

## **Tom's Tech-Notes**

### **Free Windows Programming Tools**

Visual Studio 2019 Community Edition

Author: Tom Burt, Vice-President, Sun City Summerlin Computer Club, NV

July 2019 issue, The Gigabyte Gazette

www.scscclb.club

tomburt89134 (at) cox.net

Most of our club members are content to simply use software programs that are provided with their computer's operating system, or programs that they purchase or download for free. However, a few may wish to have the freedom and flexibility that comes with being able to create your own custom programs to do some specific function or to use a preferred user interface. Programming an application for Windows can be daunting because so much of the coding effort goes into managing the screen, keyboard and mouse interactions.

For several years, Microsoft has been providing a free version of their Visual Studio developer tools called "Visual Studio Express" or, recently, "Visual Studio Community Edition". The newest version is "Visual Studio 2019 Community Edition". It includes the Visual Basic.Net, Visual C#.Net and Visual C++ languages, an extensive set of built in interactive components, plus an Integrated Development Environment with debugger and Microsoft's "Intellisense" assistance to make Windows programming easier than ever. It is designed to run on the Windows 7, 8.1, and 10 desktops. It can create many types of applications, including traditional desktop apps, "Modern/Metro/Universal" apps, web applications and even apps for Android and iOS mobile devices.

You can download Visual Studio 2019 Community Edition at the following hyperlink:

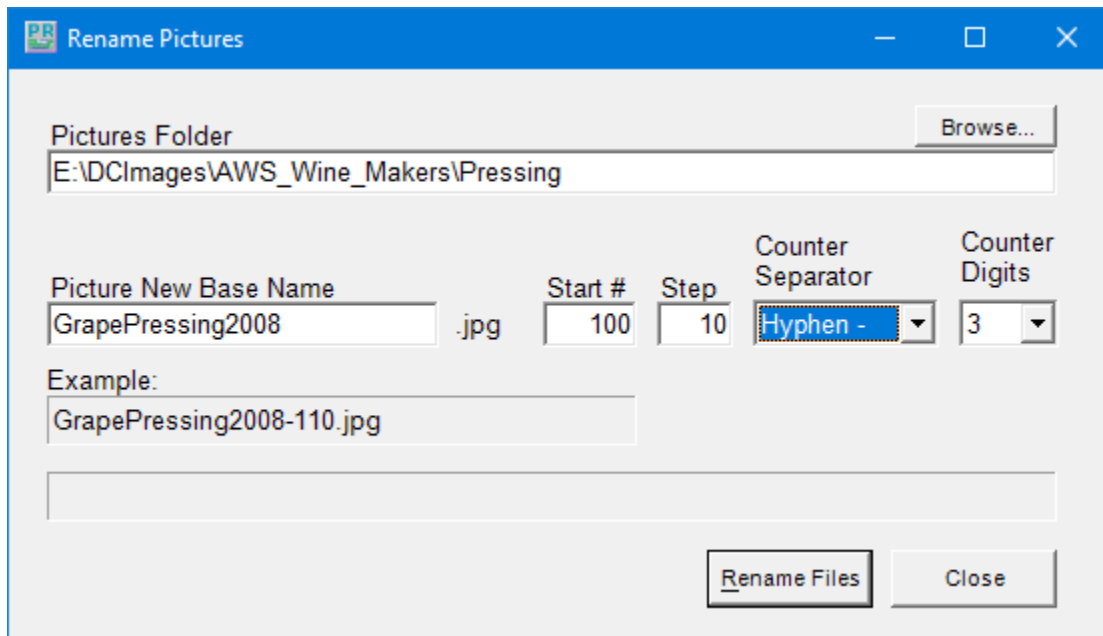
<https://visualstudio.microsoft.com/vs/>

Space here doesn't allow me to even scratch the surface of programming in Visual Studio, which uses the ".Net" framework runtime environment. However, for those interested, I've built PictureRenamer.exe, a simple bulk picture renaming tool using VB.Net 2019. I use this tool often to rename a set of images in a folder to something more meaningful than the "Img-nnnn.jpg" names that come from my cell-phone camera.

