



Select Survey .NET

Developers Manual



SelectSurvey.NET Developers Manual

(Last updated: 5/6/2016)

Table of Contents:

SelectSurvey.NET Developers Manual	1
Overview	2
Before Starting - Is your software up to date?	2
General Design	2
Debugging Source Code with Visual Studio	3
Database Model	8
Look and Feel	9
Common Modifications	10
Modifying the Application Security	10
Running the Software under HTTPS	11
Adding a New Item Type	11
Modifying Buttons and Images	11
Increasing the Length of the Answer Fields	12
Translating into a Foreign Language	12
Integrating with an Existing Database	13
Ajax Control Toolkit	13
Visual Studio Structure	13



Overview

The SelectSurvey.NET developer's manual provides documentation about the design of the software, as well as answers many commonly asked questions about modifying or enhancing the software. This document is an ongoing work that reflects the needs of our customers. If the information you are looking for is not in this document, please contact us at tech@classapps.com, and we will gladly answer your questions. Additionally, with your input, we can continue to modify this document and add to it so that it eventually addresses the needs of our entire developer community.

Before Starting - Is your software up to date?

Every month a new feature/fix release is published into the client portal of all clients that are current on support. Clients have access to download and upgrade at their own discretion at any time 24x7. ReleaseNotes.txt is included with every release, indicating which files were changed, and what new features or fixes were added. Release notifications are published on facebook and twitter as well as the client portal. 99% of the time the functionality clients are looking for has already been added and they just don't know where to find it. There are training documents with screenshots and videos linked on the classapps.com website under support > documentation (Training section). Also, our public sandbox allows current clients to test that their issues have been resolved or feature requests have been added.

The application has been designed to allow users to specify look and feel, logos, style sheets, through the interface without any programming involved. Also the application has the ability to query 3rd party databases dynamically with the dynamic token features. You can add database tokens through the interface as needed. Automation can be achieved also by using the API along with the numerous features of copying surveys/responses/ and data retrieval.

General Design

Version v4.146.005 (released 11/8/2015) or higher require .NET 4.5 framework selected in App Pool in IIS

Version v4.083.008 – v4.146.004 (Released 4/28/2013) require .NET 4.0 framework selected in App Pool in IIS

Version v4.026.000 – v4.083.007: Visual Studio 2010 and require .NET 3.5 framework selected in App Pool in IIS

Version's v2.0- v4.026.000: Visual Studio 2005 and .NET 2.0 framework selected in App Pool in IIS

Versions v1.0-v2.0: Visual Studio 2003 and .NET 1.1 framework selected in App Pool in IIS

