Vladimir Yesipov

This document contains the description, analysis, requirements, resources and materials that are going to be used for developing PDF text extractor application.

Planning document

Planning document for HND Graded project “Extracting text from PDF files”. Software Development.

Table of Contents

[Interpretational brief. 1](#_Toc476314320)

[Initial functional requirements. 1](#_Toc476314321)

[Initial non-functional requirements. 2](#_Toc476314322)

[Resources and materials used to research the PDF file text extraction. 3](#_Toc476314323)

[Other resources and materials. 5](#_Toc476314324)

[References: 5](#_Toc476314325)

# Short note.

All information about the highlighted fragment extraction below will be the feature that may be added to the project if there is enough time for doing this.

This document can be improved or changed during the next product development stages.

# Interpretational brief.

The aim of the project is to make a program helping to extract a text from the pdf report scan file or as a feature from many files. The idea of the project appeared when one of my friends said that he often looks for similar fragments of text from pdf scanned documents and takes data from these fragments only.

The text fragment that is to be extracted will be found between two symbol sequences given by the product user and saved into a resulting text file which would give a possibility to approach the information in text format and copy and paste it.

# Initial functional requirements.

To start the program extracting the text fragment user should choose the pdf file(s) they want to use for the text fragment extraction.

To find and extract the fragment of the text program the app should be able to give the user a possibility to input 2 text fragments which would mark start and finish of the text fragment that is going to be extracted. For the better user experience this could be done by selecting this piece of text in the window with pdf file text or the program could be improved in future by finding and extracting similar fragments to the highlighted.

Program expected to extract data from multiple files and for this reason the user should be given an opportunity to select several files or to select the folder with these files.

The program should organise the resulting text file in the location that would be easy to find. Ideally, the user should be able to find the location of resulting file.

Program interface should have 3 text boxes:

1. Text fragment starting point (user can type symbol sequence into this box)
2. Text fragment finishing point (user can type symbol sequence into this box)
3. Initial pdf file text (at the best-case user should be able to select the fragment he/she needs), in case the text is extracted from multiple files, the program will be looking for the same text fragment in all targeted files. Pasting the file fragment may be added as a feature in future.

Additionally, the box with the text that is going to be extracted and added into the text file could be added.

For the program interface, there should be the buttons to proceed the text extraction. In initial design I will only make search of the fragment by starting and finishing points.

But in the future program it could be improved with the extraction of the highlighted text, which will possibly add another button for the text extraction:

1. When selecting the text from the pdf file window.
2. When selecting the text by pointing the starting and finishing points.

Exit button should be created for user to quit program.

Resulting file should have heading for each text fragment extracted. This heading should say from what file the data was extracted.

# Initial non-functional requirements.

The program is expected to work on the Microsoft windows platform. And it can later be improved to work under OS X and android platform.

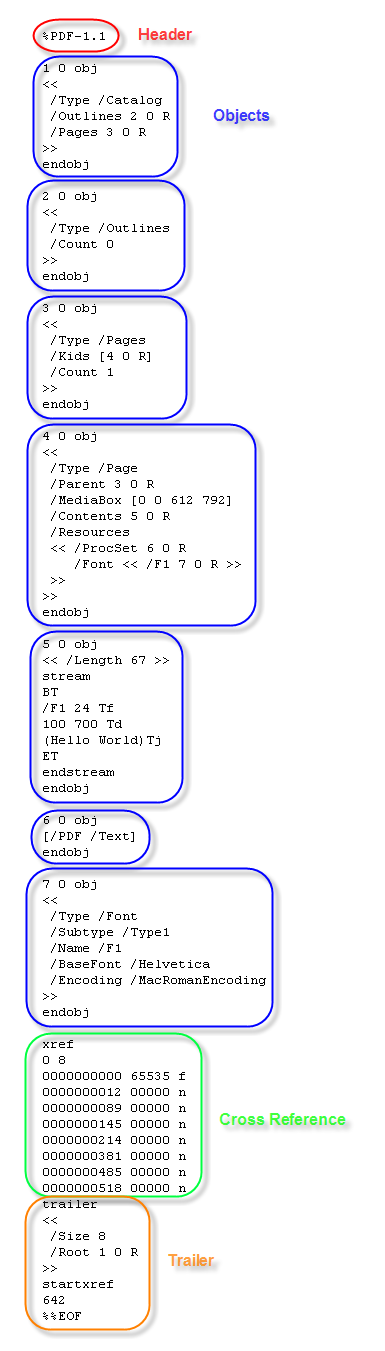
When no pdf files were selected, the message should be displayed and the program should return to the main window.

If no fragment was found or starting and finishing points are not chosen, the message should be displayed and the program should return to the main window.

Optionally for the program to work properly in case there is only starting or finishing point pointed or starting and finishing points selected together with the fragment highlighted in the initial pdf file window the program should be giving the message to the user referring to the problem.

