System Installation

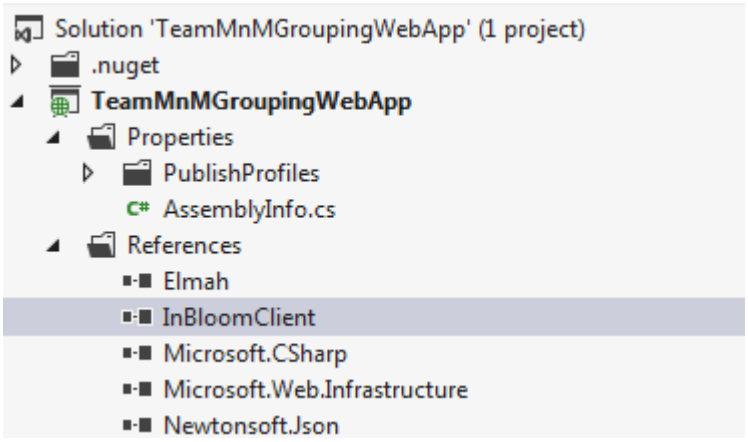
The Student Group Tool is a web application developed using the .NET 4.5 framework on Visual Studio 2012. It can be installed on a stand-alone IIS server or on a Windows Azure Website instance.

Requirements:

- .NET 4.5 runtime
- IIS version 7+

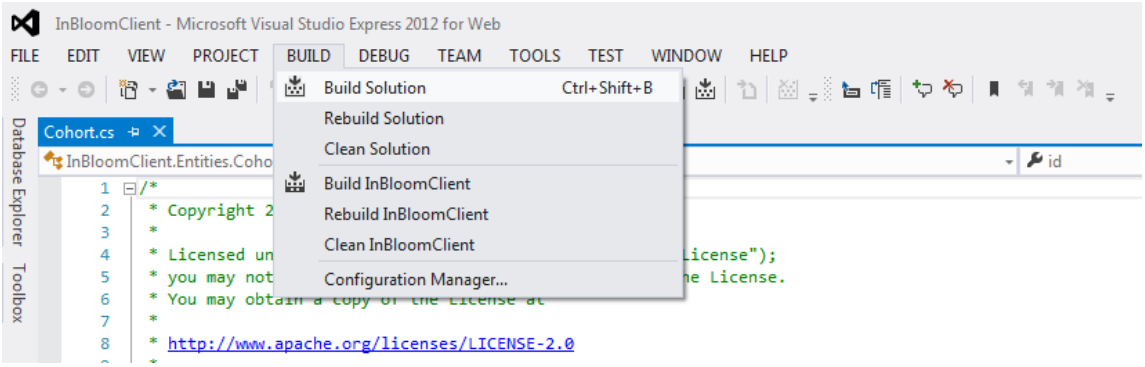
In order to build the Student Grouping Tool application, you must make sure that the **TeamMnMGroupingWebApp** contains the reference to the **InBloomClient** library. To do this follow these steps:

1) Open the **TeamMnmGroupingWebApp** in Visual Studio 2012
2) Under the **Solution Explorer** expand **TeamMnmGroupingWebApp**
3) Expand the **References** folder
4) Make sure the **InBloomClient** reference is there (as shown in the figure below). If it is not, there will be an exclamation mark next to the name.

