Change request log

Team

Specify the team members.

Brad Jakobitz

Nate Olson

Change Request
Provide the id and description of the change request.

Concept Location

Step #	Description	Rationale
1	Ran PDFsam, launched the Alternate Mix module and attempted to reverse a single document	we wanted to see exactly what happened when we reproduced the issue and wanted to make sure we was able to reproduce the issue
2	Error messaged shows "Input parameters are invalid, open the application messages for details" so we opened the application messages to figure out what the message is. The message states "Task execution failed due to invalid parameters: At least two elements required" and it also contains a stack trace. The lowest part of the stack trace that contains PDFsam code is TaskExecutionController.lambda\$request\$0 (TaskExecutionController.java:87)	Tracking down exactly what the messages say normally give me a good idea of what to search in the code to figure out which source files we need to examine first. In this case though, since the message gave me a stack trace which contained a pointer to a source file and a line number it was even more helpful.
3	Closed PDFsam, opened PDFsam in Eclipse, put a break point in TaskExecutionController.java on line 87 (where the stack trace pointed to). We then re-ran PDFsam in debug mode, reproduced the issue again and this time hit the break point.	The stack trace in the messages was a very good first start, but at this point, we needed to dive a little deeper and debugging typically allows this. we was right as we was able to look a little deeper into the call stack and see that in AlternateMixModule it has a pointer to BaseTaskExecutionModule which sets up the broadcast of TaskExecutions. In the 'initModuleSettingsPanel' method of 'BaseTaskExecutionModel.java' there is also a String on error that says "Invalid Parameters" which is what we am seeing when the error comes up, so we think we am on the right track.
4	Looking further into the debugging stack we was able to see that the 'initModuleSettingsPanel' sets what the run button at the footer does during events. It calls the 'getBuilder' method of the same class to build the event and if there is an error on build this is when it displays the errors, so we opened the 'getBuilder' method that it calls to get the builder to build the event and it	we need to figure out what builder is being built with the Alternate Mix module in order to determine why the error is happening and where it is coming from.

	opened and abstract method in 'BaseTaskExecution.java'	
5	We did an 'Open Call Hierarchy' on "getBuilder" method to see what non-abstract classes implement this method. The AlternateMixModule was in the list, so we opened the 'getBuilder' method of the AlternateMixModule	Same reasoning as the previous step.
6	The 'getBuilder' method of the AlternateMixModule creates and returns an 'AlternateMixParametersBuilder' class so we opened that to see what it looks like. The AlternateMixParametersBuilder class extends the AbstractPdfOutputParameterBuilder with a template class of AlternateMixMultipleInputParameters so we opened that file.	We need to figure out where the restriction of not allowing a single document came from. As we dove further down the list, when we found something that had "MultipleInputParameters" in the name, we thought this was a good place to look.
7	The 'AlternateMixMultipleInputParameters' class has a private member variable of 'inputList' that has an annotation on it of '@AtLeastTwo' – which looks like our problem. When we opened this file we see that the '@AtLeastTwo' annotation class as an annotation of '@Size(min=2)' which has a default message of 'At least two elements required' which is exactly the detailed message we see in PDFsam so this would explain why we see this issue.	We have now tracked down why we need two inputs to the AlternateMixModule and we cannot reverse just a single document.
	After returning when we realize AlternateMixMultipleIn	_
8	<u> </u>	If we cannot change the class (which, we believe if we were actually making this fix in
9	Looking at the 'AlternateMixMultipleInputParameters' class we see it extends 'AbstractPdfOutputParameters' class and implements a 'SingleOutputTaskParameters' interface. we am looking through other org.sejda.model code to see if anything else looks like it could be effective for our situation	
10	we decided to first look at other classes which implement the 'SingleOutputTaskParameters' interface	We still want a single output to be a requirement so looking at other classes that support this will narrow down my focus of what could be used as an alternative.
11	'MultiplePdfSourceSingleOutputParameters' class looks like it could be useful. It implements the same 'SingleOutputTaskParameters' interface and it extends the MultiplePdfSourceParameters class. It looks like the 'MultiplePdfSourceParameters' class does not have the same restriction on requiring 2	we am still trying to find a valid substitute in

