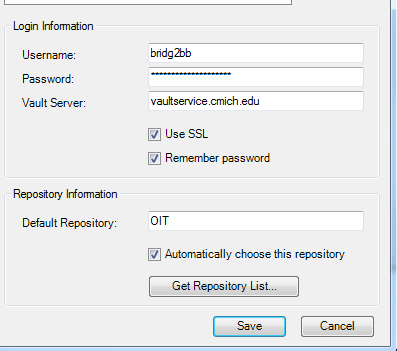
# Assignment 10 Intro to Source Control/Core API

In this assignment you will be learning about the source control used here and how to compile/use the main CoreAPI. For source control Vault Server Standard is used, which integrates into visual studio. If no other student users your machine, you will need to download the client from <https://vaultservice.cmich.edu/vaultservice/downloads/VaultClient.msi>. If another student uses the same computer as you the client should already be installed.

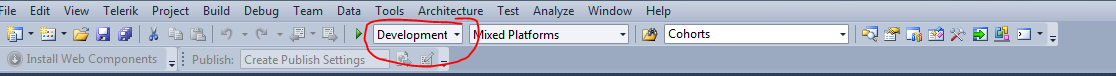
Our CoreAPI is a set of assemblies that standardize some of the common functionality we use across many different applications. In order to use the assemblies properly, you will need to setup your machine to our standards.

### Setup on your Computer

1. Vault Standard (<http://www.sourcegear.com/vault/>)
   1. The client can be found at <https://vaultservice.cmich.edu/vaultservice/downloads/VaultClient.msi>
   2. Default profile settings (uses Active Directory credentials for sign-in) 
   3. We require our working folder is setup in a consist manner. The location of these files is usually in the folder C:\Projects\, but can be located on another hard drive if the need calls for it
   4. The main repository (OIT) is Located at Projects\Solutions. To set this right click on repository root ($) and select “Set Working Folder…” You will need to set your working folder to C:\Projects\Solutions. Unless you used a different option in step c. then replace C:\Projects\ with what you used.
   5. The folders References, Resources, and utilities should point to C:\Projects\Name. Once again if you used something different in step C. then replace C:\Projects\ with what you used.   
       e.g. C:\Projects\References, C:\Projects\Resources, and C:\Projects\Utilities
   6. Certain projects need to be downloaded for any development. Each of the following will need to be downloaded (by right clicking and selecting Get Latest)
      1. References
      2. CMich API
      3. Resources
      4. Utilities
2. Operating System
   1. Environment Variables (<http://www.itechtalk.com/thread3595.html>)
      1. DEV\_ROOT – Should point to your vault working folder that you setup above. (by default this is C:\Projects) This should be a user variable

### Core API

In the folder C:\Projects\Solutions\CMich API\ you will see a few different solution (.sln) files. Open CMich.sln. This is the base CoreAPI, it contains multiple projects that all have a different purpose. It also leverages build modes. Rule of thumb, you should always be compiling in “Development” mode unless otherwise told to. Also, you are not to change code inside the core API without permission. Go ahead and compile the core API in development mode.



If you setup everything in the previous portion correctly you should have no errors. If you did have errors, please contact a mentor to take a look to make sure everything is setup correctly.

### Additions to Your Application

There are two ways of including the core API into your solution, for now we are just going to focus on referencing the assemblies. In order to do this, you will need to right click on references in your .Web project and select “Add Reference.” Under the “Browse” tab navigate to C:\Projects\References\CMichAPI (if this folder does not exist you did not complete the above section correctly). Inside this folder you should see a bunch of assemblies (.dll), select CMich, CMich.Data, CMich.ConnectionSettings, CMich.Logging, CMich.Sap, CMich.Security, CMich.Settings, CMich.Web, and CMich.WebServices, finally click “Ok”. You have now added the entire core API into your project and can use it.

Inside of CMich.Web we have a folder called Base, which includes some classes. These classes inherit from a specific object inside of the .NET framework and then add functionality to that object. In this project, you will be using the object CMich.Web.Page. Your pages will need to inherit from this object, currently they inherit from System.Web.UI.Page, which is what our object inherits from in the end. The code behind class definition for your Default.aspx page would look like this: 

This gives you the ability to call into our session object that contains information on the current user. In this project you will need to check if a user is authenticated in code behind, if they are use the CurrentSession object to prepopulate FirstName, LastName, and GlobalId.

e.g. this.txtFirstName.Text = this.CurrentSession.FirstName;

### Documentation

After that, please take a look through files inside the CoreAPI and document everything that makes sense to you, also note anything that you don’t understand. There is a lot of code here, so please break your documentation into sections based on the projects inside of the CoreAPI. For example a section on CMich, CMich.Data, CMich.ConnectionSettings, etc…

You are not expected to know everything and a lot of it will not make any sense, but make the best effort possible.