu in jedit  | This is because we need to add a option to toggle scrollbars in the view menu.   |
| 4      | We explored Toggle Status Bar option, Toggle Line Numbers more in the view menu.   | This is because we need to add a similar functionality for scroll bars.  |
| 5      | We began searching “Toggle ” in the entire project using cmd+shift+F as there are multiple options in the view menu that has Toggle key word in the label. | We found 67 matches in 22 files in the entire project that had toggle and we ignored all other files except jedit_en.props file because they are not related to view menu.   |
| 6      | We explored jedit_en.props file further for “Toggle “  | We found 15 matches that had Toggle , among those we explored toggle-statusbar.label, toggle-gutter.label as they are in the view menu. And realized that we need to add a label for Toggle Scroll Bar here in the file. |
| 7      | We further searched toggle-statusbar in the entire project using cmd+shift+F   | We found two more appropriate usages apart from jedit_en.props . Which are jedit_gui.props and actions.xml and ignored the rest as they are for different languages.   |
| 8      | We explored jedit_gui.props and actions.xml further for toggle-statusbar.  | We found out that the code for functionality for toggle-statusbar is implemented in jedit-gui.props and action.xml   |
| 9      | We explored actions.xml file further.  | We searched for scroll inside the actions.xml file and we found few methods that are related like textArea.scrollDownLine() are being called in TextArea.java file.  |
| 10     | We explored TextArea.java file further.  | We found out that TextAtea.java is initializing the Vertical and Horizontal Scroll Bars.   |
| 11     | So we marked jedit_gui.props, actions.xml, jedit_en.props , TextArea.java as “located”.  | We confirmed that appropriate changes for toggling scroll bars needs to be added in these files.   |

**Time spent (in minutes): 150**

Classes and methods inspected:

- org/gjt/sp/jedit/textarea/TextArea.java
  - TextArea (propertyManger : IPropertyManager, inputHandlerProvider:InputHandlerProvider)
  - fireScrollEvent(vertical : boolean)
  - updateScrollBar()
- org/gjt/sp/jedit/textarea/ScrollListener.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/View.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)

Files inspected :

- org/jedit/localization/jedit\_en.props
- org/gjt/sp/jedit/actions.xml
- org/gjt/sp/jedit/textarea/TextArea.java
  - TextArea (propertyManger : IPropertyManager, inputHandlerProvider:InputHandlerProvider)
  - fireScrollEvent(vertical : boolean)
  - updateScrollBar()
- org/gjt/sp/jedit/textarea/ScrollListener.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/View.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)

## 2. Impact Analysis

| Step # | Description   | Rationale   |
|--------|---|---|
| 1      | We listed all the usages of jedit_gui.props   | This is to track all the usages of jedit_gui.props that gets impacted due to the change   |
| 2      | We listed all the usages of actions.xml and jedit_en.props in the entire project using command+shift+F.   | This is to track all the usages of actions.xml and jedit_en.props in the entire project that could be effected by the change . We discarded all those files because the change in these files wouldn't effect them .  |
| 3      | We listed all the usages of TextArea.java file in the entire project.                                     | We found that there are few usages of the file in the entire project . But we discarded all the files as we realized that the change is more likely in the TextArea() constructor so we checked for the usages of TextArea() constructor. We found 3 usages in JEditEmbeddedTextArea.java, JeditTextArea.java, StandaloneTextArea.java. |
| 4      | We discarded all files expect the four files jedit_gui.props, actions.xml, jedit_en.props , TextArea.java | It made sense to make changes only in these files as other files are not impacted by adding the Toggle Scroll Bar option.   |

**Time spent (in minutes): 90**

Classes and methods inspected:

- org/gjt/sp/jedit/textarea/TextArea.java
  - TextArea (propertyManger : IPropertyManager, inputHandlerProvider:InputHandlerProvider)
  - fireScrollEvent(vertical : boolean)
  - updateScrollBar()
- org/gjt/sp/jedit/textarea/ScrollListener.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/View.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/textarea/JEditEmbeddedTextArea.java
  - JEditEmbeddedTextArea()
- org/gjt/sp/jedit/textarea/JEditTextArea.java
  - JEditTextArea(view : View)
- org/gjt/sp/jedit/textarea/StandaloneTextArea.java
  - StandaloneTextArea( propertyManager : IPropertyManager)
- org/gjt/sp/jedit/jEdit.java
  - getBooleanProperty(name : String)
  - setBooleanProperty(name : String, value : boolean)
  - propertiesChanged()
- org/gjt/sp/jedit/gui/HistoryTextArea.java

Files inspected :

- org/jedit/localization/jedit\_en.props
- org/jedit/localization/package.html
- /jEditTeam17/JEdit-5.5.0-master/build.xml
- doc/users-guide/plugin-implement.xml
- doc/users-guide/bsh-commands.xml
- org/gjt/sp/jedit/actions.xml
- org/gjt/sp/jedit/jEdit.java
- org/gjt/sp/jedit/gui/HistoryTextArea.java
- org/gjt/sp/jedit/textarea/TextArea.java
  - TextArea (propertyManger : IPropertyManager, inputHandlerProvider:InputHandlerProvider)
  - fireScrollEvent(vertical : boolean)
  - updateScrollBar()
- org/gjt/sp/jedit/textarea/ScrollListener.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/View.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)

