

## Task: Autosave

This task involves planning a change to jEdit's autosave feature. You will need to first familiarize yourself with JEdit and one of its features: *autosave*.

### Overview of JEdit, Buffers, and Autosave

1. To run JEdit, right click on  
    `org.gjt.sp.jedit`  
    `JEdit.java`  
    and select: Run As -> Java Application. Do not close JEdit.
2. In JEdit, an opened text file is called a *Buffer*. The following is an extract from the JEdit Manual (section 2.1):

“Several files can be opened and edited at once. Each open file is referred to as a *buffer*. The combo box above the text area selects the buffer to edit. Different emblems are displayed next to buffer names in the list, depending on the buffer's state; a red disk is shown for buffers with unsaved changes, a lock is shown for read-only buffers, and a spark is shown for new buffers which don't yet exist on disk.”

3. JEdit has an *autosave* feature. The following is an extract from the JEdit Manual (section 3.3.1):

“The autosave feature protects your work from computer crashes and such. Every 30 seconds, all buffers with unsaved changes are written out to their respective file names, enclosed in hash (“#”) characters. For example, `program.c` will be autosaved to `#program.c#`.”

JEdit will also generate *backup* files, which are terminated with a tilde (~) character. These have nothing to do with your task in this study. You can completely ignore them.

Saving a buffer automatically deletes the autosave file, so they will only ever be visible in the unlikely event of a JEdit (or operating system) crash.

If an autosave file is found while a buffer is being loaded, jEdit will offer to recover the autosaved data. The autosave feature can be configured in the **Loading and Saving** pane of the **Utilities>Global Options** dialog box.

4. In the Loading and Saving pane, set the autosave frequency to 5 seconds.
5. Open the file `C:\Temp\test.txt`
6. Add a character to the file and do not save the file.
7. Look in `C:\Temp`. You should see the autosave file.
8. Save the test buffer in JEdit. The autosave file should disappear.
9. Add a character to the test buffer and do not save it. Wait 5 seconds.
10. Kill JEdit using the terminate button on the Eclipse console (the button with the red square).

11. Launch jEdit again. JEdit will attempt to recover the autosave file. Click yes. **ATTENTION:** A bug in the code of JEdit will cause the program to hang if you do not click yes or no in the recovery dialog before the time in the autosave frequency. To avoid this, just click yes or no before the 5 seconds (or whatever) of the autosave frequency are over. If the program hangs, you can kill it using the terminate button on the console. You do not have to worry about this bug for the study.

From a user perspective, that's all there is to the autosave feature. You can close JEdit now.

## Change Request

You are to create a plan for performing the task described below. The plan should include the relevant program elements that need to be changed and how they should be changed. **NOTE:** You are not actually required to perform the changes. Rather you should identify the particular classes and methods to be used and describe any new classes or methods required. Use a text file (in e.g. Wordpad) to record your plan.

### **Change Task:**

Modify the application so that the users can explicitly disable the autosave feature. The modified version should meet the following requirements:

1. jEdit shall have a checkbox labeled "Enable Autosave" above the autosave frequency field in the Loading and Saving pane of the global options. This checkbox shall control whether the autosave feature is enabled or not.
2. The state of the autosave feature should persist between different executions of the tool.
3. When the autosave feature is disabled, all autosave backup files for existing buffers shall be immediately deleted from disk.
4. When the autosave feature is enabled, all dirty buffers should be saved within the specified autosave frequency.
5. When the autosave feature is disabled, the tool should never attempt to recover from an autosave backup, if for some reason an autosave backup is present. In this case the autosave backup should be left as is.

### **During the task:**

1. You must make **no change** to the source code. You are not allowed to perform temporary changes, or try out different alternatives.
2. Do not use the debugger.

## Expert Knowledge

The starting point:

A checkbox should be added to `org.gjt.sp.jedit.options.LoadSaveOptionPane` to enable/disable the autosave.

Please notify the investigator when you are ready to commence.