
Visual WebGUI to Wisej migration white paper

Step by step guide for migrating Visual WebGUI applications to Wisej

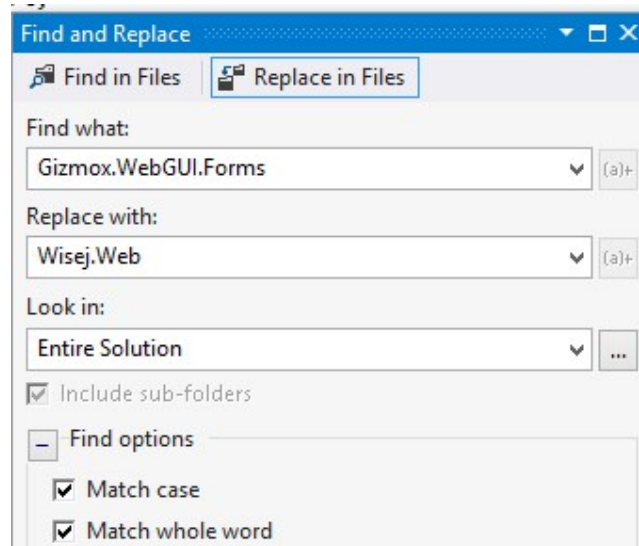
This white paper describes the steps for migrating an existing VWG application to Wisej – the first Web Integrated Server Environment for creating Real Time Web Applications. For creating a new Wisej application from scratch, check this tutorial: <http://wisej.com/docs/html/HelloWorldTutorial.htm>. If you're interested in an automatic conversion of your existing applications (WinForms, VB6, or even Visual WebGUI) please contact us at <http://www.fecher.net/contact>

1. Prerequisite: Install Wisej from <http://wisej.com/>
2. Change the project type from a VWG to a Wisej project. There are two ways to do this:
 - a. Create a new Wisej project and copy over all the files from the original WinForms application; if you choose this approach then you can skip steps 3 and 4 as the assemblies and the configuration files will be added automatically;
 - b. Change the settings of the existing project. You can do this by editing the .csproj file in a text editor and make the following changes:
 - Remove the {563295b5-8906-4a76-be2d-ff8e711c1204} project type GUID from the ProjectGuid node.
3. Open the VWG solution and add the references to the Wisej assemblies: Wisej.Web.dll and Wisej.Core.dll. Note: if these assemblies doesn't appear in the list check the target .NET framework version; it has to be at least .NET 4.5. Remove all references to VWG assemblies.
4. Add the configuration files to the project. You will need the following files:
 - a. Default.html
 - b. Default.json
 - c. Web.config

All these files are generated by default when creating a new Wisej project, so you can simply copy them over from a new project. You can find a detailed explanation of these here: <http://wisej.com/docs/html/Configuration.htm>

Although the Web.config file already exists in VWG application this is different than the configuration file for a Wisej project, so this has to be replaced too.

- Replace all occurrences of Gizmox.WebGUI.Forms with Wisej.Web. Remove the remaining usings of Gizmox.WebGUI.*



- Build the application and resolve the compiler errors: in most of the cases you will get compiler errors due to some missing properties and/or methods that are obsolete in Wisej. You can simply comment out these. There are also enumerations that have been changed to be more meaningful for a web application. Below is a table with the replacements for some of these:

FormStartPosition.WindowsDefaultLocation	FormStartPosition.DefaultLocation
FormBorderStyle.FixedDialog	FormBorderStyle.Fixed
BorderStyle.FixedSingle	BorderStyle.Solid
BorderStyle.Fixed3D	BorderStyle.Solid
Shortcut.Del	Shortcut.Delete
Shortcut.ShiftDel	Shortcut.ShiftDelete
Shortcut.CtrlIns	Shortcut.CtrlInsert
Shortcut.ShiftIns	Shortcut.ShiftInsert
Shortcut.AltBksp	Shortcut.AltBackspace

7. Change the startup method in the Default.json file:

```

{
  "url": "Default.html",
  //"startup": "WebApplication1.Program.Main, WebApplication1"
  "startup": "John.SocialClub.Desktop.Program.Main, John.SocialClub.Desktop"
}

```

For a native VWG application there is usually a start form specified in the Web.config file and set also in the Web section of the project properties as Start Action – Specific Page. The Start Action must be changed to Current Page and the form has to be created in the Program.Main method.