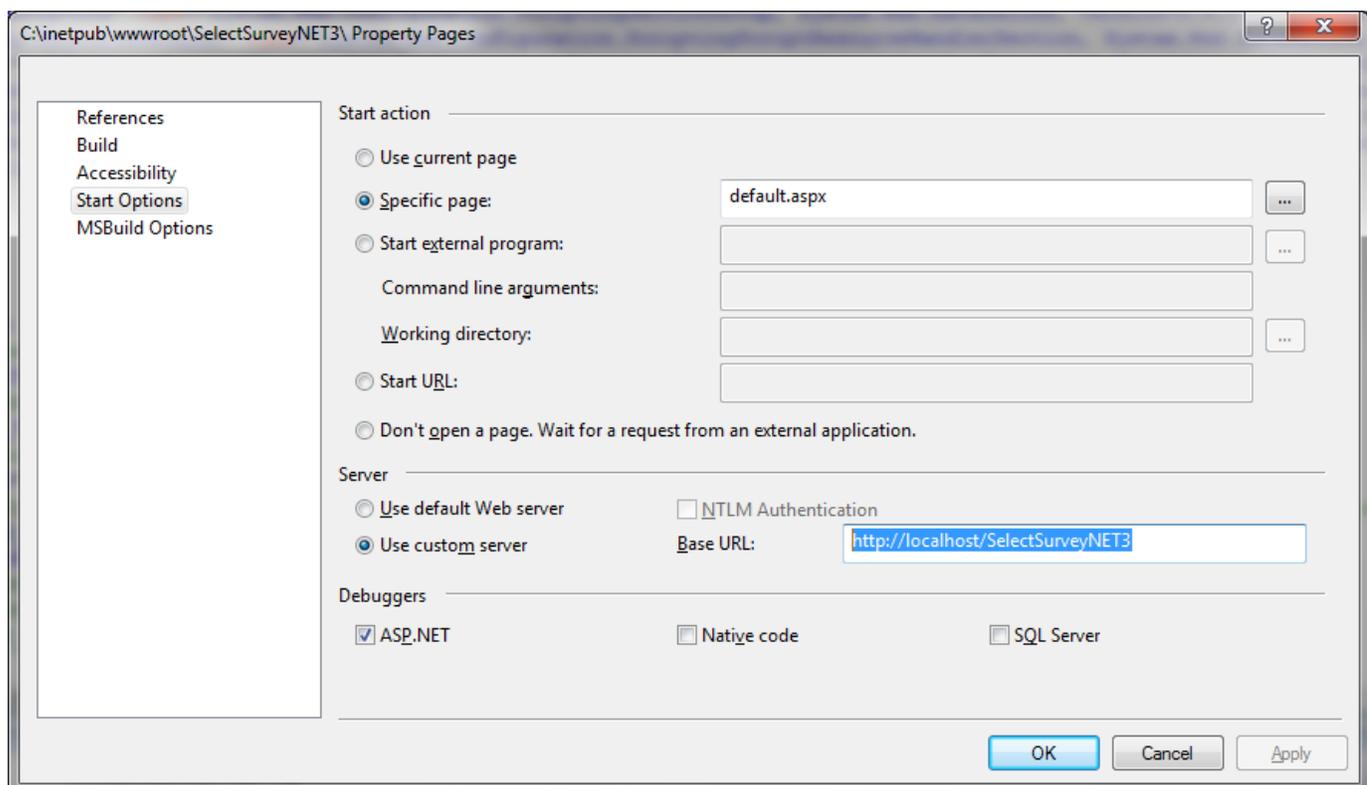
SelectSurvey.NET is written in the C# language for the Microsoft.NET framework.

Debugging Source Code with Visual Studio

Microsoft Visual Studio or similar .net programming tool is required to re-compile the application if you make changes to the code behind (.cs) or class files.

There are two folders in the software zip folder “Files to Copy to IIS” is the precompiled, ready to deploy published web site. “Source Code” is the source files which you can edit and re-publish per instructions below.

PLEASE READ: The application is SERIALIZED to work with web farms and web clusters. If you modify the application be sure to maintain the serialization by adding the [Serializable] tag to each new class file so that session can be maintained between servers. Due to the serialization, you cannot debug in a temporary IIS port. You must edit the Visual Studio options to specify a real IIS folder to debug the application, or you will get serialization errors. This is by design of Microsoft. See screenshot below for how to configure Visual Studio:



The above screenshot shows how to configure the “Custom Server” so that you can debug it without getting serialization errors.

Requirements for Visual Studio 2010 to support HTML5 features (from 2013 versions):

2016 note: Most of the below information can be ignored, because HTML5 has now been out for some time and now Visual Studio 2015 is available.

- Visual Studio 2010 SP1 (bit.ly/nQzslid) is required. In addition to adding support for new HTML5 input types and attributes, the service pack also provides some updates that enable you to use the HTML5 input types on the TextBox server control. Without this update, you'd see compile-time errors when using the new types.
- Microsoft .NET Framework 4 Reliability Update 1 (bit.ly/qOG7Ni) is required. This update is designed to fix a handful of problems related to using the new HTML5 input types with ASP.NET Web Forms.
- Preliminary support is added for HTML5 to the HTML editor so that IntelliSense and validation for HTML elements and attributes can be obtained. A complete version that brings a fuller HTML5 experience will be provided in future versions of Visual Studio. To get HTML5 support, go to Tools | Options, select Text Editor | HTML | Validation, then select HTML5 from the dropdown. Or you can also select HTML5 from the HTML source editing toolbar (see Figure 13). To enable the toolbar while editing an HTML or Razor file, select View | Toolbar | HTML Source Editing.

