**TIFFProcessor – A command line utility to process TIFF image files**.

**Overview:**

TIFFProcessor is a command line utility used to perform some basic operations on the TIFF image files such as, Merging two TIFF files with multiple pages, removing blank pages etc.. It can process TIFF files with multiple pages. It is written in C/C++ using Visual Studio 2017 on Windows 10 operating system. Details regarding the evironment and tools will be discussed later in this document.

**How you use this utility:** TIFFProcessor is an executable which perform operations based on the command line switches ( or parameters) provided to this utility.

**Usage:** TIFFProcessor.exe -<command-line-switch> inputfile.tiff outputfile.tiff

**Command Switchs and their operations:**

**-merge** : Merge two input TIFF files with multiple pages contained in them. The second input file is appended to the first input file.

Example: TIFFProcessor.exe –merge inputfile\_1.tiff inputfile\_2.tiff

Contents of “inputfile\_2.tiff” will be appended to the “inputfile\_1.tiff”, leaving the second inputfile intact.

**-rblank** : Removes all the blank pages (with no content in them) from the input TIFF file leaving only the pages which has content in them

Example: TIFFProcessor.exe –rblank inputfile.tiff outputfile.tiff

The “outputfile.tiff” is optional. If provided this output file parameter, it will write the result of the operation into the output file leaving the input file intact. If output file is not provided, it will perform the operation on the inputfile (i.e inputfile.tiff, in this case) and remove the blank pages from the original inputfile.

**-rpageno=p1,p2,p3,p4,p5**: Removes the pages associated with the page number(p1, p2 etc..) provided to this command switch.

Example: TIFFProcessor.exe –rpageno=1,2,3,4,5,… inputfile.tiff outputfile.tiff

In the above example, it removes the pages numbered 1, 2, 3, 4, 5 from the inputfile. The “outputfile.tiff” is optional. If provided this output file parameter, it will write the result of the operation into the output file leaving the input file intact. If output file is not provided, it will perform the operation on the inputfile (i.e inputfile.tiff, in this case) and remove the pages from the original inputfile.

**-fileinfo:** Displays basic information about all the pages in a TIFF file. The result will be written to an outputfile if one is provided as command line argument. Output file is an optional.

Example: TIFFProcessor.exe -fileinfo inputfile.tiff outputfile.txt

Sample output:

Total number of pages: 8

Total number of Blank pages: 0

Page Number: 1

----------------------

Width = 3840

Height = 2160

Layout(CONFIG) = 1

Phtometric = 2

Orientation = 1

Compression = 5

Bits Per Sample = 8

Samples Per Pixel = 3

**-rgb2gray**: Converts all the colour pages in an input TIFF file to Gray Scale images.

Example: TIFFProcessor.exe –rgb2gray inputfile.tiff outputfile.tiff

In the above example the colour pages in inputfile.tiff will be converted to gray scale pages and written to outputfile.tiff along with the other non-coloured pages. The page count will not be altered and will be the same in both the files. The “outputfile.tiff”, again, is an optional parameter. If no output file provided, the operation will be done on the original inputfile.

**-gray2binary -<threshold>:** TIFFProcessor.exe –gray2binary -150 inputfile.tiff outputfile.tiff

This command takes <threshold> as a numeric value, based on which the black & white values are estimated. In the above example, all the gray scale images in the inputfile.tiff will be converted to binary(black & white) images and written to the outputfile.tiff along with other non-gray scale images. The numeric 150 in the above example indicates, any value higher than 150 will be a “white” pixel and value lower than 150 will be a “black” pixel. The page count will not be altered and will be the same in both the files. The “outputfile.tiff”, again, is an optional parameter. If no output file provided, the operation will be done on the original inputfile.

**-about**: This command line switch displays the information about the utility such as, Name, version and author.

Example: TIFFProcessor.exe –about

Sample output**: Utility Name: TIFFProcessor.exe**

**Version: 1.8.2020**

**Author: Venkata Ananth Ganti**

**-help:** This command shows the entire information about the command line switches and their usage.

Example: TIFFProcessor.exe –help

**Note:** Mix and match of two or more command line switches is not permitted and lead to a graceful termination of the operation about to perform by displaying appropriate error messages.

**Setting parameters to TIFFProcessor:**

The user can set various parameters before using the TIFFProcessor application.

In current version(1.8.2020) the following parameters can be set:

1. The path for the TIFF files. Example: C:\Imagestore\TiffprocessorFiles\TIFFimages\
2. The compression type.
3. The threshold value while converting the files to Binary (Black & White) images.

These parameters can be set using “settings.txt” file which MUST exisit in the same directory/folder where TIFFProcessor.exe resides.

Settings.txt looks like this:

**imagefilespath=C:\VenkataGanti\Images\tifs\**

**compression=JPEG**

**threshold=150**

If any of the above parameter is not included in the settings.txt, the default values are used.

Default values:

**Imagefilespath = “”;**

**Compression=JPEG**

**Threshold=100**

**Future Improvements:**  There are few other improvements targetted for the next version of this utility. They are;

1. Range based deletion of pages. Providing range of page numbers to the command line switch “-pageno”.

Example: TIFFProcessor.exe –rpageno=1-4, 7-9 inputfile.tiff outputfile.tiff

1. Delete pages by colour, i.e. deleting only coloured(RGB) pages, or deleting only Grayscale images or binary images.
2. Supporting PALETTE photometric layout.
3. Converting from TIFF to other image formats like BMP.
4. Recognize content in the TIFF pages.
5. Adding UI and so on……

**Development Environment:**

“TIFFProcessor.exe” is written in C/C++ using Visual Studio 2017 on Windows 10 operating system. Following are the tools and dependencies used:

**Operating System:** Windows 10

**Programming Languages**: C/C++

**IDE(Integrated Development Environment**: Visual Studio 2017 community edition.

**External Libraries**: libtiff ver-4.1.0(compiled from source) for windows, libjpeg binaries(libjpeg.lib, jpeg62.dll) necessary for compressing TIFF files using JPEG.