	inputs so this could work.	
12	'MultiplePdfSourceParameters' class is abstract so we need to see what implements it. There is a class 'CombineReorderParameters' extends it and, after reading its Javadoc, looks like it could be a valid substitute. we will try this class in place of 'AlternateMixMultipleInputParameters' as a substitute for our AlternateMixBuilder	Finding a class in the same org.sejda.model package that will work given our input and output requirements is the best, lowest impact change we could make. We believe we have found a class that may work, so we will try.
	After not using 'CombineRe	
13	Decided not to use CombinReorderParameters as this deals with pages and not files. We do not believe this is the best approach either so we will create my own class 'AlternateMixInputParameters' in the pdfsamcore/org.pdfsam.support.params directory and package so it should be usable across multiple pdfsam projects. This class will be almost identical to the 'AlternateMixMultipleInputParameters' except without the '@AtLeastTwo' annotation on the inputList variable.	We do not believe changing from documents to pages is smart because it could have a very high impact for the base use case of alternating pages of documents. It is likely this would be a very high impact change and would require us to refactor all of the code that deals with the typical AlternateMix use case. Since our change request is an enhancement request, it should be treated as such and we should not risk breaking the base use case to make this change.
		Note: Again, if we was actually making this change in real life, and this was a required change request, we would most likely be adding this file to the org.sejda code or modifying the existing file.
	After getting error in org.sejda co AlternateMixInputPara	• • •
14	Since we cannot change what type of parameters we am using as this has to be set by org.sejda code, we am switching my changes to AlternateMixInputParameters all back to AlternateMixMultipleInputParameters and deleting the AlternateMixInputParameters that we created.	We need to revert all of my changes since
15	Since the AlternateMixMultipleInputParameters requires 2 files and we am required to just allow a single file, we will have to write application code to allow an exception for this.	We do not like this solution and this will be an ugly solution. But without being able to change the org.sejda code and because and error is thrown in org.sejda when we try to set a template builder class in the application – we have no choice, but to leave the template builder class coming from the existing org.sejda code and add application code to deal with this situation. This should not be done in real life, but we do not know what other choice we have since we cannot modify the org.sejda code.
16	In 'AlternateMixSelectionPane' where the AlternateMixParametersBuilder gets called, we am going to add a test of whether there is a single input file and if reverse is selected. If and only if these two things are true, we will add a dummy second input in such a way to get around the AlternateMixMultipleInputParameters requiring two input documents.	we am not exactly sure yet what this "dummy" second input needs to be (we am pretty sure it cannot be empty), but we do know we will need to somehow turn one input into two inputs to work around the requirement of two documents at a lower level.

First Pass: 19 minutes

Second Pass 19 minutes

Third Pass: 6 minutes

Fourth Pass: 10 minutes

Impact Analysis

Use the table below to describe each step you follow when performing impact analysis for this change request. Include as many details as possible, including why classes are visited or why they are discarded from the estimated impact set.

Do not take the impact analysis of your changes lightly. Remember that any small change in the code could lead to large changes in the behavior of the system. Follow the impact analysis process covered in the class. Describe in details how you followed this process in the change request log. Provide details on how and why you finished the impact analysis process.

Ste p#	Description	Rationale
1	My biggest concern before we even get started on Impact Analysis is that if we change the AlternateMixMultipleInputParameters class to allow a single input there are a few potential issues a) The name 'MultipleInput' will no longer be correct b) We only want to allow the reversal of a single document. What will happen if "Reverse" is not checked and we am allowing a single input? Will it simply output the same document? Is this acceptable?	The change request does not necessarily address some of these issues and if we were working directly with a customer or with instructions from a boss we would be able to make sure these side-effects would not be an issue before we started.
2	Starting at the 'AlternateMixMultipleInputParameters' class, we ran an 'Open Call Hierarchy' to see where else in the code this class gets used. The chain of uses looks as follows: c) AlternateMixMultipleInputParameters o AlternateMixParametersBuilder • AlternateMixParametersBuilderTest • BaseTaskExecutionModule. initModuleSettingsPanel -> EventHandler	Seeing where the class we am changing is being used gives me a very good indication of where we need to start looking to see what is going to be impacted.
3	The first class we decided to look at was the AlternateMixParametersBuilder. The only place AlternateMixMultipleInputParameters class is being used is during the build() method.	This is the only class that directly uses the class we are changing so this is the first place to look. If this class is not impacted, we likely do not need to worry about any other classes.
4	AlternateMixParametersBuilder should not be impacted.	Its build method should still return the same AlternateMixMultipleInputParam eters object (or a renamed version). It is the parameter itself that will be modified during the building for a run that will need to be checked.
5	Even though the only class using AlternateMixMultipleInputParameters should not be impacted, we	We know, from previous

