Wisej Migration

Basic Migrations

1. Install Wisej from https://wisej.com

2. Change the project type from a WinForms executable to a Wisej web project.
   There are two ways to achieve this:

   a) Create a new Wisej project and copy over all the files from the original Win-Forms 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

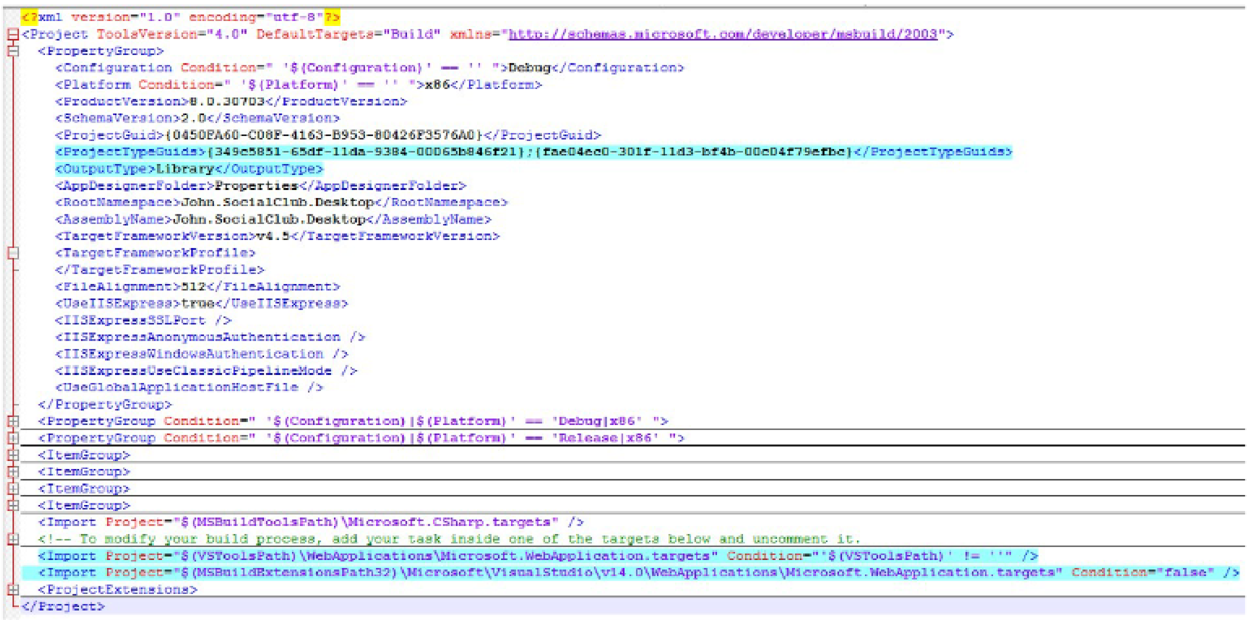
   - Add the project type GUIDs under the ProjectGuid node:

   ```
   <ProjectTypeGuids>{349c5851-65df-11da-9384-00065b846f21};{fae04ec0-301f-
   11d3-bf4b-00c04f79efbc}</ProjectTypeGuids>
   ```

   - Change the OutputType from WinExe to Library
   - Add the import nodes for the web applications under the main Project node:

   ```
   <Import
   Project="$(VSToolsPath)\WebApplications\Microsoft.WebApplication.targets"
   Condition="'$(VSToolsPath)' != ''" />
   <Import
   Project="$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v14.0\WebApplica
   tions\Microsoft.WebApplication.targets" Condition="false" />
   ```