# Resources and materials used to research the PDF file text extraction.

PDF file structure.

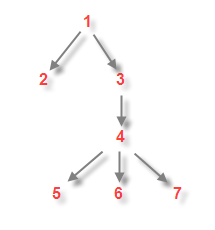
This is a simple “Hello World”-PDF viewed with a text editor:

It is composed of:

* a header
* a list of objects
* a cross reference table
* a trailer

What I describe here is the physical structure of a PDF file. The header identifies that this is a PDF file (specifying the PDF file format version), the trailer points to the cross reference table (starting at byte position 642 into the file), and the cross reference table points to each object (1 to 7) in the file (byte positions 12 through 518). The objects are ordered in the file: 1, 2, 3, 4, 5, 6 and 7.

The logical structure of a PDF file is an hierarchical structure, the root object is identified in the trailer. Object 1 is the root, object 2 and 3 are children of object 1, etc…, giving this logical structure:

[](https://didierstevens.files.wordpress.com/2008/04/pdf-tree.png)

(Didier Stevens, 2008)

“The official PDF file format specification ([published by Adobe](http://www.adobe.com/devnet/pdf/pdf_reference.html)) is large and complex. PDF files can be rich, dynamic documents, and getting to all of the interesting and useful parts of them (i.e. their content, text, metadata, etc) is a daunting task.

Further, Adobe's specification only provides normative descriptions of how PDF documents should be constructed. Experience shows that applications must often process PDF documents from multiple sources, each of which may (and do) generate PDF files that sometimes bend and often break the "official" PDF specification — similar to how web browsers are forced to support broken and malformed HTML documents as best as they can.” (Snowtide Informatics Systems, Inc., 2001-2017).

# Other resources and materials.

The application is going to be developed on Java Platform (JDK) 8u121 in NetBeans IDE 8.2. Documentation for the project is made with the help of Microsoft Office 365, the diagrams are made in Microsoft Visio, for PDF file decoding I’m going to use an open source java library.

Different libraries for PDF file data extraction written in different languages were overviewed. The list of the web resources: “Code to extract plain text from a PDF file. C/C++ Language” (Code to extract plain text from a PDF file, 2004), Code to extract plain text from a PDF file. PHP Code. (jonbeckett73 ([http://l-stat.livejournal.net/img/userinfo.gif?v=17080?v=145.4](http://ex-jonbecket43.livejournal.com/profile)[**~~ex\_jonbecket43~~**](http://ex-jonbecket43.livejournal.com/)),2005), “Navigating the Internal Structure of a PDF Document” (Thom Parker,2016), “PDF xStream. C# Language. Snowtide Library.” (Snowtide Informatics Systems, Inc., 2001-2017), Java programs. PDF to text converter (Dara Yuk, 2013), Java PDF Reader/Writer Library/Component/API (Aspire Software, 2017), Extracting Text from PDF file (Madhura Oak, 2013).

# References:

Code to extract plain text from a PDF file (2004) Available at: <https://www.codeproject.com/Articles/7056/Code-to-extract-plain-text-from-a-PDF-file> (Accessed: 31 January 2017)

jonbeckett73 ([http://l-stat.livejournal.net/img/userinfo.gif?v=17080?v=145.4](http://ex-jonbecket43.livejournal.com/profile)[**~~ex\_jonbecket43~~**](http://ex-jonbecket43.livejournal.com/)) (2005) Code to extract plain text from a PDF file. Wrote in PHP, 5 April. Available at: <http://php.livejournal.com/295413.html> (Accessed: 24 January 2017)

### Didier Stevens (2008) Quickpost: About the Physical and Logical Structure of PDF Files, 9 April. Available at: <https://blog.didierstevens.com/2008/04/09/quickpost-about-the-physical-and-logical-structure-of-pdf-files/> (Accessed: 7 February 2017)

Thom Parker (2016) Planet PDF. Navigating the Internal Structure of a PDF Document. Available at: <http://www.planetpdf.com/developer/article.asp?ContentID=navigating_the_internal_struct&page=1> (Accessed: 7 February 2017)

Snowtide Informatics Systems, Inc. (2001-2017) PDF/xStream. Getting data out of PDF documents really is this easy. Available at: <https://www.snowtide.com/#pdfts> (Accessed: 14 February 2017)

Dara Yuk (2013) Java programs. PDF to text converter, 22 June. Available at: <http://java.worldbestlearningcenter.com/2013/06/pdf-to-text-converter.html> (Accessed: 14 February 2017)

Aspire Software (2017) Java PDF Reader/Writer Library/Component/API. Available at: <https://asprise.com/product/javapdf/java_text_extract_pdf.php> (Accessed: 21 February 2017)

Madhura Oak (2013) e-Zest blogs. Extracting Text from PDF file, 30 July. Available at: <http://blog.e-zest.com/extracting-text-from-a-pdf-file/> (Accessed: 21 February 2017)