To open the Source Code in Visual Studio do the following:

(Versions after 4.047.000):

1. Create new solution file in visual studio (2010 – 2015) with SRC folder referenced.
2. Make sure visual studio is set to run from custom server directly from C:\inetpub\wwwroot folder or real IIS folder, you will get errors debugging from a temporary port, due to the serialization. See developer manual for screenshots (documentation folder)
3. click "ADD existing project" and select "ClassApps.Common"
4. Click "add existing project" and select "UnitTest" (if this project is in the source code folder – in some versions it may not be present)
5. When you build, it should build all three projects together, ClassApps.Common is referenced in the bin folder of SelectSurvey.NET. All files will display in Visual Studio, in the explorer panel on the right side RIGHT CLICK the top outer solution icon and click "BUILD". The application should build with no errors. Right click the top "src" folder and click "PUBLISH" and browse to the folder "C:\inetpub\wwwroot>SelectSurveyNETPrecompiled" this is the compiled files which you can copy to your live application folder on your web server. The first time you click to open the web browser for the live site it will compile, which may take a minute. This only happens the first time you load the site, and each time you edit the web.config for the site it recompiles, otherwise subsequent queries will load much faster.

(Versions prior to 4.047.000):

1. Download the software zip file. Extract all files by entering the zip password. Open the “SelectSurveyNET” folder and browse to the “Source Code” folder. Copy all contents of the “Source Code” folder to a folder in IIS such as c:\inetpub\wwwroot\SelectSurveyNETSource
2. Create an alternate folder for the published files:
3. C:\inetpub\wwwroot\SelectSurveyNETPrecompiled
4. Open Visual Studio. Click “open web site” from the file menu, browse to the folder “C:\inetpub\wwwroot\SelectSurveyNETSource” (or where you put the files) and click “open”.
5. All files will display in Visual Studio, in the explorer panel on the right side RIGHT CLICK the top folder and click “BUILD”. The application should build with no errors. Right click the top folder and click “PUBLISH” and browse to the folder “C:\inetpub\wwwroot\SelectSurveyNETPrecompiled” this is the compiled files which you can copy to your live application folder on your web server. The first time you click to open the web browser for the live site it will compile, which may take a minute. This only happens the first time you load the site, and each time you edit the web.config for the site it recompiles, otherwise subsequent queries will load much faster.

Structure:

The “Classes” folder contains all classes.

The “ClientInclude” folder contains all javascript.

The “Resources” folder contains all buttons, images, icons and style sheet.

The “UserControls” folder contains the user controls for questions.