The picture renamer program lets you select a folder of images, set a base name, a starting number, a step increment, a separator pattern and the number of digits to use in the sequence number to use for the resultant file names. As these settings are changed, an example of the resultant file name is displayed.

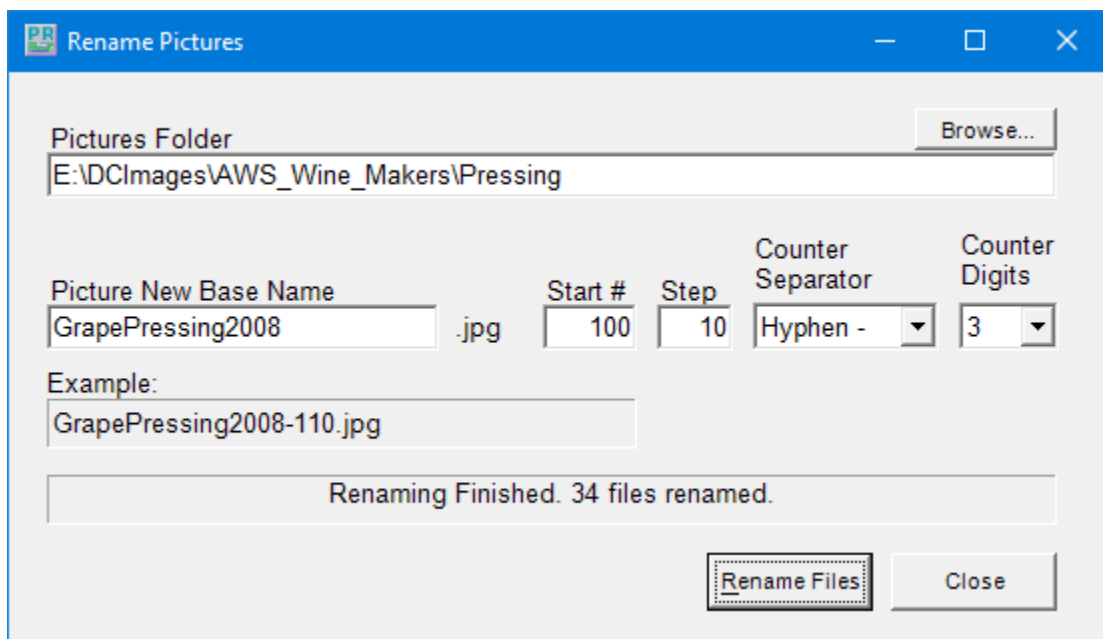
When the rename settings are suitable, the user clicks the "Rename Files" button and the program sets about renaming the image files in that folder. A file is considered an image file if its suffix is: .jpg, .png, .bmp or .tif. Files with other suffixes are ignored.

See the following screen shot to get an idea how this actually looks.



