Change request log – ps3

# Team

Binh Doan - developer

# Change Request

The Rotate module allows to rotate the pages of multiple PDF documents by:

(i) Providing a range of pages to rotate, which can be done for each document; or

(ii) Selecting even, odd or all pages to rotate for those documents whose range was not defined.

These two options are exclusive, i.e., the even/odd/all-pages selection is disregarded whenever the page range is

specified for a document. For example, the results of executing the task depicted in Figure 1 are:

- For document #1, a pdf file consisting of 371 pages, where all even pages are rotated 90º clockwise.

- For document #2, a pdf file consisting of 612 pages, where the first 10 pages are rotated 90º clockwise.

- For document #3, a pdf file consisting of 420 pages, where pages 20 to 100 are rotated 90º clockwise.

You are requested to modify this feature to consider both the page-range and the even/odd/all options in

conjunction when executing the job. This means that for the task depicted in Figure 1, the results should be:

- For document #1, a pdf file consisting of 371 pages, where all even pages are rotated 90º clockwise.

- For document #2, a pdf file consisting of 612 pages, where all even pages in the range 1 to 10 (i.e., 2, 4, 6, 8, 10)

are rotated 90º clockwise.

- For document #3, a pdf file consisting of 420 pages, where all even pages in the range 20 to 100 (e.g., 20, 22, 24,

…) are rotated 90º clockwise.

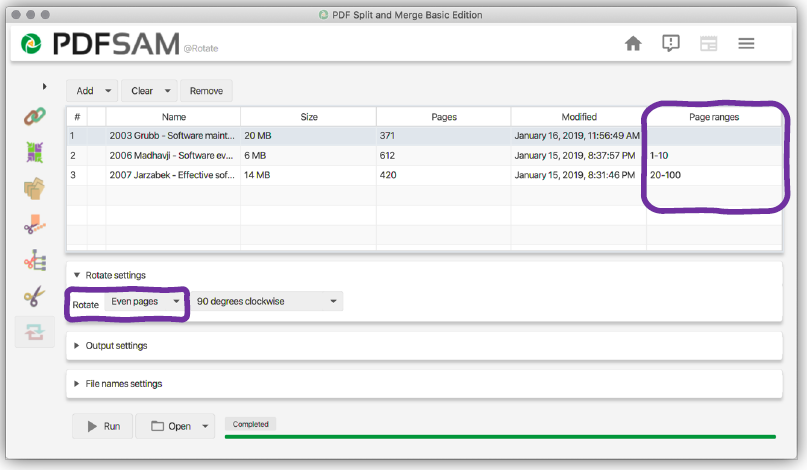


Figure . Rotate module of PDFsam—Page selection options

# 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 | “rotate” 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-Rotate immediately. Within this package, there are only a few files. | Pdfsam source is categorized by its feature. For the change related to Merge feature, it should be located under pdfsam-Rotate directory. |
| 4 | Searching for “page\*range”, I saw that only the RotateSelectionPane and RotateParametersBuilder classes have the matches. | Reduce the concept locations. |
| 5 | I debugged the RotateSelectionPane and I saw that the result was as expected that the page range is added to the builder. I do not see the option even/odd page range here. Therefore, I discarded this class. |  |
| 6 | Odd/even pages are selected by the variable predefinedRotationType. The variable is used in addInput() method together with the page range. Therefore, I marked this method of the class RotateParametersBuilder as “located”. |  |

**Time spent (in minutes):** 15

# Impact Analysis

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

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | RotateParametersBuilder class cannot be not found by jRipples. | I had to do the impact analysis manually. |
| 2 | RotateModule, RotateOptionsPane and RotateSelectionPane are the next classes. | This class calls the RotateParametersBuilder class. |
| 3 | AddInput() is called in RotateSelectionPane class. However, the impact in this class is expected. Therefore all next classes are marked as “unimpacted”. |  |

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

# 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 | The current implementation shows that either predefinedRotationType or page ranges is accepted. |  |
| 2 | Combining these two features together will satisfy the change request. |  |
| 3 | Within the PredefinedSetOfPages, there is a includes() method to check is the input pages are odd or even. I used this method to save some coding effort. |  |
| 4 | I tested and it worked as expected. |  |

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

# 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: test addInput() method  Inputs:   * pageSelection [1,3] * predefinedRotationType ALL\_PAGES   Expected output: rotate pages 1,2,3 | This is the regular expected behavior.  The test passed. |
| 2 | Test method: Unit Test  Test case defined: test addInput() method  Inputs:   * pageSelection [1,5] * predefinedRotationType ODD\_PAGES   Expected output: rotate pages 1,3,5 | This is the regular expected behavior.  The test passed. |
| 3 | Test method: Unit Test  Test case defined: test addInput() method  Inputs:   * pageSelection [1,5] * predefinedRotationType EVEN\_PAGES   Expected output: rotate pages 2,4 | 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 rotation  Inputs:   * An 8-page pdf file * Page range [2-6] * Rotate Options: Odd   Expected output: Page 3 and 5 of the file are rotated. | 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 | 15 |
| Impact Analysis | 10 |
| Prefactoring | 0 |
| Actualization | 20 |
| Postfactoring | 0 |
| Validation | 40 |
| Total | 85 |

# Reverse engineering

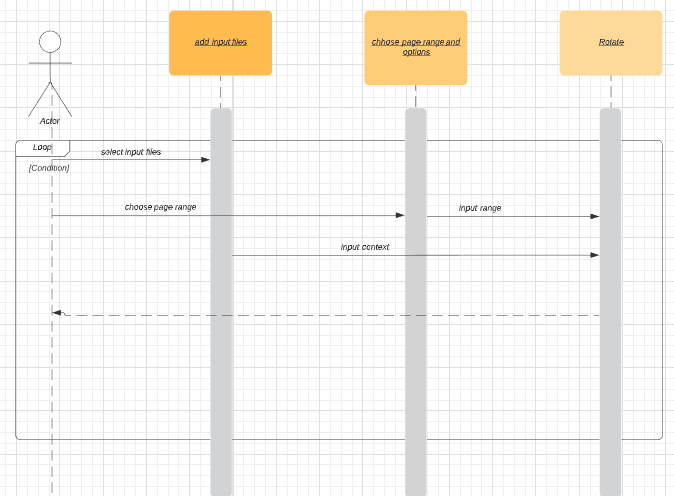


Figure 2. UML sequence diagram

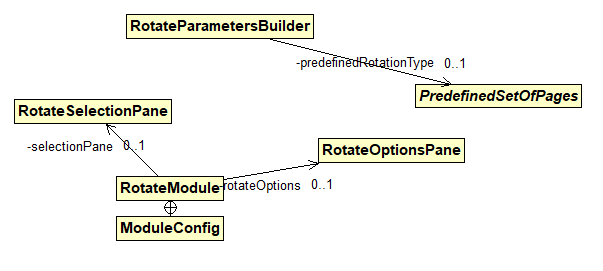


Figure 3. UML class diagram shows the visited classes

# 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 the unit test.

The classes I have changed:

* RotateParametersBuilder