

How to create, add, edit and load resources

1. What is a resource?

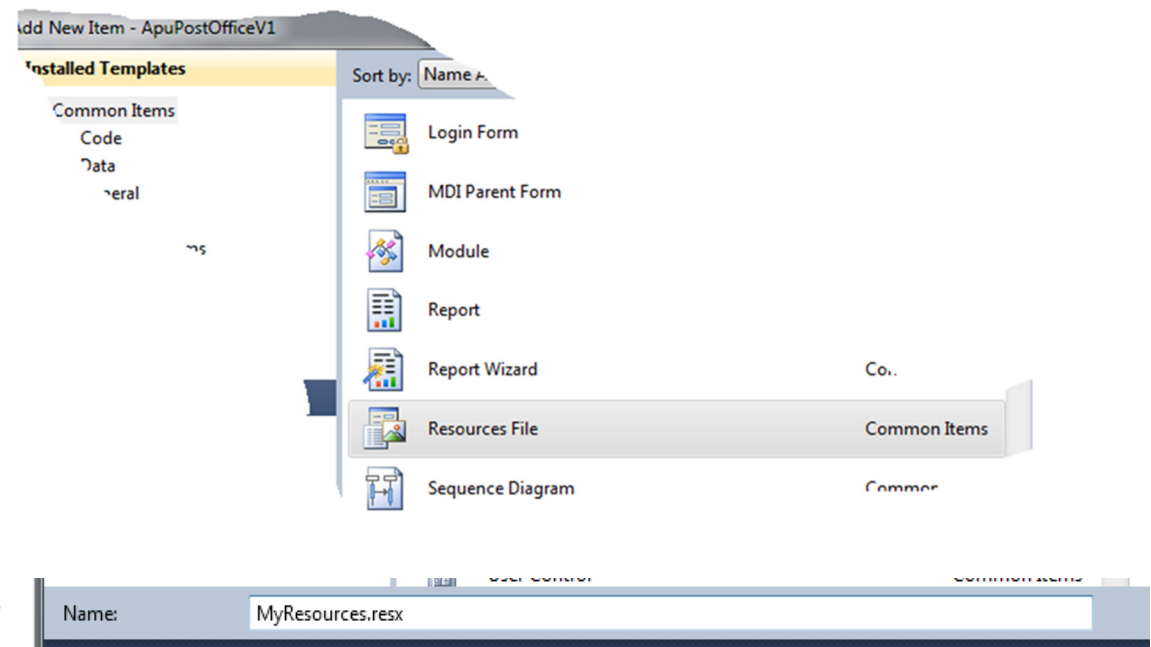
A part of a .NET application is not source code. Examples are strings, images, icons and other such data and are referred to as “**resources**”. Every project maintains its own resource data which is stored in XML format in a file with the extension **.resx** (named Resources.resx by default). Resources can also include text files, xml files, audio and video files. Resources can be linked or embedded into a **resx** file.

The resources are compiled into the assembly by a separate compiler called resource generator (resgen.exe). This compiler converts the resources into an internal class called Resources. Visual Studio creates a **resx** file for every Form you create, although the **resx** file may not be visible in the project explorer. If you check the project directory, you will find a file with the same name as the form but with the extension **resx** there. Strings, images, icons and other such components used in the form are saved in this file. For Visual C#, the Resource Designer generates strongly-typed resources in the **Properties** namespace (specifically, `ProjectName.Properties` in `Resources.Design.cs`).

2. Working with resources using Visual studio

As mentioned, Strings, images, icons and other such components are resources in your application. Visual Studio creates a resource file with the extension **.resx** in which you can embed such data. These will be then available for easy access to your application at run time.

- 2.1 The resources common for the whole application are found in the **Properties** folder in the Project Explorer in Visual Studio. Right-click on the **Properties**, select **open**, or simply double click on it and then select **Resources**.
- 2.2 For using embedded resources in Visual Studio, you must directly work with **resx** files. If you would like to have different resource files, you can create a **resx** file in your project from the templates in the same way as you add another new item. Right-click on the project name in the Project Explorer in Visual Studio, select **Add** and **New Item**.