decided to look BaseTaskExecutionModule.initModuleSettingsPanel EventHandler. We do not believe this will need to be changed because it just ensures the parameters are set to what we expect after a successful build. There is no test for ensure a single file fails or anything like that would be required to change after this. It is possible we should add a new test to ensure our single document use case functions as expected. AlternateMixParametersBuilderTest looks like it does not need to be changed because it just ensures the parameters are set to what we expect after a successful build. There is no test for ensure a single file fails or anything like that would be required to change after this. After returning when we realized we cannot change AlternateMixMultipleInputParameters Since we am no longer changing the inner workings of a class, but instead completely changing classes, we will need to re-
to be changed because the builder should no longer report an error and it should continue as usual. 6 The next place that could possibly be impacted is the testing. AlternateMixParametersBuilderTest looks like it does not need to be changed because it just ensures the parameters are set to what we expect after a successful build. There is no test for ensure a single file fails or anything like that would be required to change after this. After returning when we realized we cannot change AlternateMixMultipleInputParameters 7 Since we am no longer changing the inner workings of a class, but instead completely changing classes, we will need to re-
AlternateMixParametersBuilderTest looks like it does not need to be changed because it just ensures the parameters are set to what we expect after a successful build. There is no test for ensure a single file fails or anything like that would be required to change after this. After returning when we realized we cannot change AlternateMixMultipleInputParameters Since we am no longer changing the inner workings of a class, but instead completely changing classes, we will need to re-
be changed because it just ensures the parameters are set to what we expect after a successful build. There is no test for ensure a single file fails or anything like that would be required to change after this. After returning when we realized we cannot change AlternateMixMultipleInputParameters Since we am no longer changing the inner workings of a class, but instead completely changing classes, we will need to re-
After returning when we realized we cannot change AlternateMixMultipleInputParameters 7 Since we am no longer changing the inner workings of a class, but instead completely changing classes, we will need to re-
7 Since we am no longer changing the inner workings of a class, but instead completely changing classes, we will need to re-
but instead completely changing classes, we will need to re-
examine the impact as it could potentially be larger now
Since we am changing the AlternateMixParametersBuilder.build method to return CombineReorderParameters class instead of an AlternateMixMultipleInputParameters class. we look at who is calling this method, they are: d) AlternateMixParametersBuilderTest.build e) BaseTaskExecutionModule. Since this is the method that is now changing, we need to know who is calling this exact method instead of just looking at who is using the class.
initModuleSettingsPanel -> EventHandler 9 The test will definitely need to be changed. The test explicitly expects ar
AlternateMixMultipleInputParam eters object to be returned and this will not be the case anymore.
The BaseTaskExecutionModule will not need to be changed When this class calls the AlternateMixParametersBuilder. build method it only expects an "AbstractParameters" class to be returned. Since CombinReorderParameters is already subclass of this everything will be fine.
When looking through the changes that needed to be made for CombineReorderParameters to be used instead of AlternateMixMultipleInputParameters in the actual code that mixes the documents together, we decided this was a bad solution. AlternateMixMultipleInputParameters in the actual code that mixes the documents together, we decided this was a bad solution. AlternateMixMultipleInputParameters class uses pages instead of documents. AlternateMixMultipleInputParameters class uses pages instead of documents.
type of change that could have a high impact for the norma use case should not be performed for an enhancement request that is not likely to be the main use case.