   Summary of changes:



```xml
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="4.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">x86</Platform>
    <ProductVersion>8.0.30703</ProductVersion>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>{0450FA60-C08F-4163-B953-80426F3576A0}</ProjectGuid>
    <ProjectTypeGuids>{349c5851-65df-11da-9384-00065b846f21};{fae04ec0-301f-11d3-bf4b-00c04f79efbc}</ProjectTypeGuids>
    <OutputType>Library</OutputType>
    <AppDesignerFolder>Properties</AppDesignerFolder>
    <RootNamespace>John.SocialClub.Desktop</RootNamespace>
    <AssemblyName>John.SocialClub.Desktop</AssemblyName>
    <TargetFrameworkVersion>v4.5</TargetFrameworkVersion>
    <TargetFrameworkProfile>
    </TargetFrameworkProfile>
    <FileAlignment>512</FileAlignment>
    <UseIISExpress>true</UseIISExpress>
    <IISExpressSSLPort />
    <IISExpressAnonymousAuthentication />
    <IISExpressWindowsAuthentication />
    <IISExpressUseClassicPipelineMode />
    <UseGlobalApplicationHostFile />
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|x86' ">
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|x86' ">
  <ItemGroup>
  <ItemGroup>
  <ItemGroup>
  <ItemGroup>
  <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
  <!-- To modify your build process, add your task inside one of the targets below and uncomment it.
  <Import Project="$(VSToolsPath)\WebApplications\Microsoft.WebApplication.targets" Condition="'$(VSToolsPath)' != ''" />
  <Import Project="$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v14.0\WebApplications\Microsoft.WebApplication.targets" Condition="false" />
  <ProjectExtensions>
</Project>
```

Changes to *.csproj file

3. Open the WinForms solution and add the references to the Wisej assemblies:
Wisej.Framework.dll

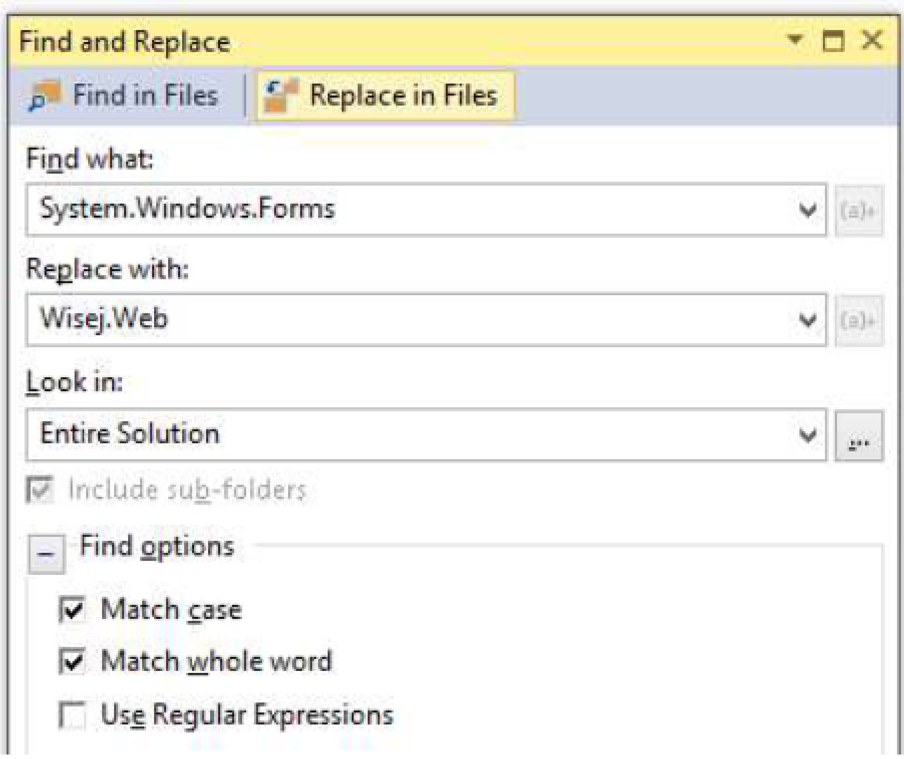
> ⓘ If these assemblies don´t appear in the list, check the target .NET framework version; it has to be at least .NET 4.6.

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

> You might also want to adjust the Title in default.html to match with your migrated project.