### 3. Actualization

| Step # | Description  | Rationale   |
|--------|--|---|
| 1      | From the concept location and impact analysis, we know that we need to make change in the jedit_gui.props, actions.xml, jedit_en.props , TextArea.java files . | The changes made in these files will sufficient enough to implement the required Toggle Scroll Bar Feature.   |
| 2      | We started by making the change in jedit_gui.props file.   | We added a field called toggle-scrollbar in the file in the view menu option. This is used to display a option for Toggle Scroll Bar in the view menu option.   |
| 3      | We made the necessary change in jedit_en.props file .  | We added a label for toggle-scrollbar element so that it will be displayed on the gui instead of the name.  |
| 4      | We made the code change in actions.xml and TextArea.java files.  | We added a action block in actions.xml that performs action when user clicks on Toggle Scroll Bar. We implemented two methods hideScrollBars() and scrollBarsAreHidden() in TextArea.java that will be used in action block of toggle-scrollbar element. We realized that hideScrollBars() and scrollBarsAreHidden() in TextArea.java would be enough to control the scroll bars visibility |
| 5      | hideScrollBars() method toggles the visibility of both horizontal and vertical scroll bars. scrollBarsAreHidden() methods tells if the scrollbars are hidden.  | We made use of setVisible() and isVisible() methods for vertical(represents vertical scroll bar ) and horizontal(represents horizontal scroll bar) to hide them .   |
| 6      | We tested out the code changes in the four files.  | We could successfully implement the Toggle Scroll Bar option under the view menu and it worked fine.  |

**Time spent (in minutes):** 60

Classes and methods inspected:

- org/gjt/sp/jedit/textarea/TextArea.java
  - TextArea (propertyManger : IPropertyManager, inputHandlerProvider:InputHandlerProvider)
  - fireScrollEvent(vertical : boolean)
  - updateScrollBar()
- org/gjt/sp/jedit/textarea/ScrollListener.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/View.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/jEdit.java
  - getBooleanProperty(name : String)
  - setBooleanProperty(name : String, value : boolean)

- propertiesChanged()

Files inspected :

- org/jedit/localization/jedit\_en.props
- org/gjt/sp/jedit/actions.xml
- org/gjt/sp/jedit/textarea/TextArea.java
  - TextArea (propertyManger : IPropertyManager; inputHandlerProvider:InputHandlerProvider)
  - fireScrollEvent(vertical : boolean)
  - updateScrollBar()
- org/gjt/sp/jedit/textarea/ScrollListener.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/View.java
  - scrolledVertically(textArea : TextArea)
  - scrolledHorizontally(textArea : TextArea)
- org/gjt/sp/jedit/jEdit.java
  - getBooleanProperty(name : String)
  - setBooleanProperty(name : String, value : boolean)
  - propertiesChanged()

Class and Methods changed :

- org/gjt/sp/jedit/textarea/TextArea.java
  - hideScrollBars()
  - scrollBarsAreHidden()

Files changed :

- org/gjt/sp/jedit/actions.xml
- org/gjt/sp/jedit/jedit\_gui.props
- org/jedit/localization/jedit\_en.props

## 4. Validation

| Step # | Description   | Rationale   |
|--------|---|---|
| 1      | <p>Test case defined: Should have an option under view menu for Toggling Scroll Bar.</p> <p>Expected output : Toggle Scroll Bar option under view menu.</p>                                       | <p>We could see the Toggle Scroll Bar option under the view menu and is as expected behaviour..</p> <p>The test case passed.</p>  |
| 2      | <p>Test case defined: Open a document that has content more than the window size and when clicked on the Toggle Scroll Bar option</p> <p>Expected output : Both Scroll Bars should be hidden.</p> | <p>We could see that the Scroll Bars are hidden when the user clicks on Toggle Scroll Bar option. And the Test case passed.</p>   |
| 3      | <p>Test case defined: When the user clicks back on the Toggle Scroll Bar option</p> <p>Expected output : Scroll Bars should be visible again.</p>   | <p>When we clicked on the Toggle Scroll Bar again we could see that the Scroll Bars are visible to the user again and is expected behaviour and the test case passed.</p> |

|   |   |  |
|---|---|--|
| 4 | Test case defined: Test for other features like selecting different options like Toggle Line Numbers etc.<br>Expected output : These all should work just fine even after the change. | When we checked for the remaining features, we realized that all are working as expected and the new feature didn't impact any of them . And the test case passed. |
|---|---|--|

**Time spent (in minutes):** 60

## 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 | 150            | 3                        | -                      | 7                        | -                      |
| Impact Analysis  | 90             | 8                        | -                      | 13                       | -                      |
| Actualization    | 60             | 4                        | 1                      | 10                       | 2                      |
| Verification     | 60             | 8                        | -                      | 13                       | -                      |
| <b>Total</b>     | 360            | 8                        | 1                      | 15                       | 2                      |

## 6. Conclusions

To conclude, concept location and Impact Analysis took more time as the code base is large and system architecture is complex and we had to search and analyze for various existing options in jedit to get an idea of the changes and the files that has to be updated and inspected to find the impact. Actualization and Validation were easy comparatively once we figured the files, methods that has to be updated. Verification was pretty comprehensive as we ensured that all features are working as expected using the jedit UI despite the use of testing frameworks like Junit. We relied on IntelliJ for code navigation( we used cmd+shift+f, cmd+f, shift shift, cmd+[, cmd+]) and modification. Overall The change request process was smooth and we ensured the integrity of the system throughout.