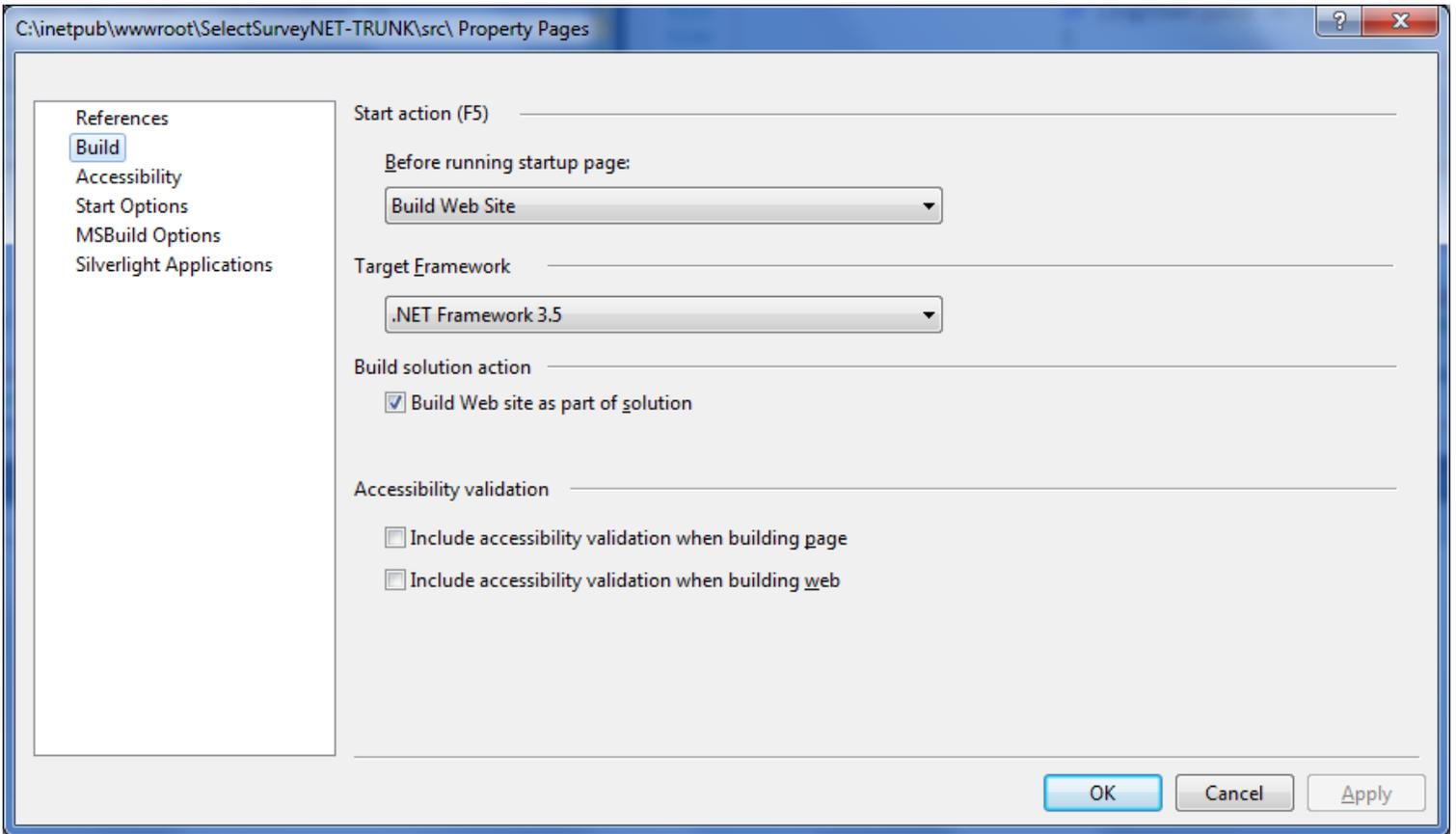
Base Pages:

BaseSelectSurveyNetPage.cs is inherited for all survey creation pages.

