

Saving User's Preferences in Window Form Application

One of the most common tasks for a Windows Application Developer is saving Users preferences like Windows form Size, location, color, font etc, so that when User exits the application and on next start-up, the application loads with previously saved settings.

In .NET 2.0 Window Form Application settings can be specified in settings section of project designer. To access **Settings** section, select your **Windows Project** in **Solution Explorer** --> **Right click** --> **Properties** --> **Settings**.

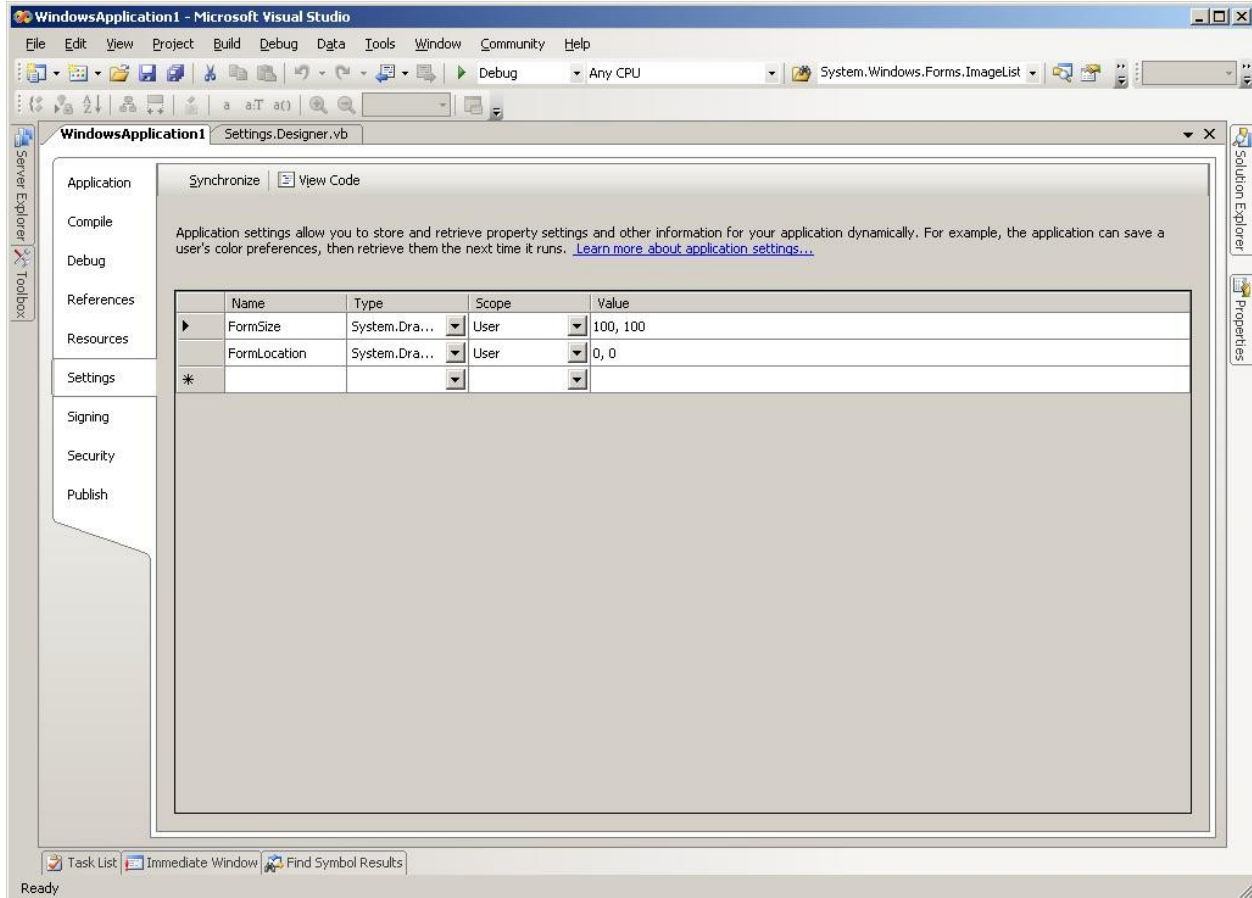


Fig. Settings section of Project Designer

Sample:

- For saving Window form size and location, I am adding two parameters (i) **FormSize** and (ii) **FormLocation** through settings of project designer.
- FormSize is of type System.Drawing.Size with default size set as 100,100.
- FormLocation is of type System.Drawing.Point with default size set as 0, 0.
- Both have **User** as scope. A scope can be defined as either **User** or **Application**. An application scope setting parameter cannot be modified by the User during Runtime and hence parameters like connection string which remains constant during runtime can be set to Application scope. A user scope setting parameter is per User specific and can be modified during runtime. For more details please refer the link in reference section.

- Now, we simply need to assign these settings in form load event of the form. This can be done as shown below:

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles Me.Load
        Me.Size = My.Settings.FormSize
        Me.Location = My.Settings.FormLocation
    End Sub
```

- We also need to save the current settings of the form at the form closing event . This is shown in the code below:

```
Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    My.Settings.FormSize = Me.Size
    My.Settings.FormLocation = Me.Location
    My.Settings.Save()
End Sub
```

This is a just simple example of saving User preferences. Various other application and user level settings can be saved as discussed above. Since the settings parameters are not kept in encrypted form, so keeping password and other sensitive information should be avoided else we can also encrypt and save.

Note:

In Visual Basic projects, Application **Settings** is accessible by **My.Settings** object and in C#.net projects it is by **Settings** object.

References:

1. <http://msdn.microsoft.com/en-us/library/system.configuration.applicationsettingsbase.aspx>
2. <http://blogs.msdn.com/rprabhu/articles/433979.aspx>
3. [http://msdn.microsoft.com/en-us/library/cftf714c\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/cftf714c(VS.80).aspx)