



# Select Survey .NET

Developers Manual



# SelectSurvey.NET Developers Manual

(Last updated: 11/10/2011)

Table of Contents:

<b>SelectSurvey.NET Developers Manual</b> .....	<b>1</b>
<b>Overview</b> .....	<b>2</b>
<b>General Design</b> .....	<b>2</b>
<b>Debugging Source Code with Visual Studio</b> .....	<b>2</b>
<b>Database Model</b> .....	<b>7</b>
<b>Look and Feel</b> .....	<b>8</b>
<b>Common Modifications</b> .....	<b>9</b>
<b>Modifying the Application Security</b> .....	<b>9</b>
<b>Running the Software under HTTPS</b> .....	<b>10</b>
<b>Adding a New Item Type</b> .....	<b>10</b>
<b>Modifying Buttons and Images</b> .....	<b>10</b>
<b>Modifying Existing Surveys with Responses</b> .....	<b>11</b>
<b>Increasing the Length of the Answer Fields</b> .....	<b>11</b>
<b>Translating into a Foreign Language</b> .....	<b>11</b>
<b>Integrating with an Existing Database</b> .....	<b>12</b>





## 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](mailto: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.

## General Design

Version 4.026.00 and higher is built with Visual Studio 2010 .NET 3.5

Version's before 4.026.000 used Visual Studio 2005 and .NET 2.0

Versions before 2.0 used Visual Studio 2003 and .NET 1.1

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

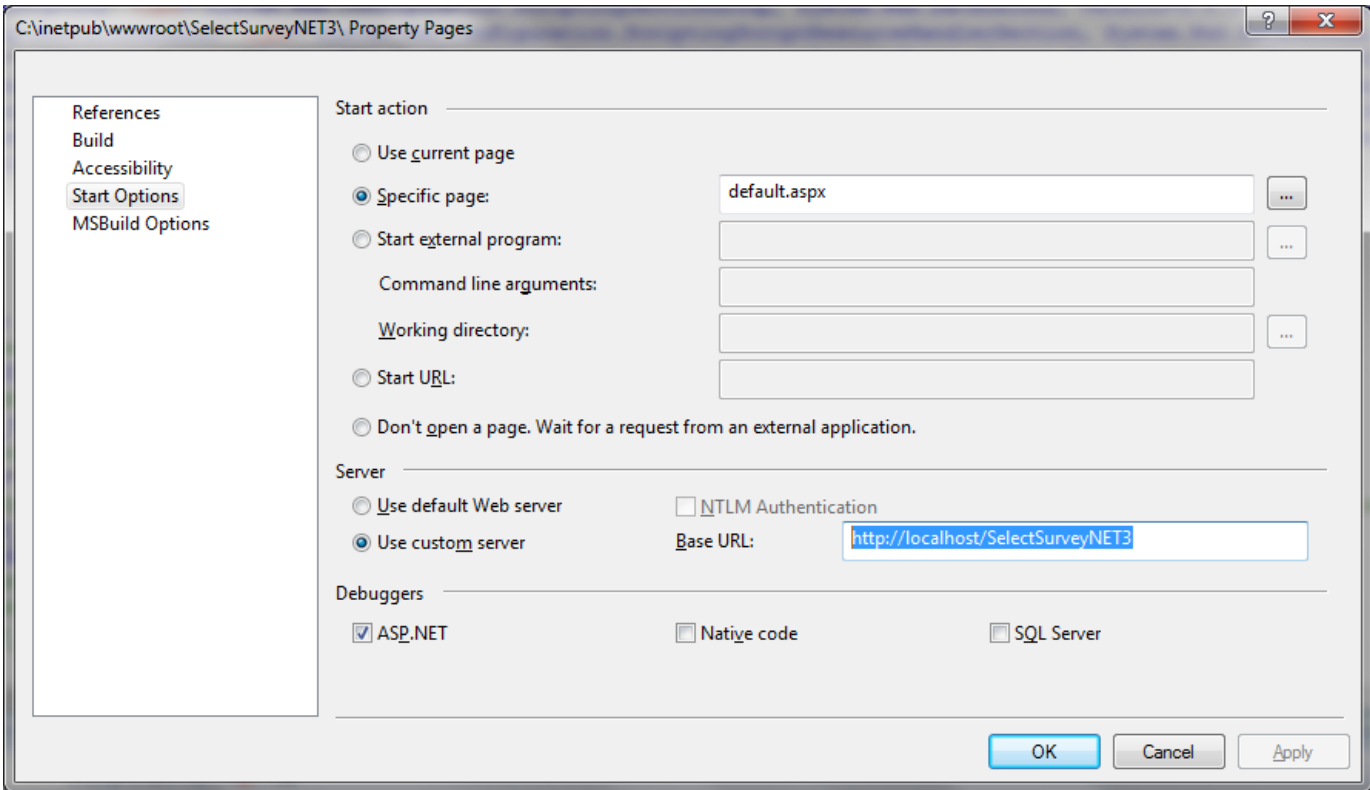
## Debugging Source Code with Visual Studio

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.

