# Modes:

Regular License plugins are run in 3 modes:

1. Normal
2. Sanctions
3. Expirables

If a site has license expiration dates, the plugin must support Expirables mode.

If a site has sanctions(board actions, disciplinary info, etc.), the plugin must support Sanctions.

Provider with sanctions must be RED FLAGGED even if the plugin is run in Normal or Expirables mode.

1. In Normal mode, provider details are always returned to the end-user. Providers with sanctions are RED FLAGGED.
2. In Sanctions mode, providers with sanctions are RED FLAGGED and their details are returned to the end-users. Providers without sanctions are not returned.
3. In Expirables mode, if the license expiration date on the site does NOT match the date sent by the end-user, the provider details are returned. Otherwise, provider details are flagged to not return in Verity (indicated by the “SendBack” flag). Providers with sanctions are RED FLAGGED.

# PlugIns:

When creating plugins, make sure you have the Initialize() method. (See ALBMEPlugIn solution)

If sanctions are found on the provider’s page, RED FLAG the provider by calling setSanction(true). Otherwise call setSanction(false).

To run a plugin in expirables mode, you must set the **action**=”E” in your test xml.

Call ProcessResults(output, message) at the end of the Fetch() when you are done scraping. If you add a “message” the plugin will always return a message instead of the output. The “output” is your parsed html with td pairs.

Setting up the Web Scraper:  
Your WebScraper will need to have references linked to your local .dll files.  
All third party libraries used by the WebScraper can be found here in TFS:  
TFS > Echo PSV > Third Party Libraries

The following references should be re-added:

PdfMetaMorphosis.dll

RestSharp.dll

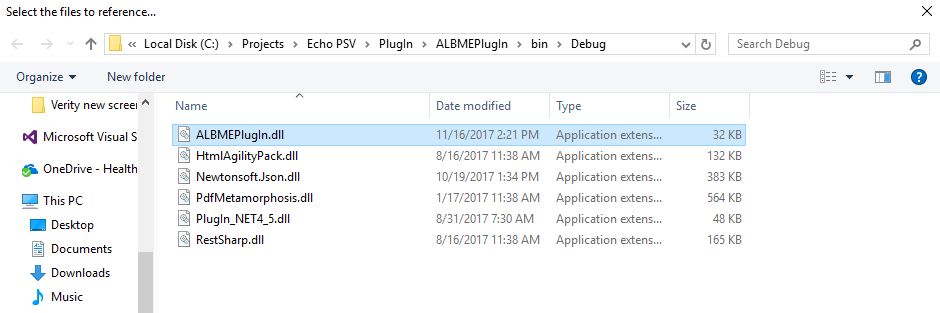
Newtonsoft.Json.dll

HtmlAgilityPack.dll

ABCpdf.dll

You will need an installation to ABCPdf by running abcpdf.net64.exe. This can also be found in the Third Party Libraries of TFS.

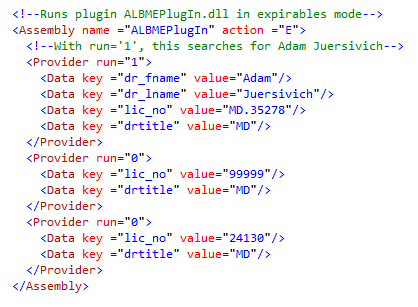
After your plugin has successfully built and met all the requirements of a standard plugin, you can now add the plugin as a reference in the WebScraper.   
  
Right click References > Add Reference > Browse to your plugIn project’s bin/Debug folder > Select the .dll of the plugin



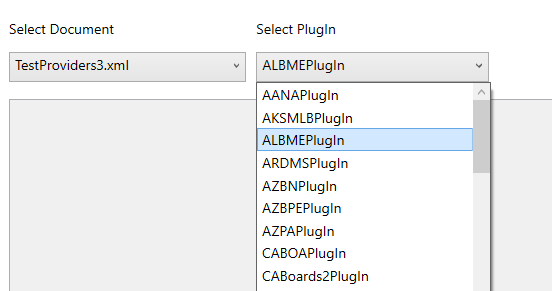
To set up your test providers, all test providers are located in the WebScraper solution’s TestDocuments folder. You can add new documents, but make sure in the properties you set the Build Action to “Content” and Copy always to output directory.

