Change request log – ps2

# Team

Binh Doan - developer

# Change Request

The Merge module throws an exception upon attempting to merge page ranges that intersect (see Figure 1). You are to fix this issue by allowing intersection of ranges during the merging operation.

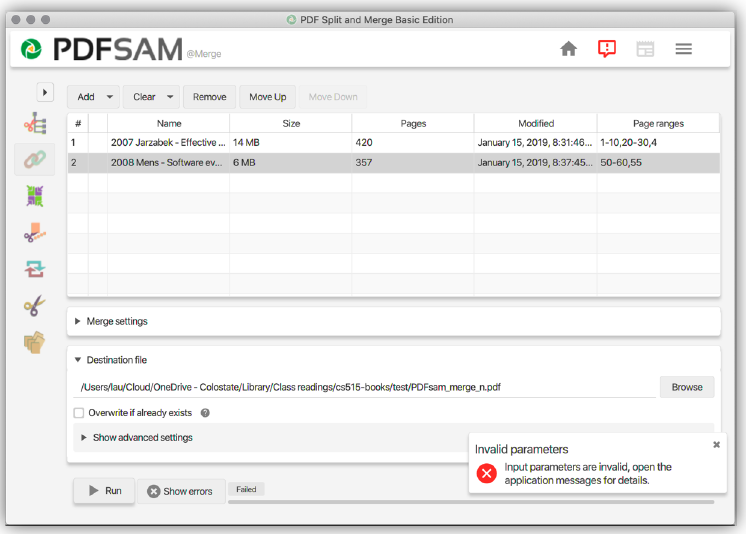


Figure . Merge module of PDFsam—Error when trying to merge overlapping page ranges

# Concept Location

The table below describes each step I perform the concept location for this change request.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | I started the pdfSam v4.0.5 |  |
| 2 | “merge” and “page\*range” are the concepts extracted from the change request. |  |
| 3 | The pdfSam source code is very well organized. I could see the package pdfsam-merge immediately. Within this package, there are only a few files. | Pdfsam source is categorized by its feature. For the change related to pdfsam -merge feature, it should be located under Merge directory. |
| 4 | Search for “page\*range”, I see that only the MergeSelectionPane has the matches. | Reduce the concept locations. |
| 5 | I marked the MergeSelectionPane as “located”. |  |

**Time spent (in minutes):** 5

# Impact Analysis

The table below describes each step I follow when performing impact analysis for this change request.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | MergeSelectionPane class is not found by jRipples. |  |
| 2 | MergeModule is the next class. | This class calls the MergeSelectionPane class. |
| 3 | Discard MergeModule class as it does not have anything related to page range. |  |
| 4 | For the change request, no other classes are impacted. However, a new issue is found similar to the change request (see Actualization step 6). Impact analysis has been done and there are a few classes that need to be modified:   * SelectionTableRowData * ConversionUtils | This is an extra work. |

**Time spent (in minutes):** 5

# Prefactoring (optional)

Not implemented.

**Time spent (in minutes):** 0

# Actualization

The table below describes each step I followed when changing the code.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | There are 2 solutions to solve this issue. One is to fix from the inputs. The other is to fix the process. |  |
| 2 | The process is handled by the library sejda which I cannot modify. Therefore, I have to find a way to fix it from the input perspective. | There are many reasons not to modify a library:   * Library is used by many other applications. Modifying it will cause a big impact. * Library is treated as external source that we usually do not have control of. |
| 3 | I debugged by writing to file everything within this class. | This class has only two methods. |
| 4 | When testing with the application, I see that instead of putting page ranges like [1-5,2], I can do [1-5] for the file, and then [2] for another input of the same file.  Applying that trick to the code, instead of parsing [1-5,2] to the buider before sending to sejda, I separated 1 input file into 2 input files. One with the page range [1-5]. The other is with the page range [2]. |  |
| 5 | I tested and it worked as expected. |  |
| 6 | However, I concerned about another case that has the page range as [1,2,3,2]. Even though this case is not covered in the change request. The issue is similar, so I decided to fix it. | [1,2,3,2] is similar to [1-3,2]. Both should work without issues. |
| 7 | While debugging, I see that the input [1,2,3,2] is reduced to [1,2,3] before added to the builder. It is because of the definition of “Set”. To fix this, I changed it to List. |  |
| 8 | I added a similar methods to the existing ones, but using List instead of Set. | This is to minimize the impact by not changing the old utility methods. |
| 9 | I tested and it worked perfectly as expected. The merged file contains pages 1,2,3 and 2. |  |

**Time spent (in minutes):** 45

# Postfactoring (optional)

Not implemented.