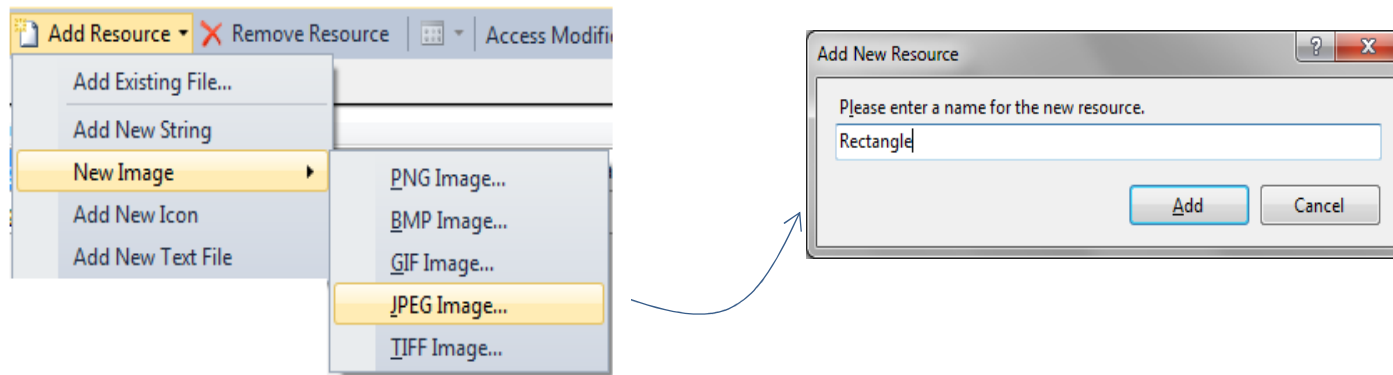


- 2.3 **Add an existing resource** (an image for example): You can add an existing resource in VS by, double clicking on the **resx** file. If you do not have a resx file, double-click on **Properties**, select **Resources**. Change the type of the resources to (for example) **Images**. You can then browse to the image file that you want to upload, as illustrated in the figure below. The figure below is a snap shot of the menus in VS's resource editor, where you can add an existing item. Select **Add Existing File** and then browse to the location on your computer where you have saved the file. It will be possible to edit the image after uploading, using the drawing tools in the resource editor (see the figure on the next page).

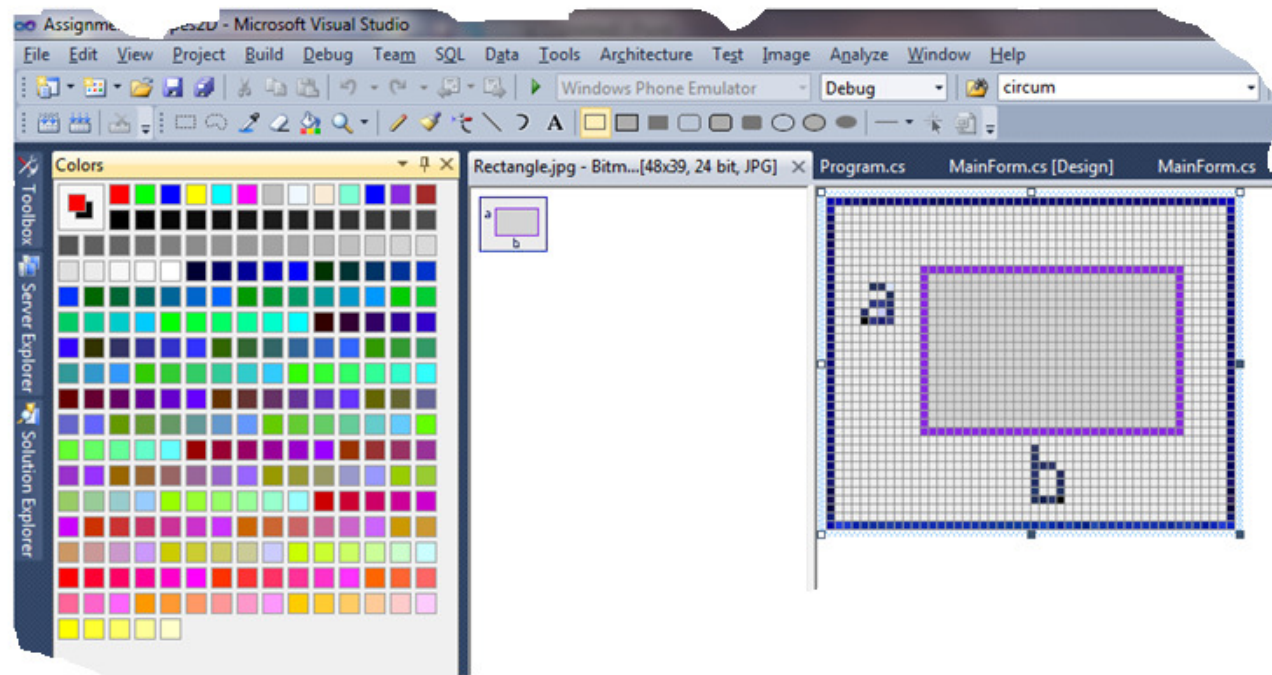
You can also add an image file (or other resource file) by selecting **Add, Existing Item** from the project menu. You have to add the item to the **Resources** as explained in the above step.