AlternateMixMultipleInputParameters was a bad decision because it was a risky fix for the main use case and could have too high of an impact.

After not using 'CombineReorderParameters'

12 The impact of creating a new 'AlternateMixInputParameters' class will be the exact same switching the а CombineReorderParameters class, except we will now also be document based like existina the AlternateMixMultipleInputParameters list so we will not run into the same underlying impact.

It does not matter that we have to create a new class, it is the same as the change we previously made. All we have to do is ensure that my new class is a subclass of the AbstractParameters class and the impact will be the same as the last attempt.

After getting error in org.sejda code (trying to create new AlternateMixInputParameters class)

The impact of adding new, targeted code exactly where we need this check to be performed will actually be less than any of my other attempts.

Since this is a targeted fix, the new code will only be called if and only if there is a single file provided and the reverse is marked true. Since this is the only use case where the new code will be called, this is the only use case we can possibly break. All of the other standard use cases for Alternate Mix Module will remain unchanged.

Time spent (in minutes): 29

First Pass: 16 minutes

Second Pass: 9 minutes

Third Pass: 2 minutes

Fourth Pass: 2 minutes

Prefactoring (optional)

Step #	Description	Rationale
1	Thought about renaming 'AlternateMixMultipleInputParameters'	At first, we was planning on renaming this to 'AlternateMixSingleInputParameters', but realized this class is not accurate either as it still supports multiple.
		Then thought about 'AlternateMixInputParameters' as a way to not specify 'Single' or 'Multiple' as either of those names could be misleading

This is where we realized that AlternateMixMultipleInputParameters is not a part of the PDFsam project, but is instead included in a dependency (sejda-model-3.2.38.jar) so modifying this class is not possible. We am not returning to 'Concept Location' to look for a different way to make this change request.

After returning when we realized we cannot change

AlternateMixMultipleInputParameters			
After not using 'CombineReorderParameters'			
3	We am not sure whether creating a new class is considered 'Prefactoring' or 'actualization'?	We am going to put it under actualization in this case because the new class is how the change request is being implemented.	
After getting error in org.sejda code (trying to create new AlternateMixInputParameters class)			
4	No prefactoring required for my targeted solution.	Because this solution is directly targeted at our specific use case and new code will be added for this, the existing code does not need to be modified to suit our needs.	

First pass: 6 minutes

Second pass: 2 minutes

Third pass: 2 minutes

Fourth pass: 1 minute

Actualization

Step #	Description	Rationale	
	After realizing we cannot change Alterna	ateMixMultipleInputParameters	
	After not using 'CombineReorderParameters'		
1	We added a new class 'AlternateMixInputParameters' to the org.pdfsam.support.params package of the pdfsam-core project. To do this we basically copied and pasted the 'AlternateMixMultipleInputParameters' class	We want this class to be identical to the existing class we are using with the one exception of the inputList allowing a single document. Because of its similarity, there is not much to	
	and modified: the name, the Javadoc, to not require '@AtLeastTwo' on the inputList and made note in the class Javadoc of where the source came from and the original author.	change other than the naming and to ensure we give the author of the class we copied credit.	
2	we changed the 'AlternateMixParametersBuilder' class to use the new 'AlternateMixInputParameters' class as a template input to both the superclass AbstractPdfOutputParametersBuilder and also the super interface of SingleOutputTaskParametersBuilder	These were all the locations that 'AlternateMixMultipleInputParameters' was being used, so our new class now needs to be used in those same locations.	
3	The changes to the super class and super interface templates requires me to change the return value of the 'build' method to 'AlternateMixInputParameters', which required changing the class being instantiated within that method.	Since we simply copied the AlternateMixMultipleInputParameters we should not have to change the method calls after the instantiation – they should all still be the exact same.	
After getting error in org.sejda code (trying to create new AlternateMixInputParameters class)			