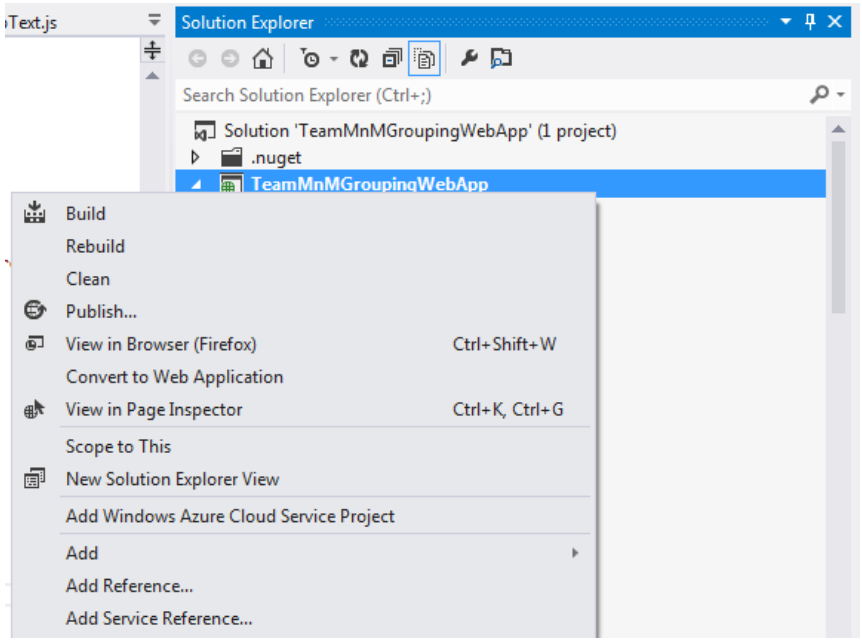


If there is an exclamation mark next to **InBloomClient**, follow these steps to build the application:

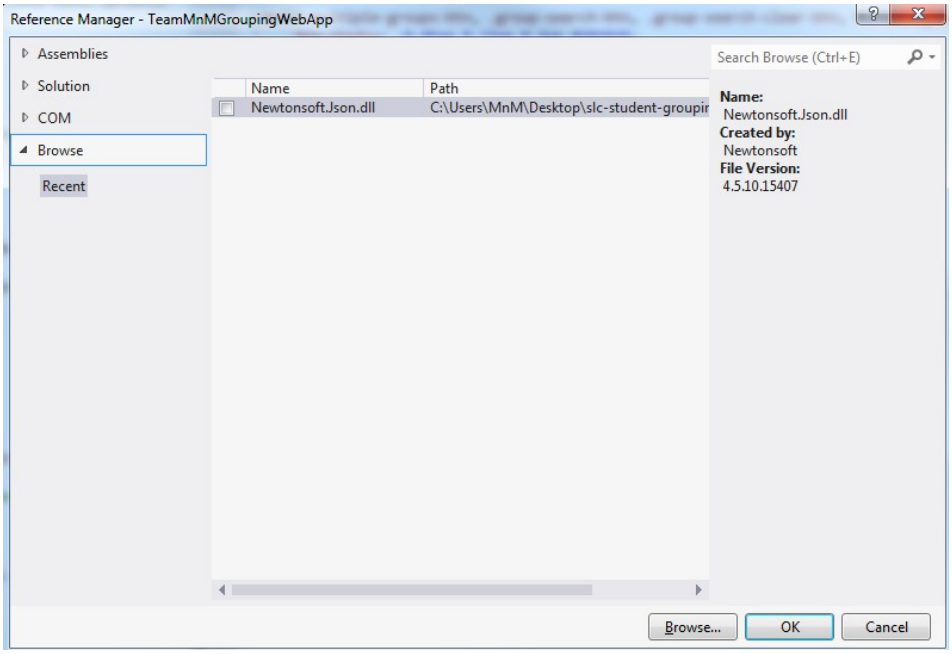
1) Open the **InBloomClient** project in Visual Studio 2012
2) Go to the **Build** menu
3) Click on **Build Solution.** This will generate the required assemblies.