5. Replace all occurrences of System.Windows.Forms with Wisej.Web



Find what: System.Windows.Forms

Replace with: Wisej.Web

Look in: Entire Solution

☑ Include sub-folders

Find options

☑ Match case

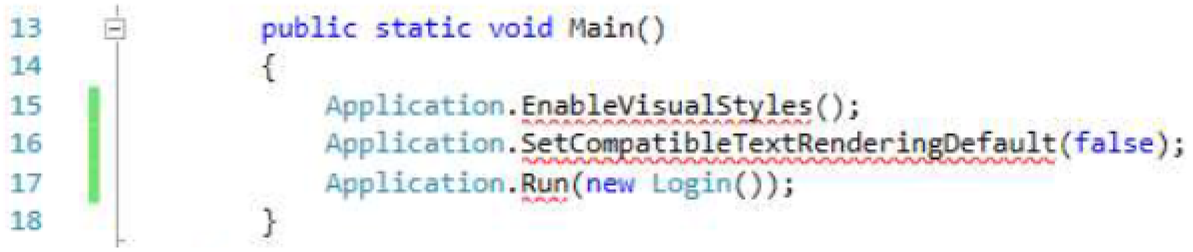
☑ Match whole word

□ Use Regular Expressions

6. 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 obso-lete in Wisej. You can simply comment out these.

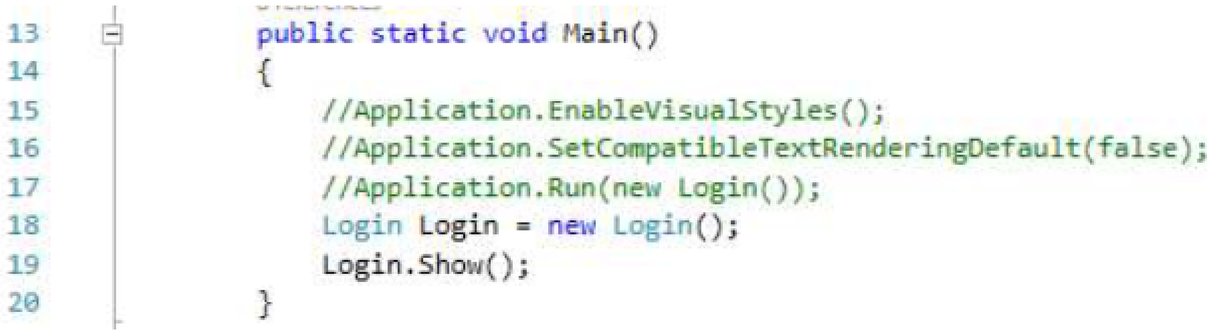
> If you find many similiar code lines to clean up, regular expressions in Visual Studio Replace can come in quite handy.
> Use for example `.*UseVisualStyleBackColor.*\n` to remove all those lines as they are not needed anymore.

7. Change the Main method; in a typical WinForms application you probably had something like this in the Main method:



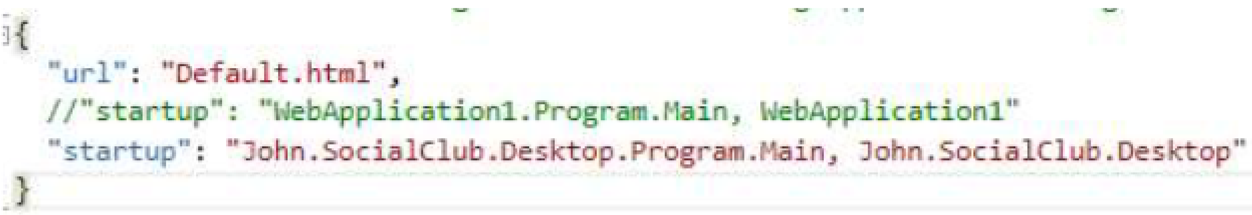
```
public static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Login());
}
```

The first two lines can be commented out as they are not needed in a web appli-cation. The third one has to be changed to show instead the Login dialog:



```
public static void Main()
{
    //Application.EnableVisualStyles();
    //Application.SetCompatibleTextRenderingDefault(false);
    //Application.Run(new Login());
    Login Login = new Login();
    Login.Show();
}
```

8. Change the startup method in the Default.json file to the Main method from Pro-gram.cs:



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

9. Check the Output Path. It should just be \Bin. If it´s \Bin\Debug please change it to \Bin

> ⓘ If you´ve been using ConnectionStrings in App.Config, copy them over to Web.Config.
> (in configuration section).

10. You´re done. Congratulations. You should now be able to run your Wisje web application.