In each .xml test file, when you edit test providers, the layout must be in a format that the WebScraper can read.  
  
The **Assembly** tag encloses each different plugin with the **name** attribute indicating the plugin.  
Each assembly has child nodes **Provider**. Set **run**=”1” to run that provider. The **action** attribute indicates the mode (“” for normal/ “E” for expirables/ “S” for sanctions). See above “Modes” for more details.

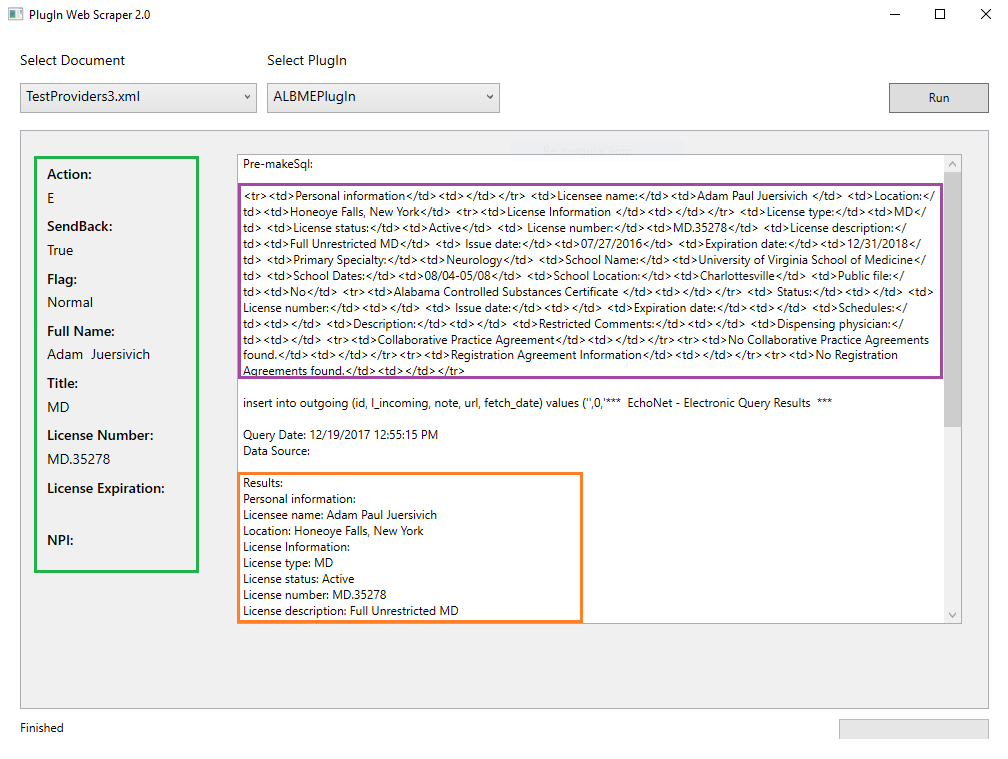
Each provider has child nodes **data** that indicates the properties of the datarow that is passed into the plugin’s Fetch() method. You can add data rows but only if they match the pre-defined columns in XmlLoader.WebCrawler. If that data tag is not present, the plugin will see that field as an empty string.



If you add new TestDocuments and new Assembly tags in your test document, the WebScraper will allow you to select them



You are now ready to run the plugin. Click the “Run” button and you will see results like this:



The green box shows data that is processed by the plugin. The purple box shows the result that is inserted into the “outgoing” sql table as results. The orange box is a preview of what clients will see after Verity processes data from the “outgoing” table.

Often you will need to step into the code of your plugin. The plugin assembly initializes in WebCrawler.DoPSV() and you can step into the plugin at instance.Fetch(). Put a breakpoint there to step in during runtime.

To see the images (or PDFs) returned by your plugin, you must set MainViewModel.PlugInSupportsImages = “2” and your plugin must have set the pdf.html. Your images are located in PlugInWebScraper\PlugInWebScraper\bin\Debug\\_SamplePDFs

# Tips:

If you run into issues, try these tips:

1. Try removing the referenced assembly of your plugin and re-adding them (If it’s your first time using that assembly it may not be properly linked)
2. Try cleaning your plugin project & rebuilding it (in debug mode)
3. Make sure you don’t have extra Nodes or attributes in your .xml file
4. Make sure the provider you are testing has attribute run=”1” in the xml file