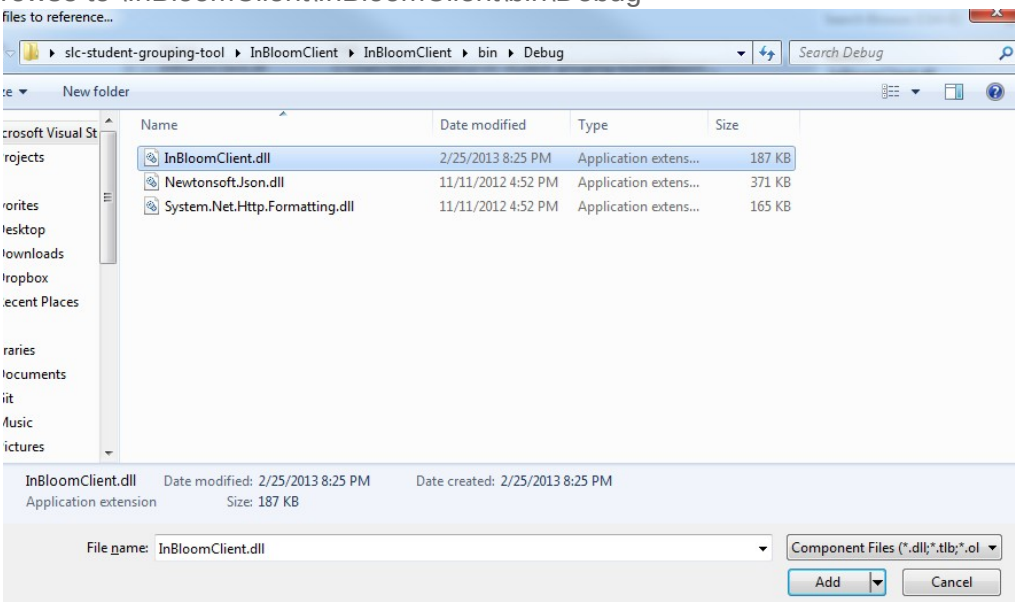
4) Open the **TeamMnMGroupingWebApp** project in Visual Studio 2012
5) Under **Solution Explorer** right-click on **TeamMnMGroupingWebApp**



6) Click on **Add Reference.** This should pop up a window.



7) Click on the **Browse** tab on the left
8) Click on the **Browse…** button. This should pop up another window
9) Browse to \InBloomClient\InBloomClient\bin\Debug



10) Select the **InBloomClient.dll** file
11) Go to the **Build Menu**
12) Click on **Build Solution**
13) The Student Grouping Tool application should be built and ready to deploy

The **TeamMnMGroupingWebApp** project contains a web.config file that defines the following credentials and authentication endpoints to authenticate against the inBloom APIs:

- InBloomClientId – unique id assigned by inBloom for accessing the APIs
- InBloomSharedSecret – secret code assigned by inBloom for accessing the APIs. To be used with the clientId
- InBloomRedirectUrl – the url to redirect to upon successful login
- InBloomSandboxLogin – the oauth authorization URL
- InBloomOAuthUrl – the oauth token URL

```
<appSettings>
  <add key="webpages:Version" value="2.0.0.0" />
  <add key="webpages:Enabled" value="false" />
  <add key="PreserveLoginUrl" value="true" />
  <add key="ClientValidationEnabled" value="true" />
  <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  <add key="InBloomClientId" value="Pw2MV5HnIl" />
  <add key="InBloomSharedSecret" value="v246RgngbAnLBZ2lEvaxBffMdlLRilU3ZOku79h7edvIQ069" />
  <add key="InBloomRedirectUrl" value="http://localhost:49560/home" />
  <add key="InBloomSandboxLogin" value="https://api.sandbox.inbloom.org/api/oauth/authorize?c
  <add key="InBloomOAuthUrl" value="https://api.sandbox.inbloom.org/api/oauth/token?client_id
  <!-- Hard-coded ed org id -->
  <add key="CurrentEdgOrgId" value ="82c1927f5114727c7d2c7494b46b355d5da5624b_id"/>
  <add key="FtpServerUrl" value="ftp://scratch.osintegrators.com" />
  <add key="FtpServerUser" value="slcftpuser" />
  <add key="FtpServerPass" value="slcFTPUs3r" />
  <add key="FtpUseSSL" value="true"/>
  <add key="aspnet:MaxJsonDeserializerMembers" value="16777216" />
</appSettings>
```

Figure - Web Config Details

These credentials (shown in Figure 1) must be configured for the Sandbox account that will be used for the web application.

