

# CS/SE 6356 Software Maintenance, Evolution & Re-engineering Spring 2024 Assignment 2: jEdit changes

Due date: 02/28/2024 (end of day)

**This is a team assignment. You are allowed to exchange information with other teams about installation, but NOT about the changes.**

## 1 Goals of the assignment

In this assignment, you are to implement two change requests for jEdit, a free and open-source text editor.

The source code for jEdit-5.5.0 and the installation instructions are available in the following OneDrive folder:

<https://cometmail.sharepoint.com/:f/s/CS6356001SE6356001SYSM63080012242/Enljz9Z8jFJlub9teSQo3d8B-v7obIYJD6pYg5Fh11xGQQ?e=eFL2TQ>

Those instructions were based on the IntelliJ IDE, but you could use any other IDE you are familiar with, such as, Eclipse or NetBeans. The README file in the zipped source includes tips about using Eclipse and Netbeans.

Read carefully the assignment before proceeding.

## 2 Change requests

Each team will have to make all three of the following change requests and implement them in jEdit.

### 2.1 Change request #1

The left side of the *status bar* of jEdit reports: the line number containing the caret, the column position of the caret, the character offset of the caret from the beginning of the file, and the number of characters in the file (see Figure 2). You are requested to modify the status bar to show: the word offset of the caret from the beginning of the file and the number of words in the file. As a result of this change, the status bar for the example of Figure 2 should report: "2,6 (76/138886)(12/23731)".

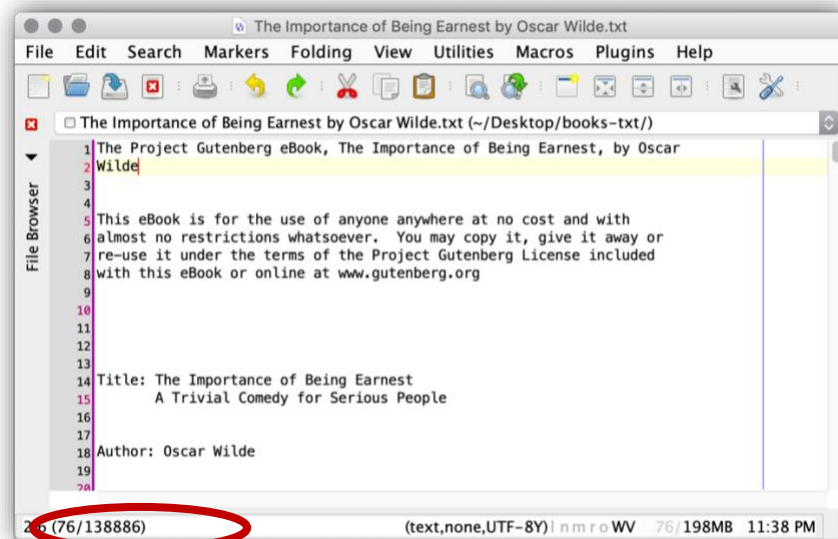


Figure 1. Caret position information in jEdit

## 2.2 Change request #2

In the **File » Recent Files** main menu of jEdit, the text box on top of the recent files list allows to highlight recent files names that match with a given string (see Figure 2). The string in the text box should match all the files that contain it anywhere in their name. However, the highlight works only when the string matches the beginning of a file name. You are requested to modify this feature so that the highlight occurs for the cases when the string is contained anywhere in the file name.

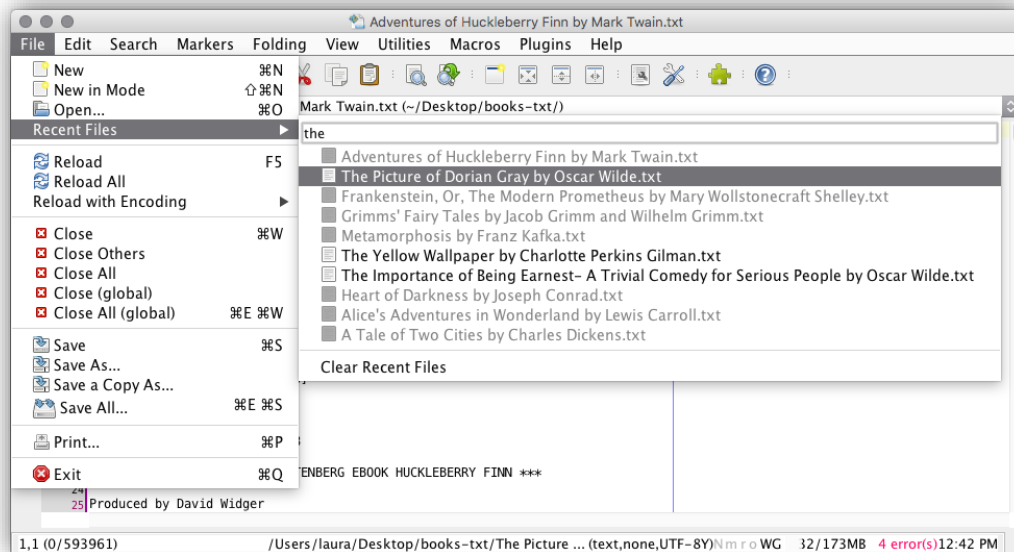


Figure 2. Recent Files menu in jEdit

## 2.3 Change request #3

Currently, jEdit displays the horizontal and vertical scroll bars (see Figure 3) whenever the content of the opened document exceeds the size of the editor. Implement an option in the **View** menu that allows to hide the scrollbars.

