EUROPEAN COMMISSION
Legal service
IT Group

European Commission

PDFMerge - general test case

Date:	13/02/2013
Version:	1.0
Authors:	████████████████
Revised by:	█████████
Approved by:	████████
Public:	
Reference Number:	

TABLE OF CONTENT

History of document

Version	Date	Comment	Modified Pages
1.0	07/02/2013	Creation of General Test Case Document for first production roll-out	All

1. INTRODUCTION

1.1. Purpose of this document

This is a generic test case to be executed before any production roll-out of the software.

1.2. Objectives

The rationale beyond this document is to provide a set of minimal testing patterns whose completion provides a reasonable insurance about reliability of main software functions.

Those patterns are extensive, meaning that their path covers the main functions of the application. Nevertheless, due to the infinite possibility of data and action sequence combinations within the scope of the application, it cannot be qualified as an exhaustive test.

It is the basic testing procedure to be associated to more specific ones when specific modules are undergoing evolutions.

Unless specified, it can be executed by people having reasonable computer-literacy without any prior knowledge of the software application or its business domain.

1.3. Dependencies

This has been done for a user having at least the following profile:

- Be a Tester

- PDFMerge test case: http://sjdapps/apps/sj-solon/?fa=documentservice.pdfmergetest

- Have access to the first test document:
  http://sjdapps/apps/solon/index.cfm?fuseaction=documentpropertysheet.view&docid=Ares(2012)36737&showdetailattr=true&repository=ares

- Have access to the second test document:
  http://sjdapps/apps/solon/index.cfm?fuseaction=documentpropertysheet.view&docid=080166e480abdd79&showdetailattr=true&repository=ares

2. OUTLINE OF THE TEST PATTERNS

The test case pattern breakdown is presented in the table hereafter. The different steps are to be executed in sequence.

Each step is itself made of unitary steps. Further to test execution, each unitary step can receive a OK (compliant result) or NOK (non-compliant result) status.

If one of the unitary steps gets a NOK status, it is required as far as possible to resume testing procedure. In case it becomes obviously impossible, the test is stopped and the project leader is to be informed without delay.

As far as possible, the test steps are to be executed with data given as example. In some situations, it will not be possible due to differences between the Test and Production environments. In this case, the step result still can be positive but it is required to keep track of this as a remark in the Result column.

ID	Libelle	Condition 1	Condition 2	Result
	Merging documents			
	File cleanup of merged documents			
	File cleanup of downloaded documents			
	Exception handling for document downloading			
	Exception handling for document merging			
	Load testing	Unmodified first test document & attachments	Make sure you have an active ECAS session before running the load test	5 merged PDF documents

3. TESTS

3.1. STEP 1 – MERGING DOCUMENTS

3.1.1. Merging documents

Description	Expected Result	OK/NOK
Use the URL provided by the project leader to access the <u>first test document propertysheet</u>.	The document property sheet appears	OK
Tick on all checkboxes marking the files to be merged	Boxes are ticked	OK
Press the PDFMerge button	The merge button is disabled for a few seconds An animation shows that an operation is on-going Eventually, the merged PDF file appears in a new window	OK
Look at the resulting PDF file	The file is made of the PDF documents ticked on the property sheet.	OK

3.2. STEP2 – FILE CLEANUP

3.2.1. Check that no obsolete file is left in the merge folder

Description	Expected Result	OK/NOK
Run a PDF Merge operation as in 3.1.1	The PDF files are merged.	OK

Wait for 6 minutes		
Run a PDF Merge operation as in 3.1.1	The PDF files are merged.	OK
Open the Test case page: http://sjdapps/apps/sj-solon/?fa=documentservice.pdfmergetest	In the merge folder, no file is more than 5 minutes old (no value above 5 should appear in the Minutes column for this folder)	OK

3.2.2. _Check that no obsolete file is left in the download folder_

Description	*Expected Result*	*OK/NOK*
Run a PDF Merge operation as in 3.1.1	The PDF files are merged.	OK
Wait for 6 minutes		OK
Run a PDF Merge operation as in 3.1.1	The PDF files are merged.	OK
Open the Test case page: http://sjdapps/apps/sj-solon/?fa=documentservice.pdfmergetest	In the CF download folder, no file is more than 5 minutes old (no value above 5 should appear in the Minutes column for this folder)	OK

3.3. STEP3 – EXCEPTION HANDLING

3.3.1. _Check that insufficient remaining disk space is properly handled by PDFMerge downloader_

Description	*Expected Result*	*OK/NOK*
On the test case PDFMerge downloader test case setup, set the PDFMerge test settings to "download"	The Minimum free disk space threshold is increased	OK
Run a PDF Merge operation as in 3.1.1 using this document. instead	PDF Downloader stops, Solon displays an error message to the user and sends a mail to the si support mailbox stating that the PDFMerge downloader encountered a low	OK

	disk space condition.	

3.3.2. *Check that insufficient remaining disk space is properly handled by PDFMerge*

Description	Expected Result	OK/NOK
On the test case PDFMerge downloader test case setup, set the PDFMerge test settings to "merge"	The Minimum free disk space threshold is increased.	OK
Run a PDF Merge operation as in 3.1.1 using this document. instead	PDFMerge displays a message to the user and sends a mail to the si support mailbox stating that it encountered a low disk space condition.	OK

3.4. STEP4 – LOAD TESTING

Description	Expected Result	OK/NOK
On the test case PDFMerge downloader test case setup, click on the link to the sample document	The sample document property sheet is displayed in a new tab or new window	OK
Click on the Run Load Test button	5 new tabs (or windows) appear, each one contains a merged PDF file.	OK

4. TEST PLAN OUTCOME

The test plan ran successfully