Picture Renamer Set Up to Perform Renames

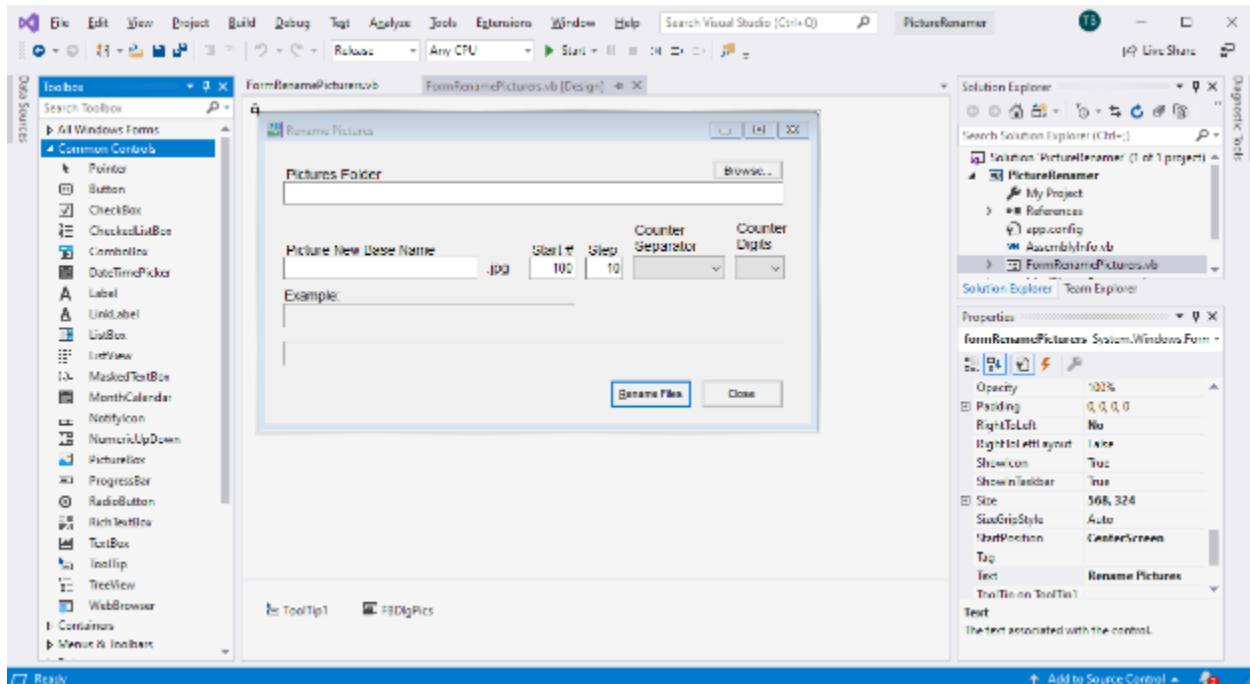
After the user clicks the “Rename Files” button, the program renames the qualifying image files in the folder using the base name, separator and successive sequence numbers.



Picture Renamer After Run

The Picture Renamer program is implemented as a Windows Desktop application. It has only a single screen. The “Browse” button opens a standard Windows “Select Folder” dialog that allows the user to pick the folder whose images are to be renamed.

To view or edit the program's project file and source code, you must first have installed Visual Studio 2019 Community Edition. Double click the PictureRenamer.vbproj file in the extracted set of files. From there you're off and running.



Visual Studio 2019 Developer Interface – Design View

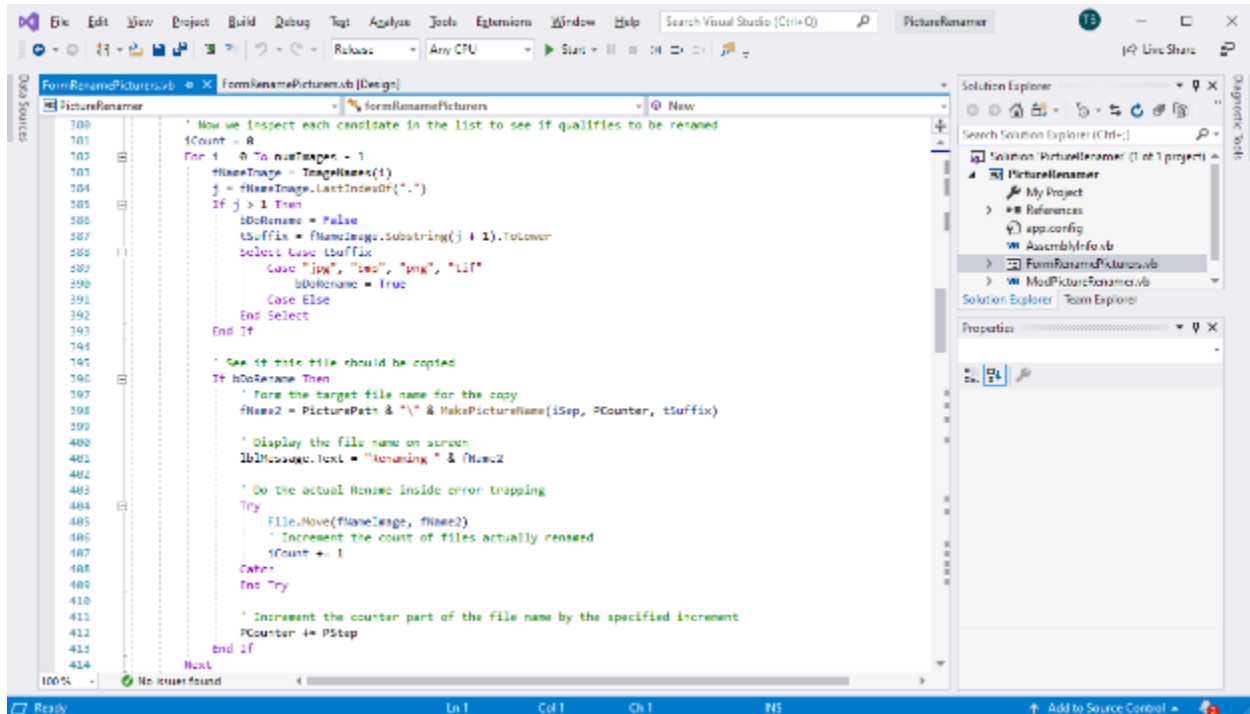
In the Design view, the program's window appears as it would look when it is running. The developer can drag and drop Windows "Controls" (buttons, text boxes, check boxes, labels, dropdown lists, ...) onto the program window and then set their properties by dragging with the mouse or by typing into the control's property sheet. Controls can be given meaningful names (e.g. btnRenameFiles instead of Button1) to make the underlying program code easier to understand.