Publishing to Windows Azure

1. Sign up for a Windows Azure account at http://www.windowsazure.com/en-us/
2. Log into the Windows Azure portal with your account
3. Click on the 'Websites' tab on the left
4. Create a new website
5. Once created, download the Publish profile for that website
6. In Visual Studio, right click on the **TeamMnMGroupingWebApp** project
7. Click on 'Publish' and the Publishing wizard will appear
8. Click on 'Import' and select the publish profile you just downloaded
9. Click on the 'Publish' button at the bottom of the window
10. The Student Grouping Tool app will be published to your Windows Azure website instance

Configuration

The inBloom Client project contains an **app.config** file that defines the following URLs for accessing the inBloom API endpoints:

- InBloomSandboxUrl – root inBloom sandbox URL
- InBloomApiSandboxUrl – inBloom API Sandbox URL used to make the API calls agains
- InBloomApiSandboxSessionUrl – URL used to retrieve the user info, such as the user id, whether the user is authenticated, and the user's roles.

```
<applicationSettings>
    <InBloomClient.Properties.Settings>
        <setting name="InBloomApiSandboxUrl" serializeAs="String">
            <value>https://api.sandbox.inbloom.org/api/rest/v1/</value>
        </setting>
        <setting name="InBloomApiSandboxSessionUrl" serializeAs="String">
            <value>https://api.sandbox.inbloom.org/api/rest/system/session/</value>
        </setting>
        <setting name="InBloomSandboxUrl" serializeAs="String">
            <value>https://api.sandbox.inbloom.org/</value>
        </setting>
    </InBloomClient.Properties.Settings>
</applicationSettings>
```

Figure - inBloom Endpoint Configs

Educational Organization ID

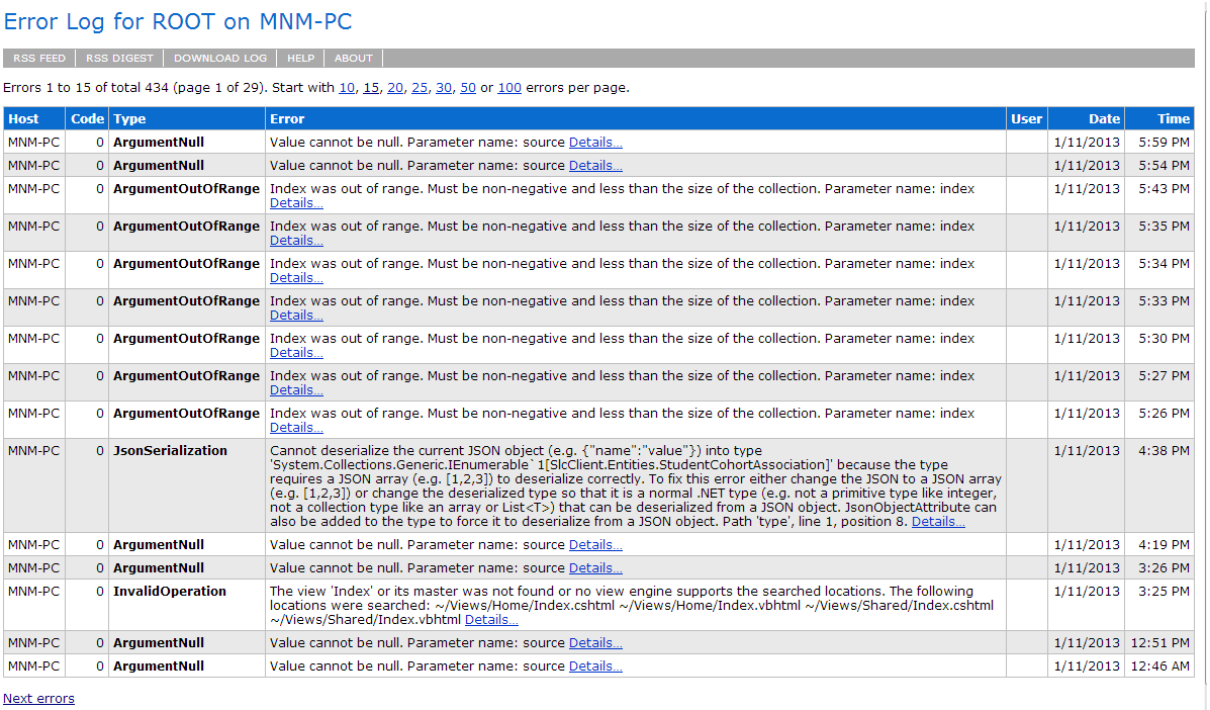
The Educational Organization ID (**EdOrgId**) is required to create cohorts. Because we are using the sample data set, there is no EdOrgId associated with the existing users, therefore we hard-code this ID for the user **Amy Kopel** (akopel) in the **web.config** file. However, the code for retrieving the **EdOrgId** associated with the current authenticated user has already been implemented. If the code cannot find the associated **EdOrgId**, then it will fall back to using the hard-coded one specified in the **web.config** file.

```
<!-- Hard-coded ed org id -->
<add key="CurrentEdgOrgId" value ="82c1927f5114727c7d2c7494b46b355d5da5624b_id"/>
```