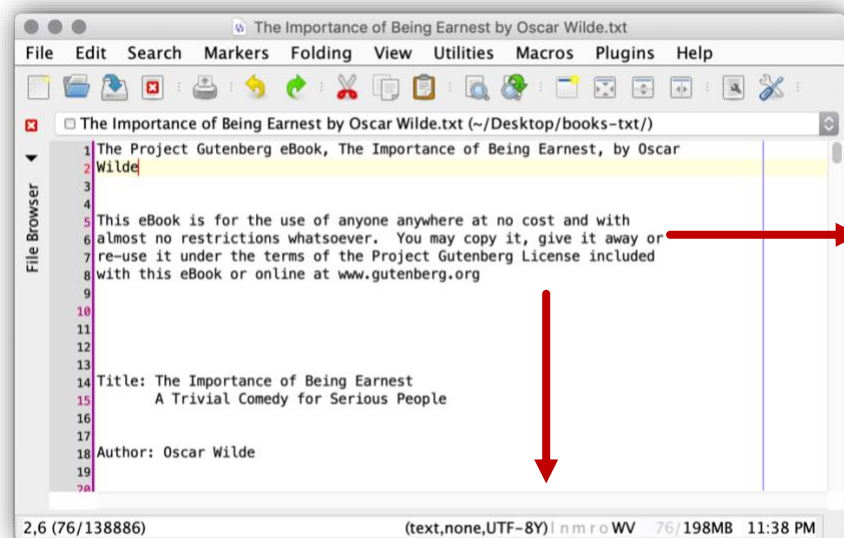


Figure 3. Scroll bars in the jEdit text editor

### 3 What are you supposed to do?

1. Create a private GitHub repository called JeditTeamXX, where XX is your team number. If you do not have a GitHub account, please get one. Share the repository with *mohPYdev* and *ChaoticCross12*.
2. Copy the jEdit source code from the shared OneDrive folder, mentioned above.
3. Follow the instructions from the Mango Installation Guide.pdf and make sure you can compile and run the software.
4. Run the system to familiarize yourself with its functionality.
5. Do the changes for this assignment.

For each change request, complete the following steps:

6. Read and understand the change request. Ask any question you may have about the change requests to the TA. Last-day questions are strongly discouraged and will not help you finish the assignment on time.
7. Create a branch in the repository called "jedit-cr-<#>", where <#> is the change request number. For example, for change request #3, the branch should be called "jedit-cr-3". After you make your change, with possible multiple commits in the branch, merge the change branch into the main branch. This way each change can be analyzed independently.
8. You will have to work together. Follow the change process covered in class (i.e., concept location, impact analysis, actualization, etc.) to implement the change request. Prefactoring and postfactoring are optional. During the change, one team member will change the code in the IDE while the others will track the activities and record the major decisions taken by the team using a change request log (see the log template for specific instructions). Feel free to decide who changes the code and who tracks the process. Ideally, all of the team members should alternate their roles for each change request. In order for this to work, you will have to discuss out loud each decision. You can do this in person or via MTeams or other virtual collaboration tool. Do not work asynchronously, meaning that one student makes the changes and then later the other (or the same one) completes the log. This will result in incomplete logs and defeats the purpose of the later assignments, where you will analyze the logs produced by all teams.
9. After you make your change, perform functional testing by executing several scenarios related to the features affected by the change. Make sure you define a set of test cases (i.e., usage scenarios) and keep track of the ones that fail and pass. Track the testing activity in the change request log. If you want to add unit test, feel free to do so, but it is optional.
10. After you implement and test the change request, push your local changes to the remote GitHub repository. Feel free to make several commits to your local working copy before pushing.
11. Name the log file as follows: jedit-cr#X-teamYY.pdf, where X is the number of the change request and YY is the number of your team. You may submit the log as docx or pdf.

### 4 Deliverables

There are three types of deliverables for this assignment:

1. **Modified source code of the system in GitHub.**
  - a. Make sure you have one separate branch for each change request
  - b. Remember to add the TAs as collaborator of your repositories (GitHub accounts: *mohPYdev* and *ChaoticCross12*)
2. **Change log for each change request.** Consider the next instructions:
  - a. Fill-in the change log form for each implemented change request while implementing the change requests
  - b. Add the log forms to your GitHub repository.
  - c. Submit the logs through eLearning (one log file per change request)

3. Contribution information. Add another file where you include the name of each team members and how much they contributed to this assignment (in percentages). For example: John Doe 40%; Jane Doe 30%, Mark Doe 30%. The percentage should add up to 100%. Note. Read the information on eLearning about how the assignments will be graded.

## 5 Notes

- The logs should not contain any gap in the process you followed. Provide enough details about queries, tools used, classes visited, decisions, etc. Any person should be able to precisely understand and replicate the process you followed. It is very important to clearly describe the rationale of your tasks and decisions.
- Do not forget to evaluate the process you went through and provide enough details in the conclusions in the change request logs. Include details on why some steps were harder than others.
- Proofread carefully your logs.

## 6 Grading

Item	Points
Repository and branch creation	10
Code changes, correct code compilation, and correct change requests implementation	50
Code change description (in the change log)	40
<b>Total</b>	<b>100</b>

**Notes:** Proofread carefully your scenario descriptions and change request logs. You will receive points off in case of writing mistakes, gaps in the logs, compilation errors.