Use Apache PDFBox Library in C# .Net

Summary

Apache PDFBox is an open source Java PDF library for working with PDF documents. By using IKVM.Net, PDFBox Java Library can be ported to a .DLL file used as a reference in C#.Net applications.

Steps:

1. Download the Apache PDFBox and FontBox libraries from https://pdfbox.apache.org/download.cgi and extract the library file (e.g. pdfbox-2.0.20.jar) to a folder.

2. You many need to download Apache Common Logging library as well since it is a dependency of PDFBox.

3. Download IKVM.Net from https://www.ikvm.net/download.html (SourceForge) and extract the package to a folder.

4. Run the following IKVM command with options to convert PDFBox .jar files to a Windows .dll file.
   C: \ikvm-7.2.4630.5\bin>ikvmc -out:*pdfbox-2.0.20-integration.dll* -target:library commons-logging-1.2.jar fontbox-2.0.20.jar pdfbox-2.0.20.jar

   -out:<output file name>
   -target:library set the output file as a .dll library
   For more details of the ikvmc options, visit https://sourceforge.net/p/ikvm/wiki/Ikvmc/

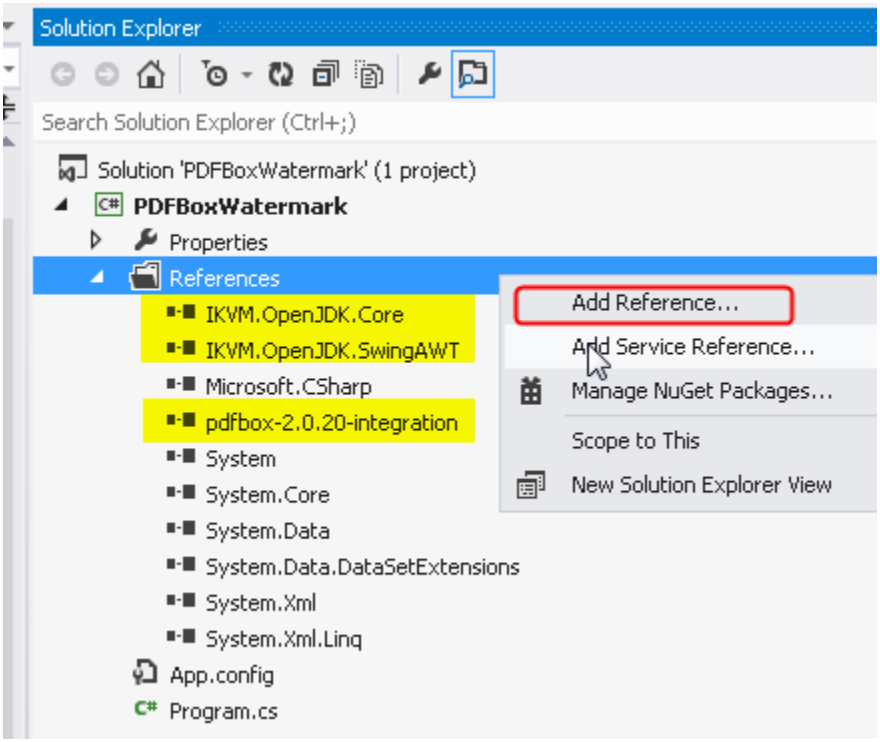
   Note: there will be some errors (e.g. ClassNotFound). You can fix some of them by adding more .jar library dependencies. The .dll file will be still generated with errors. Unless your .Net applications refer to those missing classes, you should be able use the .dll library to implement some PDF functions.

5. You may need an Open JDK environment on your machine for IKVM.

6. Open Visual Studio and create your project using PDFBox. E.g. a Windows Console application project.

7. In the project, add the required references (e.g. PDFBox.dll and IKVM libraries in the below screenshot). **NOTE**: choose the IKVM libraries from the IKVM folder (e.g. C:\ ikvm-

7.2.4630.5\bin), so the system will copy other dependent libraries to the executable folder when building the project.



8. Create an example using PDFBox in the project. You can refer to the Java examples from https://pdfbox.apache.org/2.0/examples.html since the syntax of C# is close to Java. However, you still need to make changes (e.g using the proper C# data type)
    - General steps of creating a PDF file:
        - Add namespaces (e.g. org.apache.pdfbox.pdmodel) for the API calls.

        - Initialize PDDocument
          PDDocument doc = new PDDocument();

        - Initialize PDPage and add to the PDDocument
          PDPage pg = new PDPage();
          doc.addPage(pg);

        - Initialize PDPageContentStream and start for text inputs
          PDPageContentStream cstream = new PDPageContentStream(doc, pg);
          cstream.beginText();

        - Set up a font
          PDFont font = PDType1Font.HELVETICA_BOLD;
          cstream.setFont(font, 12);

- Set the position for the content
  cstream.newLineAtOffset(10, pg1.getMediaBox().getHeight() - 20);

- Add text
  cstream.showText("Hello, world!");

- Release resources and save the PDF file
  cstream.close();
  doc.save("myDoc.pdf");
  doc.close();