Figure - Hardcoded Educational Org Id in Web.config

Error Logging

All exceptions thrown by the application are logged using the **elmah** library. The error logs can be accessed through the path root/elmah.axd. For security purposes remote access has been disabled, meaning that the logs can only be accessed locally (i.e. localhost) or by manually downloading the log files.



Error Log for ROOT on MNM-PC

RSS FEED | RSS DIGEST | DOWNLOAD LOG | HELP | ABOUT

Errors 1 to 15 of total 434 (page 1 of 29). Start with 10, 15, 20, 25, 30, 50 or 100 errors per page.

Host	Code	Type	Error	User	Date	Time
MNM-PC	0	**ArgumentNull**	Value cannot be null. Parameter name: source Details...		1/11/2013	5:59 PM
MNM-PC	0	**ArgumentNull**	Value cannot be null. Parameter name: source Details...		1/11/2013	5:54 PM
MNM-PC	0	**ArgumentOutOfRange**	Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index Details...		1/11/2013	5:43 PM
MNM-PC	0	**ArgumentOutOfRange**	Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index Details...		1/11/2013	5:35 PM
MNM-PC	0	**ArgumentOutOfRange**	Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index Details...		1/11/2013	5:34 PM
MNM-PC	0	**ArgumentOutOfRange**	Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index Details...		1/11/2013	5:33 PM
MNM-PC	0	**ArgumentOutOfRange**	Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index Details...		1/11/2013	5:30 PM
MNM-PC	0	**ArgumentOutOfRange**	Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index Details...		1/11/2013	5:27 PM
MNM-PC	0	**ArgumentOutOfRange**	Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index Details...		1/11/2013	5:26 PM
MNM-PC	0	**JsonSerialization**	Cannot deserialize the current JSON object (e.g. {"name":"value"}) into type 'System.Collections.Generic.IEnumerable`1[SlcClient.Entities.StudentCohortAssociation]' because the type requires a JSON array (e.g. [1,2,3]) to deserialize correctly. To fix this error either change the JSON to a JSON array (e.g. [1,2,3]) or change the deserialized type so that it is a normal .NET type (e.g. not a primitive type like integer, not a collection type like an array or List<T>) that can be deserialized from a JSON object. JsonObjectAttribute can also be added to the type to force it to deserialize from a JSON object. Path 'type', line 1, position 8. Details...		1/11/2013	4:38 PM
MNM-PC	0	**ArgumentNull**	Value cannot be null. Parameter name: source Details...		1/11/2013	4:19 PM
MNM-PC	0	**ArgumentNull**	Value cannot be null. Parameter name: source Details...		1/11/2013	3:26 PM
MNM-PC	0	**InvalidOperation**	The view 'Index' or its master was not found or no view engine supports the searched locations. The following locations were searched: ~/Views/Home/Index.cshtml ~/Views/Home/Index.vbhtml ~/Views/Shared/Index.cshtml ~/Views/Shared/Index.vbhtml Details...		1/11/2013	3:25 PM
MNM-PC	0	**ArgumentNull**	Value cannot be null. Parameter name: source Details...		1/11/2013	12:51 PM
MNM-PC	0	**ArgumentNull**	Value cannot be null. Parameter name: source Details...		1/11/2013	12:46 AM

Next errors

Figure - Elmah Error Console

The **elmah** configuration can be found in the **web.config** file. You can specify which directory to save the log files to, as well as the path to access the **elmah** error logs screen (shown in Figure 4). For more information on **elmah** please visit http://code.google.com/p/elmah/wiki/DotNetSlackersArticle.

If the web application is hosted on the cloud, such as Windows Azure, then instead of accessing the logs locally, the logs can be downloaded and consolidated into reports using the **elmah** Log Download tool at http://code.google.com/p/elmah/wiki/ErrorLogDownloadApplications.