# WestFax SDK Demo

## WestFax API

The WestFax API is comprised of a large set of methods that are exposed using a RESTful interface. Data can be interchanged in a number of formats including JSON and XML. Additionally, WestFax offers SOAP and RPC bindings that are described elsewhere. For the purposes of this document and the SDK, the REST interface with JSON encoding is used. WestFax hosts the API servers in its dedicated and secure datacenters, providing robust and fault tolerant access to our Cloud Fax API.

## Purpose

The purpose of this document and the associated visual studio project and code files is to demonstrate the use and common work flow for an API implementation. The Included SDK (with source code) can be used as is, or modified to suit specific needs. In conjunction with this SDK demo, WestFax has published a Postman (https://www.getpostman.com/) collection that demonstrates the usage of the API using the Postman tool.

The SDK demo includes a console application that exercises the API methods in the SDK, and shows how they might be used, from authentication to discovery of account and product information to sending and receiving faxes.

## About the Demo

The WestFax SDK Demo is presented as a Visual Studio Solution that contains 3 projects. One project is called WF.SDK.Common, and contains a set of classes that provide basic functions that will be used throughout the SDK. Another project is called WF.SDK.Fax. This project is the core of the SDK. It contains the FaxInterface class that has the fax methods, and it also contains all of the objects that are passed to or returned from the API methods. (These can be found in the Models folder).

There are 2 levels of API and 2 levels of models that are used. The first is the "Raw" interface (FaxInterfaceRaw Class). This represents the low level API request and responses. A quick inspection of the methods here will reveal that the parameters of the methods are formed into a name-value Form POST collection, and sent to the API in an HTTP POST operation. The response from the API is received as a string.

The second level of API, is a higher level interface that accepts and emits higher level objects within the FaxInterface Class. For each method in the FaxInterface class, a corresponding method in the Raw interface is called. The FaxInterface methods serialize and deserialize the passed parameters and responses to and from the Raw interface.

The models folder contains 2 sets of models. One is Internal, and the other is External. The FaxInterface consumes and emits the External models only. The Raw interface uses the Internal models. The internal models are coded to mirror the JSON data that is passed to and from the API. The External models lend strongly typed Enums and other (inheritance) features that are awkward for JSON serialization/deserialization. A "translation layer" in the form of extension methods and constructors allows for the conversion of these types. As a user of the SDK, knowledge of this translation layer is not necessary.

## *Using the Demo*

The Demo project is a console application that shows the use of the SDK and its methods. There are several methods within the WorkFlowDemo class that demonstrate various aspects of the SDK from authentication, to Account, Product, User and Contact discovery to Sending and Receiving Faxes.

## *References*

It may be necessary to re-reference some dlls that are contained in the "binaries" folder. The 2 binaries are 1) a common JSON library (newtonsoft) and 2) the NUnit framework dll. This should be evident when trying to build the project.

## *Configuration*

Inside of the demo project folder there is an App.Config file. The contents of this are self explanatory. Alter this file with your account User and Password and ProductId. You will receive this from WestFax when requesting API credentials. Additionally, you can configure an email address and a Test Fax number that will be used within the Demo. (Note that your account may include an Inbound Number, you can use this in the Demo. The WestFax system allows for an outbound fax to loop to the same inbound number. This is convenient for testing purposes.)

## *Running the Demo*

After configuring the Demo application, you should be able to start the Demo application in Visual Studio. It will run and present you with prompts as needed ("y" or "n" answers are the only responses).

**Note:** A portion of the Demo retrieves an Inbound Fax. If you do not have an inbound Fax number on your product, then skip this method in the DemoWorkflow class. It will not work.

**Note:** It is possible that you have not received any inbound faxes on you account. If you have not, then the inbound fax parts of the demo will simply exit. Send a fax to your account and then run the Demo. Again -- you can use the Demo itself to initiate this Fax. You may have to wait a minute or 2 for the WestFax system to actually deliver the fax to your account.

We recommend that you use Visual Studio and break into the Demo code as it is running. There are many helpful comments in the code that describe what is happening. You can use the debugger to explore the API and the classes that are returned more fully. We hope that the SDK along with the starter code and comments will get you underway with your integration quickly.

## *Something Missing?*

Occasionally you may find that a particular method or piece of data specific to your application is missing from the SDK. If this occurs, please contact WestFax, and ask about other interface options. This SDK ***does not*** include all methods that are available in the API. We can and will add methods and enhanced objects and data to the SDK as needed. We believe that the current set of methods and object encompasses 90-95% of our customers' needs.

## *Need Help?*

Our responsive support staff is always ready to assist with problems. Please let us know if you have questions about this document or the Demo. Please email support@westfax.com with any questions.