4	In the AlternateMixSelectionPane's 'apply' method, we am going to add two Boolean variables. One to see whether there is only a single document provided and one to see if the 'reverse' flag is checked. Only if both of these Boolean values are true, will we split the pages to be in two separate inputs.	We do not want this targeted solution to affect the normal use case at all. So we are adding these two new checks and only if both of the checks pass will anything change. This way, the typical use case remains 100% unchanged.
5	Only if the two Booleans being checked above are true do we proceed to anything else. We realized that if we are reversing a single document, we will need to get the page ranges that could be provided in the input to AlternateMix.	If a user wants to reverse a single document, but only include certain page ranges we should allow this. If we are not allowing it, we should disable the option of specifying page ranges to make it evident to the user. Instead, we am just going to support this use case.
6	After looking at how the PageRanges in PDFsam work, we realize that open end page ranges require you to tell them how many total number of pages are in your document. We will have to open and get the total number of pages in the document in order to get the correct page ranges.	Because of the way PDFsam is currently implemented, the open-ended page ranges require you sending how many total pages are in the document to return you an answer for open ended ranges. Since we am supporting every page range PDFsam supports, we will have to figure out the number of pages.
7	Added code to open the document and get the number of pages. This required two things: f) A new dependency in the pdfsamalternate-mix pom.xml on org.sejda sejda-sambox g) A new error message of "Could not open the document" to go to the user in the case we cannot open the	Opening the PDF document required a new dependency that our project did not have so we needed to also add that to the pom.xml so it builds correctly.
8	document. Once we get the page ranges, we am planning on going through all page ranges and adding the highest page to my "dummy" document.	inform the user if that happens. Adding the highest page to a "dummy" document should allow me to simply provide this "dummy" document as the first input to the AlternateMix. This will ensure the highest page comes first (which will be true in reverse order) and then the rest of the document will simply be reversed. We decided this was easier than trying to split the two inputs by every other page which was my first thought.
10	We have now split the page ranges and found the highest possible page number based on the total number of documents and the given page ranges. We simply created two identical inputs with the exact same source, the only way they vary is their page ranges.	With the page range ability, the only thing we had to make different for my sources are the page ranges they require. We did not have to copy the file or anything like that to ensure 2 inputs.
11	we have now added the highest page to a first "dummy" input, ensured that highest page is removed from the second "given" input (to not duplicate that page) and added those two inputs to the builder in that order.	Again, giving the highest page as the first input will ensure the document is reversed properly.
12	After testing: We realized that this will still fail if the user specifies just a single page in the page range. we am adding an error message of "Cannot"	Because we require at least 2 inputs and neither input can be empty, in order to be able to execute anything we need a minimum of 2 pages to create 2 non-empty inputs. Since we cannot do this with a single page,

	reverse a single page from a single document" to display to the user (again this will need to be internationalized at some point, but we am not doing that)	we cannot support this use case. We believe this is not a big deal because the user really shouldn't need to "reverse" a single page.
13	After testing: We realized that if a PageRange just contains the highest page, we will turn it into an invalid page range by removing the highest page from the page range (e.g., page range of 2-2 will turn into 2-1). So we am now checking this and if the highest page is the only page in the page range, we am simply removing that page range from the second "given" document.	

First pass: 22 minutes

Second pass: 52 minutes

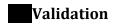
Postfactoring (optional)

Step #	Description	Rationale
	After realizing we cannot change Alterna	ateMixMultipleInputParameters
	After not using 'CombineRe	eorderParameters'
1	We had to change the 'AlternateMixParametersBuilderTest' because it gets the return object from 'build' and explicitly expected a AlternateMixMultipleInputParameters class, so we changed this to be 'AlternateMixInputParameters' instead. After getting error in org.sejda con AlternateMixInputParameters'	during our Impact analysis, but we did not do it as part of the actualization, so we instead did it as part of the post factoring. Dede (trying to create new
2	Because of the amount of code we had to add inside of the 'apply' method of 'AlternateMixSelectionPane', we decided to split this out to be its own method: 'splitDocumentAndAddInput'. This method will now do everything new to better organize the code and will only be called if there is a single document provided with the reversal flag being true.	After looking at the code once we got done implementing, we realized that the 'apply' method was very long and hard to tell exactly what was going on. We decided it would be best to split out the new functionality to make the code more readable.