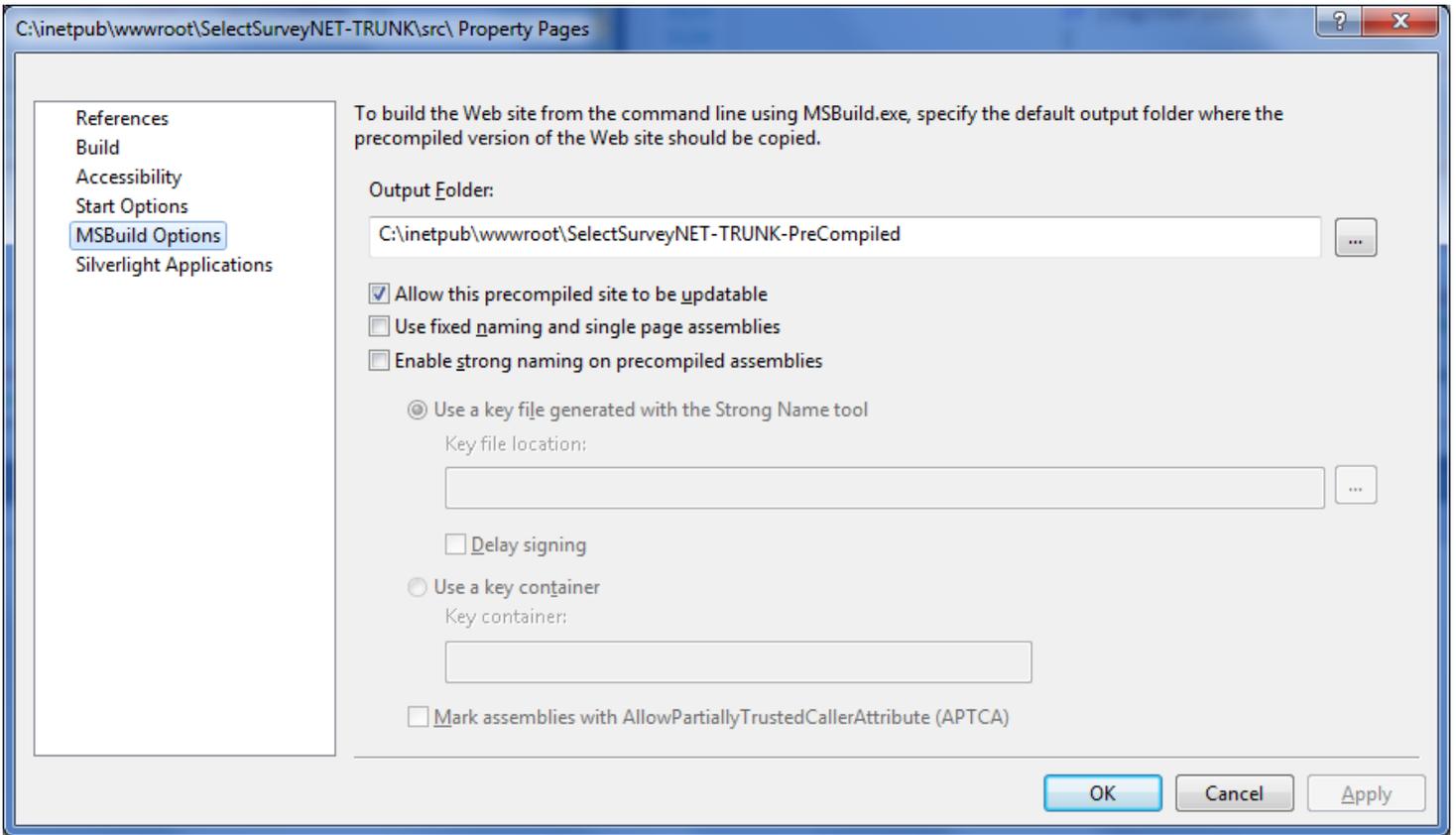
BaseTakeSurveyNetPage.cs is inherited for all take survey pages.

Header:

CtrlSurveyControlHeader.ascx



Note: Select the target .NET Framework version that is required by the version of software that you downloaded. i.e. v4.153.000 requires .NET 4.5.



Database Model

SelectSurvey.NET uses a UniDataLayer class (in the Classes folder) for data access to the supported databases:

- MS SQL Server
- MySQL
- Oracle



- MS Access (Supported prior to version 4.048.000)

This document does not contain information about the database model. The data model is discussed at length in the accompanying document, "SelectSurveyNETDatabaseModel.pdf".

Look and Feel

Look and Feel of Surveys:

Surveys use the unlimited survey template feature. You can paste in CSS stylesheets into the box provided on the edit template page in the admin interface of the application.