It is strongly recommended that you save all resources under Resources folder in the project directory on your computer. When you create or add a resource in your project, VS creates automatically the folder Resources. You can add a resource by right-clicking on the folder Properties in the Project Explorer in VS. Select the menu Open and then the tab Resources (the menu at left in the dialog box). There you find the menu for adding various types of resources (as shown in the figure below).

- 2.4 **Create a new resource:** To create a new resource, a string, an image or an icon for your application, you can select one of the **Add ...** submenus. To add a new string, select **Add New String**, and to add a new image select **Add New Image**
- 2.5 For strings, VS displays a table in which you can select an ID and a text value connected to the ID.
- 2.6 To create a new image, select an image type and a filename in which the image will be saved.



- 2.7 Using the simple tools available (as in the figure that follows), you can now draw your image. Most of the icons must be familiar to you. Otherwise, draw a test image and try the features available.



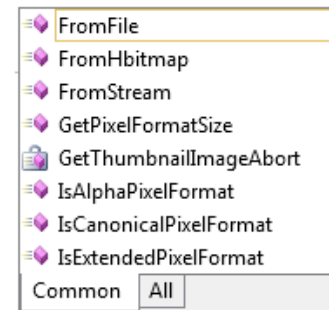
- 2.8 As Visual Studios Resource Editor offers only a set of primitive tools for creating, drawing and editing images, an alternative way is to add images and icons drawn with external tools. As far as icons are concerned, it is important to note that you would need to draw a 32 x 32 and a 16 x 16 pixels drawing for the same icon. This does not give you many pixels to draw on, but that is how life works.

2.9 **Load a resource:** To load a resource from the resources file in your code, you can use the following sample code:

```
pictureBox1.Image = Properties.Resources.happyface;
```

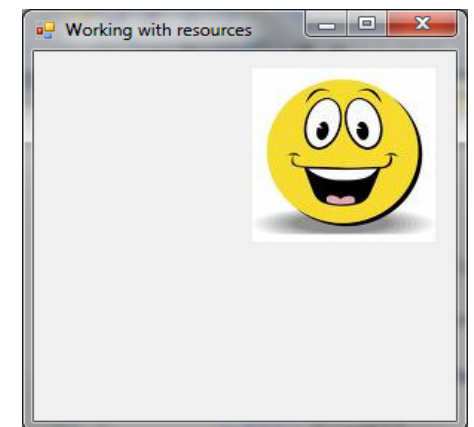
2.10 Images do not have to be embedded to the resource file. They can be loaded at run time from file, using the method **Image.FromFile**, as illustrated in the figure below. Browse to the file to location of the image to be loaded.

```
PictureBox1.Image = Image.
```



2.11 Make sure that you do not set the path to your image files to locations on your computer. If you do so, your program will work on your computer but not on others'. A recommended way is to save your image files relative to the location of your EXE file, as the example below:

```
private void ShowPicture()
{
    //Make sure that the file happyface.jpg is saved in the
    //Bin\DEBUG directory
    string filePath = Application.StartupPath + "\\happyface.jpg";
    pictureBox1.SizeMode = PictureBoxSizeMode.AutoSize;
    pictureBox1.Image = Image.FromFile(filePath);
}
```



2.12 Embedded resources are faster to access at run-time. Loading a resource from a disk (FromFile) requires disk access, which is normally a slower process.

- 2.13 It is also a very good practice to put all culture dependent strings (constant strings) into the resource file using the same menu as when creating a new image as explained in above. You can even use a separate **resx** files to store string in different languages. However, when working with different languages, .NET has a lot of support for that. Refer to **MSDN** or **google** on the Internet for information and tips.

3. Useful links

Accessing Application Resources (VB): [http://msdn.microsoft.com/en-us/library/tkkf2x4f\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/tkkf2x4f(v=vs.100).aspx)

Creating Resource Files: <http://msdn.microsoft.com/en-us/library/xbx3z216.aspx>

Good Luck!

Farid Naisan

Course Responsible and Instructor