Time spent (in minutes): 6

First pass: 1 minutes

Second pass: 5 minutes



Step #	Description	Rationale
	ateMixMultipleInputParameters	
1	After not using 'CombineRo Test case defined: h) Ensure we can still mix two documents Inputs: i) Two documents Expected output: j) Alternate mix of two documents (the main use case of this module)	This test failed. There is some error in the core org.sejda code (where this change should be made) so now I am going to have to provide an
	After getting error in org.sejda co	ode (trying to create new
2	AlternateMixInputPara Test case defined: k) Ensure we can still mix two documents Inputs: l) Two documents Expected Output: m) Alternate mix of two documents (the main use case of this module)	Everything worked as expected. We only tested this under normal criteria (no page ranges, no other custom settings, but we did not expect this targeted change to have any affect). This test passed.
	Test case defined: n) Tested reversing a single document and ensured that it was reversed Inputs: o) A single document	The single document reversal worked under no custom settings. This test passed.
	Expected Output: p) The single document in reverse order Test case defined: q) Tested reversing two documents to ensure that still works Inputs: r) Two documents Expected Output: s) The two documents mixed together and reversed.	Again, since we are only executing our code with a single document and if the reverse box is marked, we did not expect this to change, but this tested a "one off" scenario for our code. This test passed.
	Test case defined: t) Tested reversing a single document with a single page Inputs: u) A single document with a single page specified as the range Expected Output: v) The single page	Trial 1: This test failed. We are not able to reverse a single page because of the fact we are required to have two non-empty inputs no matter what. Because of this, we went back to the actualization phase and created an error if a single page from a single document is attempted to be reversed. Trial 2 (after fix from trial 1): I now get a specific error ("You cannot

Test case defined: w) Tested reversing a document with two	reverse a single page from a single document") that tells me why the operation failed. Since this use case is not possible, this is our solution. This test passed. Trial 1: This test failed. It failed due to an invalid
page ranges that each contain a single page Inputs: x) A single document with two individual pages specified as the page ranges Expected Output: y) The two specified pages in reverse order	page range. After debugging we were able to determine this was because we determined which page range contained the highest page and simply subtracted one from that to get the new page range. In this case, that left us with an invalid page range as there was only one page specified. We have fixed this to check if that was the only page and if it was to remove that specified page range completely.
	Trial 2 (after fix from trial 1): This now outputs the two specified pages in reverse of their original order. This test passed.
Test case defined: z) Maven build tests that came with PDFsam Inputs: aa) PDFsam build script	This test passed as the results of this run were exactly the same as the results when we ran this after immediately downloading PDFsam.
Expected Output: bb)The same results as when we ran this test immediately after downloading PDFsam (290 tests run, 1 failure, 0 errors and 6 skipped – failure should be in PrefixFieldTest)	The single failure was in the PrefixFieldTest. The error shown below is what we get when we run these tests (on official PDFsam v3.3.5 branch or in our code).
Test case defined:	
Test case defined: ee) Inputs:	
ff) Expected Output:	
Test case defined: gg) Inputs: hh) Expected Output:	

First pass: 2 minutes

Second pass: 14 minutes

Timing

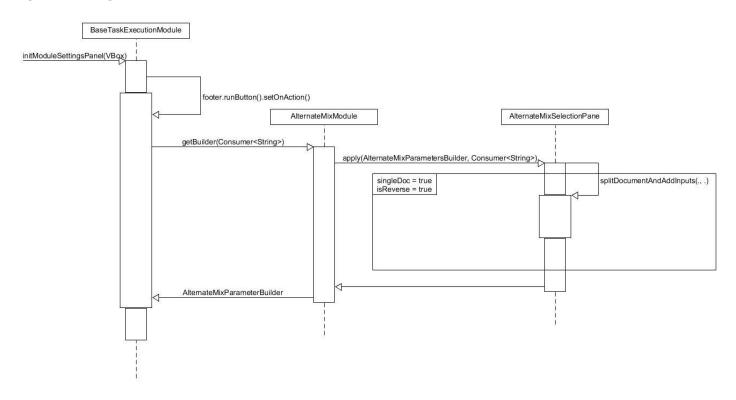
Summarize the time spent on each phase.