For VWG applications that have been migrated from WinForms it is common practice to create a Main page that calls Program.Main. This is not needed anymore; the Main method can be set directly in the Default.json file as described above.

8. VWGContext replacements:

Gizmoz.WebGUI.Common.Global.Context	Application.Session
VWGContext.Current.MainForm	Application.Session.MainForm
VWGContext.Current.Cookies	Application.Cookies

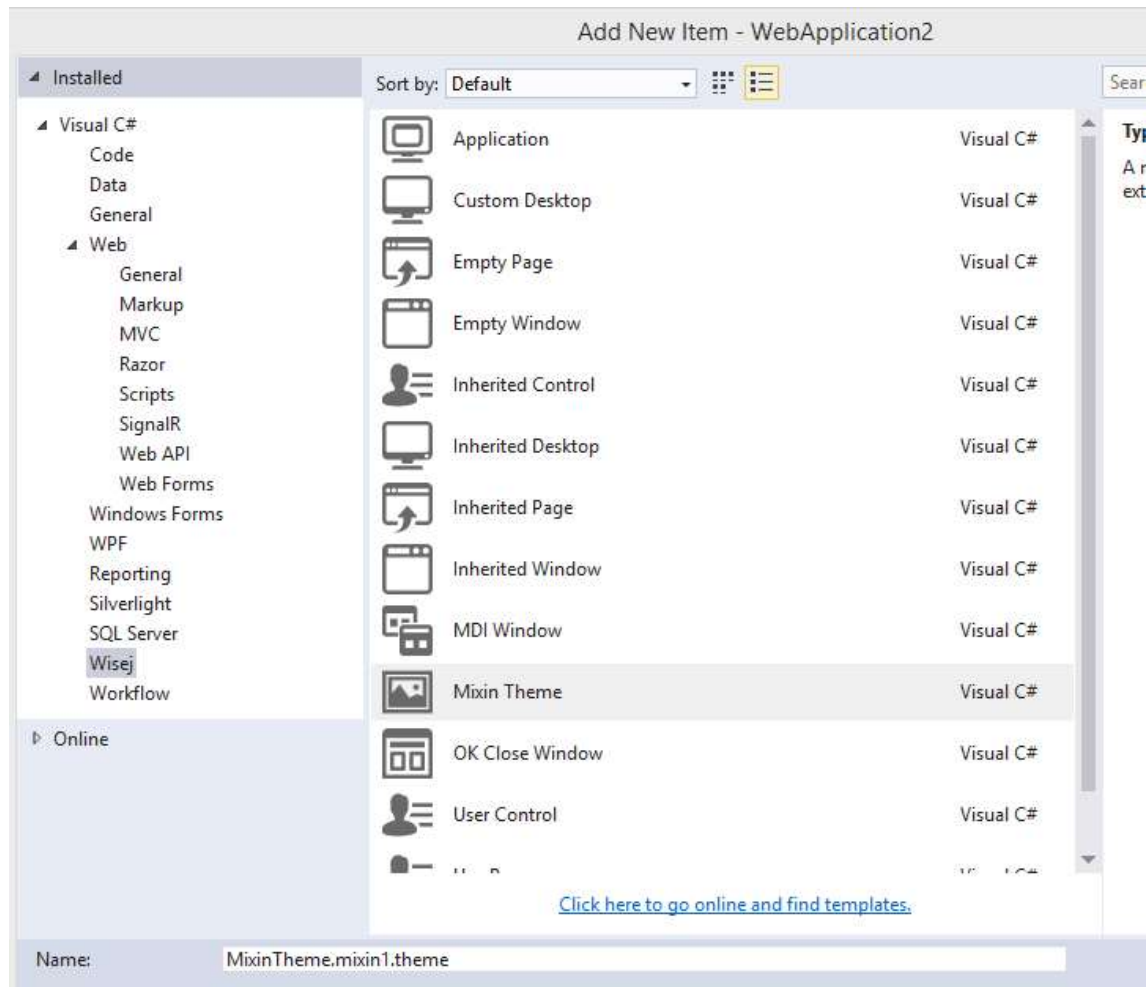
9. Controls customization

In VWG, controls customization was done either in the theme or by using skinnable controls. In Wisej, the theme can be changed using the Theme Builder (<http://wisej.com/docs/html/Themes.htm#ThemeBuilder>). Instead of the skinnable controls, Wisej offers theme mixins (<http://wisej.com/docs/html/Themes.htm#Mix-ins>).

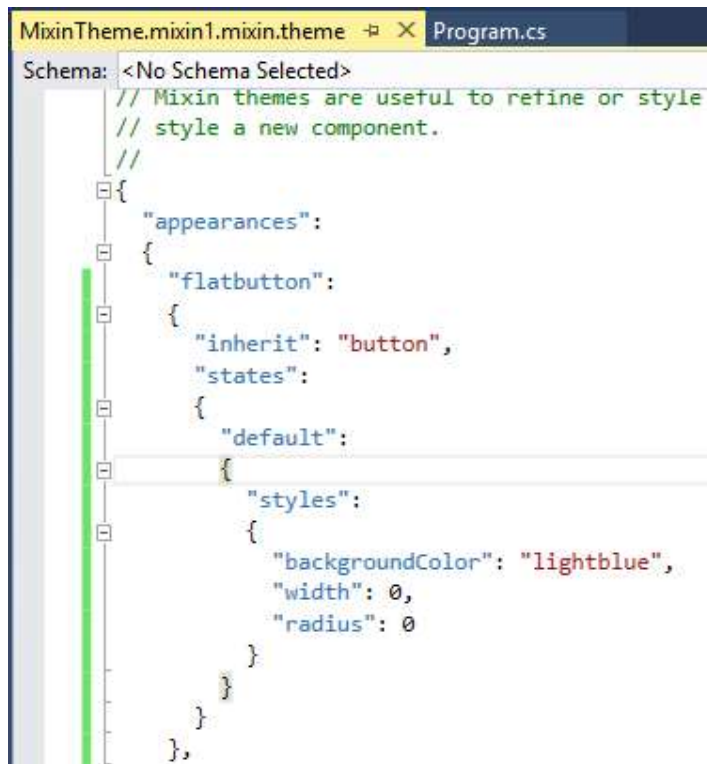
For example, let's consider a VWG application with a custom menu consisting of some flat, borderless buttons. To remove the border of a button, one would have either set the FlatAppearance or created a skinnable button and set there the properties.

In Wisej the FlatStyle and FlatAppearance properties are not available, but they are not needed at all because all the visual properties can be changed using mixins.

a. First create a new mixin:



- Set the appearance key; this will be used for the button's AppearanceKey property; only the buttons that have this key set will have the border removed.
- Add "inherit" : "button" to inherit all the styles/properties from the default button implementation.
- To remove the border we have to set the "width" style to 0 (this is equivalent to [0,0,0,0]) and also set the "radius" to 0, to get rid of the rounder corners.
- Finally we have to set the AppearanceKey property for the buttons that we want: in our example the key is "flatbutton"



```
MixinTheme.mixin1.mixin.theme  Program.cs
Schema: <No Schema Selected>
// Mixin themes are useful to refine or style
// style a new component.
//
{
  "appearances":
  {
    "flatbutton":
    {
      "inherit": "button",
      "states":
      {
        "default":
        {
          "styles":
          {
            "backgroundColor": "lightblue",
            "width": 0,
            "radius": 0
          }
        }
      }
    }
  }
},
```