Look and Feel of Administration Interface:

SelectSurveyNET has a professionally designed user interface, with an emphasis on ease of use. Since the color scheme, navigation, header, and footer are standardized for all pages in the application, the software uses a style sheet and include files to control the look and feel.

All look and feel related attributes (color, fonts, sizes, etc.) are defined in the style sheet, which is located in the Resources/StyleSheet subfolder. By making changes to this file, the entire look and feel of the software can easily be modified.

The layout of each page is controlled by a UI framework, which is defined in the files:

- CtrlSurveyControlHeader.ascx
- CtrlSurveyControlFooter.ascx
- SelectSurvey.master

By modifying these files, changes can be made on a global level to the entire application.

Currently there are many built in features in the application interface that allow changing the logos, navigation text and footer without requiring any code changes. All of these things are stored in the database

Changes to the navigation can be made by modifying the file CtrlSurveyControlHeader.ascx

Changes to the footer can be made by modifying the file `CtrlSurveyControlFooter.ascx`.

Common Modifications

Modifying the Application Security

The software uses its own table structure to store information about all users of the application. For your own installation, you may already have a user database, and would prefer to integrate with your existing user database rather than replicate your users in the survey software.

Active Directory and LDAP Integration has already been coded as an option in the application, so see the installation manual for installing in Active Directory logon mode.

NTLM Pass through authentication has already been coded as an option in the application, see installation manual for configuration.

Single Sign On with SQL Membership Provider has already been coded as an option in the application, see installation manual and `web.config` for configuration.

The software uses a table called `SUR_USER` to hold the user information, a table called `SUR_ROLE` for the list of roles, and a table called `SUR_USER_TO_ROLE_MAPPING` to map users to roles.

All references to these tables need to be updated, so a global search for these table names (particularly the `SUR_USER` table) will find all references to existing security.

The most common situation where this arises are customers who have their own database table of users. The required changes are as follows: (it is advised to use the built in features listed above where no coding is required)

- Most user-related functions are located in the `App_code/Classes/Users.cs` file. Modify this file so that all functions return the appropriate data from the correct source.
- Modify the `Login.aspx/cs` or `LoginActiveDirectory.aspx/cs` file, which is the file that checks the security for each login attempt.
- In the file `App_code/Classes/SurveySecurity.cs`, there are several function related to the user's security level that will need to be modified.
- Finally, do a global search for `SUR_USER` to find all instances where the user information is reference in the application.

Running the Software under HTTPS

There is a built in function in the “admin tools” “global application settings” that allows you to force the site to run in HTTPS. When this is set if a user types in a non https URL manually, it will automatically redirect to https url. You must already have a current and up to date SSL certificate installed on the server for this to work.

Adding a New Item Type

To add a new item type, the best place to start is to find the item type that is already in the application that is most similar to the new item type. Then, in the App_code/Classes/Constants.cs file, find the constant for the similar item, and do a global search in all of the code for instances of that constant.

In short, you will need to make changes and additions in all of the places where the constant for the item to mimic appears.

That being said, the overall steps to add a new item type are:

- Add a new entry to the SUR_ITEM_TYPE table.
- Modify the App_code/Classes Item.cs file to accommodate the new item type. Make sure to have the appropriate client-side JavaScript in the Item.asp file, make sure that the appropriate sections of input controls are displayed for the item, and make sure the appropriate SQL is generated to save the item on the ItemAction.asp page.
- Copy the file UserControls/[question type name].ascx with the closest question type, rename and modify to be the new question type.
- Modify TakeSurveyAction.aspx/cs to receive responses to the new question.
- Modify the individual and overview (and all other) reports to display data for the new item type.
- Modify the data export routine to accommodate responses to the new item type.

Modifying Buttons and Images

All of the buttons, icons, and images used in the application are located under the Resources subfolder. You can easily modify any of these graphics by replacing the files already in these folders. All icons are font-awesome fonts (<https://fontawesome.github.io/Font-Awesome/cheatsheet/>), to allow for faster page loading.