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

(versions after 4.047.000):

1. create new solution file in visual studio 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"
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.

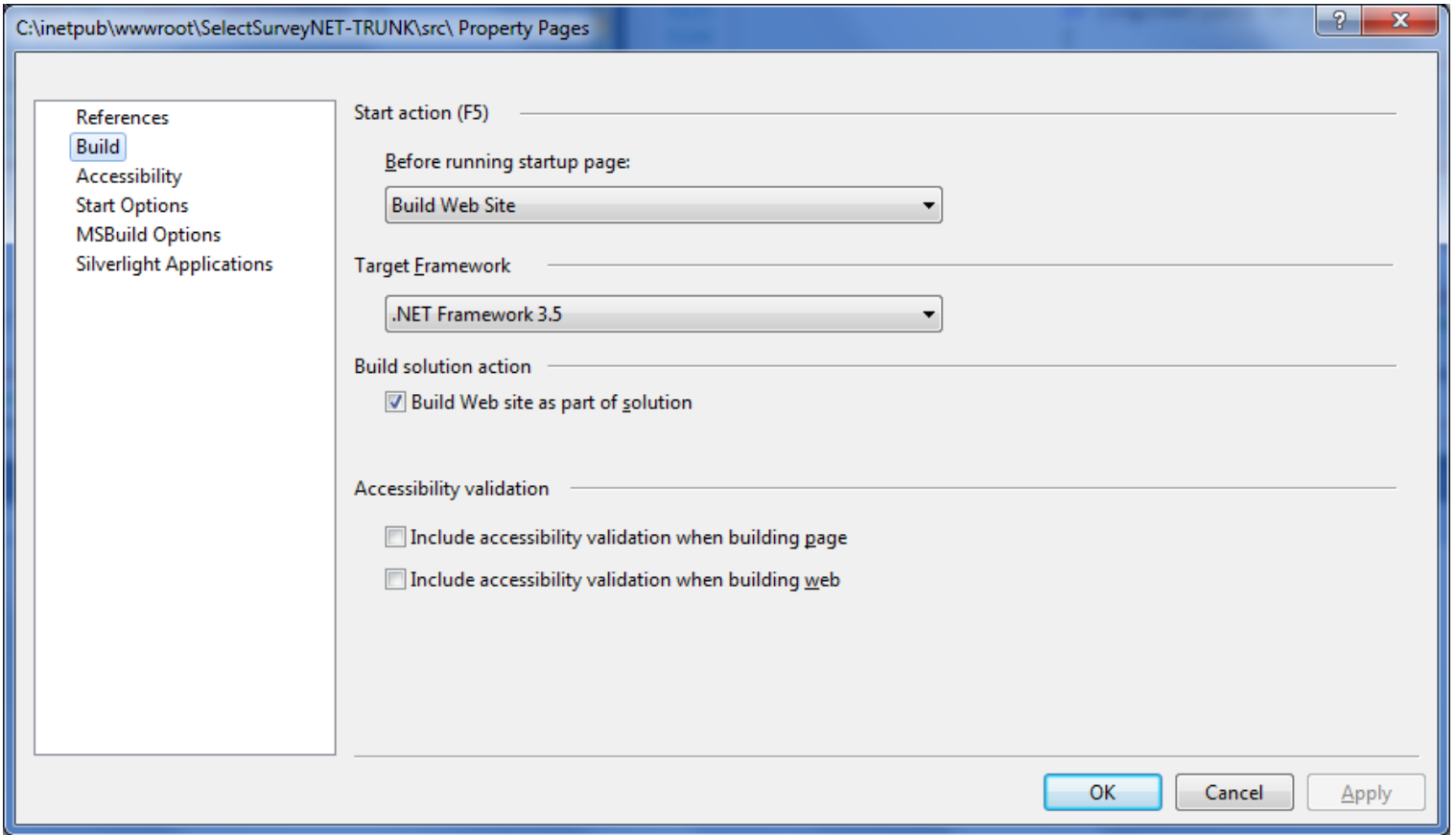
The “Resources” folder contains all buttons, images, icons and style sheet.

