

# Change request log

## 1. Concept Location

Step #	Description	Rationale
1	We ran the system and checked out the code repository folders	
2	We checked out the file menu of the system and the target option to be changed, case in point – “Recent Files”	To get familiar with the issues that is directed in the Change request 2.
3	We then searched for the query “recent files” in the regular expression matcher in VSCode	As this was the name of the file menu to be worked on, it seemed like an appropriated search query to find files.
4	From quite a few results, and skimming the file names, we opened a file called “RecentFilesProvider.java”	We selected this file because it gave direct hint that this file is responsible for handling the recently opened files.
5	We started going through the main class “RecentFilesProvider”	While going through the code we kept in mind we are looking for a textfield or a keyboard input or a usage of another class that points to keyboard input.
6	Luckily we found the keyboard listeners being used in this class and we marked this class.	This is the end of concept Location as we marked this class as located and marked.

**Time spent (in minutes):** 20

List the classes and methods that you inspected.

Classes and methods inspected:

- org\git\sp\jedit\menu\RecentFilesProvider.java/RecentFilesProvider
  - void update(menu)

## 2. Impact Analysis

Step #	Description	Rationale
1	We made a list of methods which called RecentFilesProvider in some form or the other. The list had only 2 other files/classes.	To track the classes that could be impacted by the change.
2	We inspected both these files, <ul style="list-style-type: none"><li>1. BufferHistory.java</li><li>2. GeneralOptionsPane.java</li></ul>	On inspecting them we realized none among these needs to be changed since they are just using the properties of RecentFilesProvider and doesn't get concerned with the change in search functionality.
3	Thus both these classes were then discarded and potentially no change in other classes was required.	

**Time spent (in minutes):** 10

List the classes and methods that you inspected.

Classes and methods inspected:

- org\git\sp\jedit\BufferHistory.java
- org\git\sp\jedit\options\GeneralOptionPane.java

### 3. Actualization

Step #	Description	Rationale
1	Initially we suspected a wrong pattern of regex would have been generated using the globToRE of the class "StandardUtilities" but it checked out fine.	Since the regex was generated using this function this was one of the location that may have been faulty.
2	Once it checked out we inspected the code further and found out that the method update had a pattern matching problem since the regex was correct.	
3	We then changed the pattern matching syntax and changed the method "match" to method "find".	Since match would only care if the search term is at the start and exit the function after it. Find would look for the search key in the entire term and return true or false.

**Time spent (in minutes):** 25

List the classes and methods that you inspected and separately the ones you changed.

Classes and methods inspected:

- Org/gjt/sp/util/StandardUtilities.java
  - String globToRE(string)
- org/gjt/sp/jedit/menu/RecentFilesProvider.java/RecentFilesProvider
  - void update(menu)

### 4. Validation

Step #	Description	Rationale
1	Test case 01: Inputs: We opened 4 files with filename jumbled with one common keyword in all. Expected output: the common keyword should be picked up and all files should be selected.	This was the expected behavior and it passed.
2	Test case defined 02: Inputs: we opened the same 4 files, but the search term used was not present in any of the file names. Expected output: None of the files should be selected/highlighted	This test suggests that searching works perfect and all edge cases are covered.  This test passed as well.

**Time spent (in minutes):** 10

### 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	20	1	1	1	1
Impact Analysis	10	2	0	2	0
Actualization	25	2	1	2	1
Verification	10	0	0	0	0
<b>Total</b>	65	5	2	5	2

## **6. Conclusions.**

The concept location being straightforward, targeting the location was easy along with the impact analysis since not many files were linked to this and neither were any of them affected by this change request.

The challenge we faced here was understanding the working of regex expressions and java syntax for the same. We took a few minutes to debug the code for regex. Post which it was an easy process of actualization and verification using dummy test files with similar – but- jumbled names.