To assist in modifying buttons, the PSD files for all buttons used in the survey application are located in the Documentation/Manuals subfolder in a file called, “SelectSurveyASPAdvancedButtonSources.zip”.

When making new buttons, note that all buttons with the word “Small” just before the .gif extension are short buttons, and all other buttons are full height buttons.

For both types of buttons, note that the widths of the buttons vary.

Note that CSS buttons are/can be used instead of graphical image buttons in survey templates. Click “edit template” in the survey administration area after clicking “Templates” from the main top bar navigation and paste in the CSS text into the box provided to edit/create new survey look and feel templates.

Increasing the Length of the Answer Fields

Newer versions:

No code change required, click to edit the question item through the survey interface and set the field width there through the interface.

Older versions:

Two changes are required to increase the length of answer fields for checkboxes, option buttons, dropdowns, etc.

To do this, edit the App_code/ UserControls/[question type name].ascx/cs file for the particular question to set the “maxlength” field.

Second, the length of the field in the database needs to be updated to the same value. In the SUR_ITEM_ANSWER table, update the length of the ANSWER_TEXT column to the same width.

Translating into a Foreign Language

SelectSurvey.NET contains admin tools which allow you to translate and type in javascript and other dynamic text into each language available on the survey options page. To do this click the “Admin Tools” icon that looks like a tool in upper right corner of the navigation when logged in as admin. If the icon is not displaying, turn on the visibility in the web.config by setting “Show admin tools=yes”.

- On the admin tools page, scroll down to the “Globalization and multi-language options” section.
- Click either of the links there to translate text in the application.

After you are finished setting the translated text, then set the language on the survey from the survey options page.

If you need to create custom buttons, the buttons sources are in the Documentation/Manuals subfolder in a file called, “SelectSurveyASPAdvancedButtonSources.zip” and those buttons can be referenced from the Survey template. You can create a survey template for each language.

Integrating with an Existing Database

When using MS SQL Server as the database, many developers would prefer to have the tables in an existing database rather than create a new database solely for the survey application. For this reason, all tables in the application have a prefix of SUR_ in order to make them easy to find and have them group together within another database.

Once the tables are installed in the database, you should create a separate database user that only has permissions set for these tables. Then, connect to the database as you would any other database, and make sure to use this new user in the connection.

Ajax Control Toolkit

Versions v4.069.000 and later utilize the open source Ajax Control Toolkit for iPad and touch and smartphone question types. There is a .net 3.5 and a .net 4.0 version of the toolkit. At v4.069.000 we are using the 3.5 version so that all clients are supported. Later versions are using .net 4.0 or later.

Information and to download the ajax control toolkit can be found here: <http://www.asp.net/ajaxlibrary/act.ashx>

To install the Ajax Control Toolkit into the Visual Studio Toolbox:

1. Download the Ajax Control Toolkit from link on this page for .net 3.5
<http://www.asp.net/ajaxlibrary/act.ashx> Download the binary version and not the source version.
2. Unblock and extract the files from the zip file
3. Launch visual studio and create a new ASP.NET website.
4. Open default.aspx in the visual studio editor.
5. Right click the toolbox in the left panel and select "Add Tab". Name the new tab "Ajax Control Toolkit"
6. Right-click beneath the new tab and select the menu option "Choose Items...".
7. Click the "Browse" button and browse to the folder where you extracted the Ajax Control Toolkit. Pick the AjaxControlToolkit.dll and click "OK" button to close the "Choose Toolbox Items" dialog.

The Ajax Control Toolkit should show up in the toolbox. Close the solution and go back to your survey solution.

Visual Studio Structure

Below is an example of how the projects appear in Visual Studio.

The “App_Code” folder contains the class files for business logic layer. The “Common/Data” folder contain the data layer.

