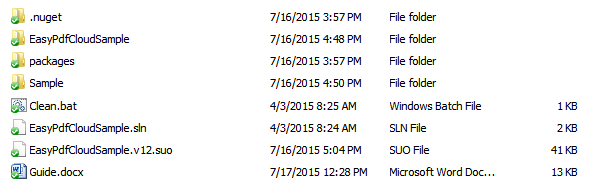
Thank you for taking the time to try out easyPDF Cloud. This tutorial is here to guide you through deploying and using the easyPDF Cloud .Net Sample.

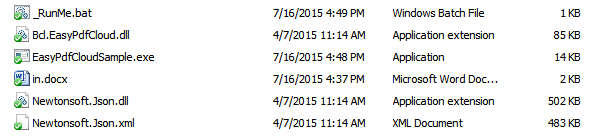
The first step is to simply choose a location on your machine, and extract the sample to that location. Once the sample is extracted, you should have a folder that looks like this.



This folder contains a Pre-Compiled sample application, and the original C# code to allow you to modify it to suit your unique needs.

Running the Sample Application:

We’ll start with the Pre-Compiled Application, so Open the “Sample” folder to find the files shown below;



These files are the bare minimum you need in order to use easyPDF Cloud, and seeing easyPDF Cloud in action is as simple as double clicking “EasyPdfCloudSample.exe”. If you do so, a Console should pop up and quickly run through the steps of a standard conversion. By default, the application will target the “in.docx” file in the same folder as the exe itself. Once it completes, you should have a new file named “in.pdf” in the folder as well

This is the simplest way to use this Sample, however it always targets in.docx, and nothing else, and this file name is hard coded into the Sample itself. What if you want to use it to convert a different file? That is what the \_RunMe.bat file is for.

Because this Sample is a simple Console Application, it can be run with Input Variables passed at launch. This can be done from the Command Prompt, or via a bat file like the one included. Bat files can be opened by a simple text editor like Notepad, and if you open \_RunMe.bat, you should see the following;



The first line is what calls the sample application, and does so simply by including the name of the application. The following three bits of text in quotation marks are the input variables. First is the Workflow ID. The one seen here is the default Workflow to convert documents to PDF. Next, is the Output Directory that the Output File will be saved into. Lastly, is the Input File itself.

The “pause” command is simply there to insure the Console does not close once the conversion is completed, allowing you to review the Report regarding the conversion.

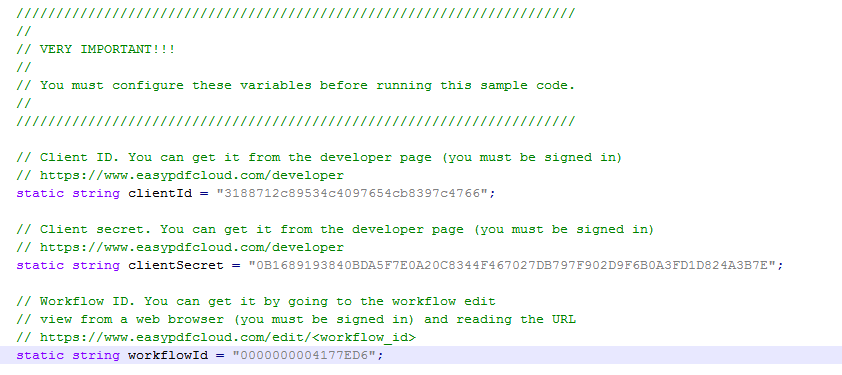
Using this bat file, or a similar one, you can target any file or directory on your machine. You can call the Application as many times as you want simply by adding more lines like the first.

Modifying Sample the Application:

And that is how you can use the easyPDF Cloud Sample, right out of the box. However what if you want to modify it? If you wish to modify it, all the source code required to do so is included with this sample. If you have Visual Studio, all you need to do is double click the “EasyPdfCloudSample.sln” file located in the main Sample Folder and Visual Studio will automatically open the Project, allowing you to edit any variables as you see fit and recompile.

Certain variables are Hard-Coded into the Application for security reasons, in particular the Client ID and Client Secret. By default, the values provided are for the Demo Account of easyPDF Cloud. If you design your own Workflows you will need to change these to point to the ID and Secret of your personal account.

Fortunately, the Variables are located at the very top of the Source Code.



To change the application to target a different User Account, all you need to do is change these two variables. Likewise, here you can modify the default Workflow ID.

Once you have finished making your changes, compile the new Console Application, and get the new Executable from the Bin Folder.

And that’s how you use the easyPDF Cloud .Net Sample.