Once the program's window is laid out nicely, with all the controls properly sized and aligned, the developer moves to the Code view and enters the program code needed to operate the user interface and perform the application's functions. When a user clicks on or types into a Control, that action is converted by Windows into a message that is sent to the application indicating which Control was acted on and what type of action it was (click, double-click, keypress). Each such message is received by the program and concerted by the .Net runtime to a function call to an event handler. The developer codes event handlers for actions he/she wants the program's UI to respond to in some special way.

In this example, they *key* action is the user clicking on the "Rename Files" button. That action triggers a Click event that is sent to the Click handler function for the "Rename Files" button. That Click event handler function executes a sequence of VB.Net code instructions that carries out the actual series of file renames for the image files in the

designated folder. In the programming world, this is called an event-driven, object-oriented model.

The following screen shot shows the code window displaying the key VB.Net code loop that does the image file renames. You can right click on it and choose “Copy Image” to copy the screen shot to an image editor like MS Paint to view it in more detail.



```
100 ' Now we inspect each candidate in the list to see if it qualifies to be renamed
101 iCount = 0
102 For i To numImages - 1
103     fNameImage = ImageNames(i)
104     j = fNameImage.LastIndexOf(".")
105     If j > 1 Then
106         bDoRename = False
107         tSuffix = fNameImage.Substring(j + 1).ToLower
108         Select Case tSuffix
109             Case ".jpg", ".bmp", ".png", ".tif"
110                 bDoRename = True
111             Case Else
112                 ' See if this file should be copied
113                 If bDoRename Then
114                     ' For the target file name for the copy
115                     fName2 = PicturePath & "\ " & MakePictureName(iSep, iCounter, tSuffix)
116                     ' Display the file name on screen
117                     lblMessage.Text = "Renaming " & fName2
118                     ' Do the actual Rename inside error trapping
119                     Try
120                         File.Move(fNameImage, fName2)
121                         ' Increment the count of files actually renamed
122                         iCount += 1
123                     Catch
124                     End Try
125                     ' Increment the counter part of the file name by the specified increment
126                     iCounter += iStep
127                 End If
128             End Select
129         End If
130     End For
131 Next
```

Visual Studio 2019 Developer Interface – Code View

I’ve uploaded the Picture Renamer program to the club website as a .zip file. The direct hyperlink is: <http://www.scscclub.com/smnr/PictureRenamer.zip>. Just click the link or copy and paste it into your web browser’s address bar. Your browser will download the .zip file to your PC’s hard drive. If you’re interested in getting the source code and project files, send me an email at tomburt89134 \*at) cox.net and I’ll send you a .zip file with the complete package.

Unlike many Windows programs, the Picture Renamer does not need to be *installed*; just copy it to some folder on your hard drive To *run* the Picture Renamer, just double-click on the Picture Renamer.exe file. To make a desktop *shortcut*, left click the PictureRenamer.exe file to select it, then right-click to bring up a context menu. Choose “Send to” and then “Desktop (create shortcut)”.