Phase Name	Time (in minutes)
Concept location	54
Impact Analysis	29
Prefactoring	11
Actualization	74
Postfactoring	6
Verification	16
Total	190

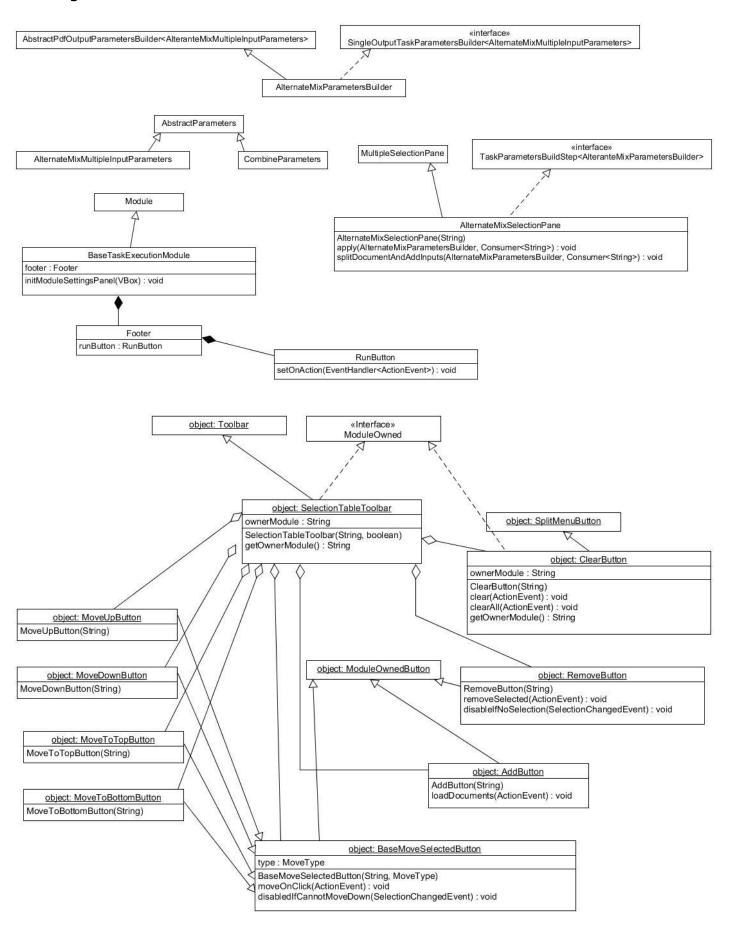
Reverse engineering

Note: The class diagrams from the first change request was completely unrelated, but is included here as per instruction. Because it was completely unrelated, there are no connections between the two class diagrams.

Sequence Diagram:



Class Diagrams:



Conclusions

Although I implemented the change request and it works as specified, I am not happy with the way I was forced to make this change. I believe this change request should have been made at the org.sejda package level where the requirements for AlternateMix Input of PDF files is. Having to write almost 100 lines of code to allow a single document to be reversed, in a module whose purpose is not to reverse documents, seems like a poor solution. If I was able to edit AlternateMixMultipleInputParameters class to remove the '@AtLeastTwo' it would have been a single line fix and it would have worked for everything. Instead, we now have a corner case for the AlternateMix module that must be supported and many lines of code to support a single use case instead of making a more generic solution. I believe that if this change request was required in real life that it would actually be handled in one of three ways:

- a) The author would go and change the org.sejda code (seeing how the author of PDFsam is also the author of org.sejda)
- b) The change request would be rejected because it is an enhancement request and not a defect request. The author would reject it based on the requirement of the Alternate Mix module requiring two inputs and not being intended to reverse a document.
- c) The author would provide a new "Reverse" module that allows reversing documents instead of hacking a solution to allow reversal of documents into a module that is not intended to be used solely for reversing a single document.

Classes and methods changed:

- pom.xml to the pdfsam-alternate-mix project
- AlternateMixSelectionPane
 - void apply(AlternateMixParametersBuilder, Consumer<String>)
 - modified
 - void splitDocumentAndAddInputs(AlternateMixParametersBuilder, Consumer<String>, SelectionTableRowData)
 - added