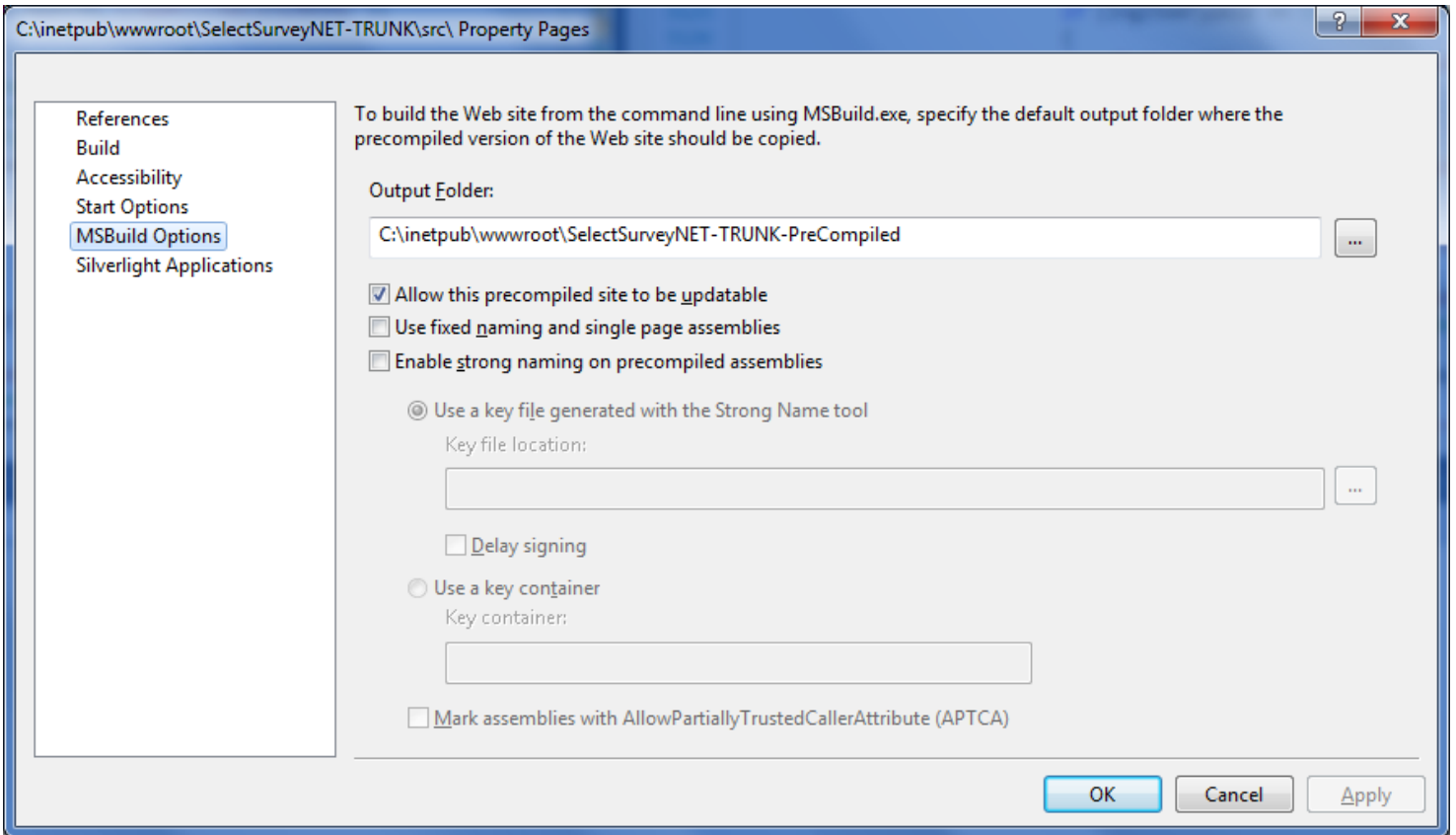
The “UserControls” folder contains the user controls for questions.

Base Pages:

BaseSelectSurveyNetPage.cs is inherited for all survey creation pages.







## Database Model

SelectSurvey.NET uses an 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

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.

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” where you can specify to run the site 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.

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.

## *Modifying Existing Surveys with Responses*

The software is designed so that surveys that have responses cannot be modified. The reason for this decision is that modifying a survey with responses is quite complicated, since adding questions means that earlier respondents never saw a question. Similarly, modifying the possible answers on a question with predefined responses can break the reporting, as earlier users may have selected an option that was later removed.

However, if you really need to modify a survey that already has responses, you can do so. If you decide to modify the survey, it is much better to remove questions than to add questions, as doing so has less potential to harm the survey.

Also, do not in any way add or remove values for predefined answers or rows. The only change that can be made to these values is to change spelling or syntax.

Finally, be sure to back up the database before making any changes, and do this 100% percent at your own risk. We cannot ensure that this will work correctly, nor can we be responsible for any surveys that are corrupted in this manner.

## *Increasing the Length of the Answer Fields*

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.