**Time spent (in minutes):** 0

# Validation

The table below describes any validation activity (e.g., testing, code inspections, etc.) I performed for this change request. Include the description of each test case, the result (pass/fail) and its rationale.

**Make sure you time yourselves when going through this process and provide the total time spent below.**

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | Test method: Unit Test  Test case defined: regression test using the original unit test  Inputs:   * Page range [1,3-10]   Expected output: no error and 2 input page selections. | This is the regular expected behavior.  The test passed. |
| 2 | Test method: Unit Test  Test case defined: new unit test per change request  Inputs:   * Page range [1-5,3]   Expected output: no error and 2 input page selections. | This is the regular expected behavior.  The test passed. |
| 3 | Test method: test the GUI directly with the running application.  Test case defined: A regular merge with 2 selections  Inputs:   * A 8-page pdf file * Page range [1-5,3]   Expected output: merged file with pages 1,2,3,4,5 and 3 | This is the regular expected behavior.  The test passed. |
| 4 | Test method: test the GUI directly with the running application.  Test case defined: A regular merge with 3 selections  Inputs:   * A 8-page pdf file * Page range [1-5,3,2]   Expected output: merged file with pages 1,2,3,4,5,3 and 2 | This is the regular expected behavior.  The test passed. |
| 5 | Test method: test the GUI directly with the running application.  Test case defined: A regular merge with 3 selections  Inputs:   * A 8-page pdf file * Page range [2,3,2,5]   Expected output: merged file with pages 2,3,2 and 5 | This is the regular expected behavior.  The test passed. |

**Time spent (in minutes):** 40

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 5 |
| Impact Analysis | 5 |
| Prefactoring | 0 |
| Actualization | 45 |
| Postfactoring | 0 |
| Validation | 40 |
| Total | 95 |

# Reverse engineering

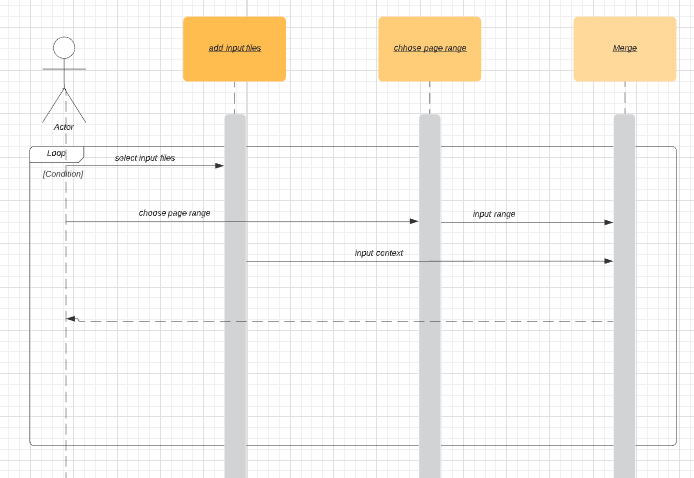


Figure 2. UML sequence diagram

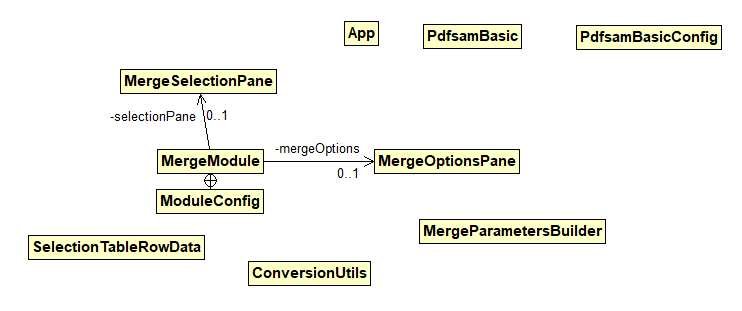


Figure 3. UML class diagram shows the visited classes

# Conclusions

The change process has been done within a short amount of time as following the change procedure. Concept location was done by using Eclipse IDE’s file search tool. Impact analysis was done with the support of both file search tool. It took me some extra time to learn and understand completely the lambda expression used in the source code. It also took me some extra time to write a new unit test.

The classes I have changed:

* MergeSelectionPane
* SelectionTableRowData (for extra bug fix)
* ConversionUtils (for extra bug fix)