**Directory structure:**

The whole project (source code and dependencies(external libraries)) are arranged in four different directories.

* **TIFFProcessor**: This folder contains the actual TIFFProcessor source code, visual studio project and solution files.
* **libtiff**: This folder contains the source, .lib and .dll files necessary for TIFF file processing. The libtiff.lib and libtiff.dll are compiled from source. Below are the links for libtiff source code.
  + <http://download.osgeo.org/libtiff/>
* **libjpeg**: This folder contains the header files, static and dynamic libraries for JPEG compression support for TIFF files. The libjpeg can be download from the following link:

<http://gnuwin32.sourceforge.net/packages/jpeg.htm>

* **Test TIFF Files**: This folder contains the TIFF files to test the TIFFProcessor utility.
* **README**: This folder contains “this” document (TIFFProcessor README.docx) and other jpeg images which explains the TIFFProcessor project creating and building in Visual Studio.

Directory sturcture looks like this:



**How to create and compile the project:**

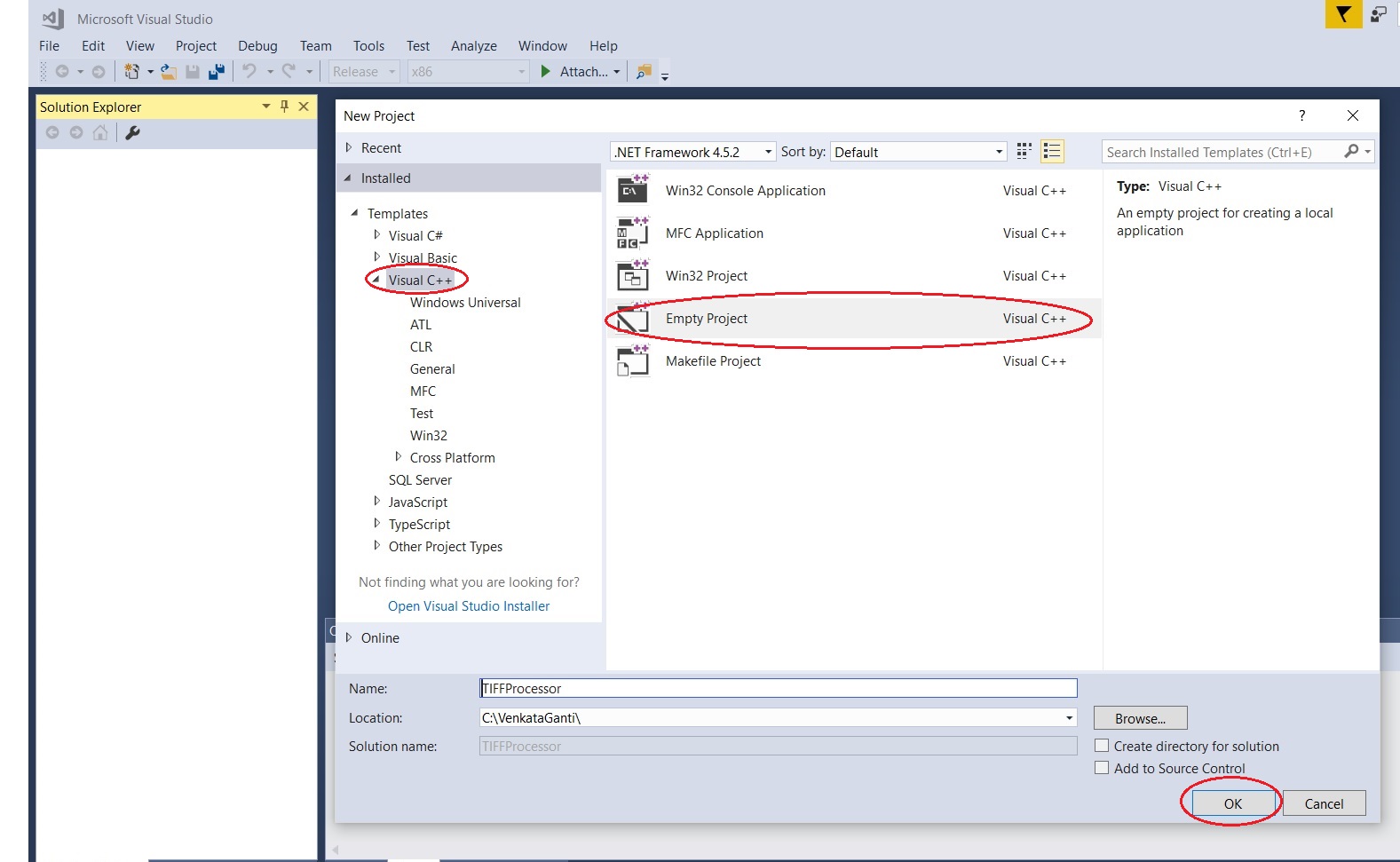
There are three source code files provided in the “TIFFProcessor” folder. They are:

1. Source.cpp
2. TIFFProvider.h
3. TIFFProvider.cpp

This is a very straight forward process which involves few steps (shown below in this document). On Windows, using Visual Studio( preferably any version after 2012), developer can create a project and add the above source files to the project, add few dependencies to the project settings and compile. Detailed steps are as follows:

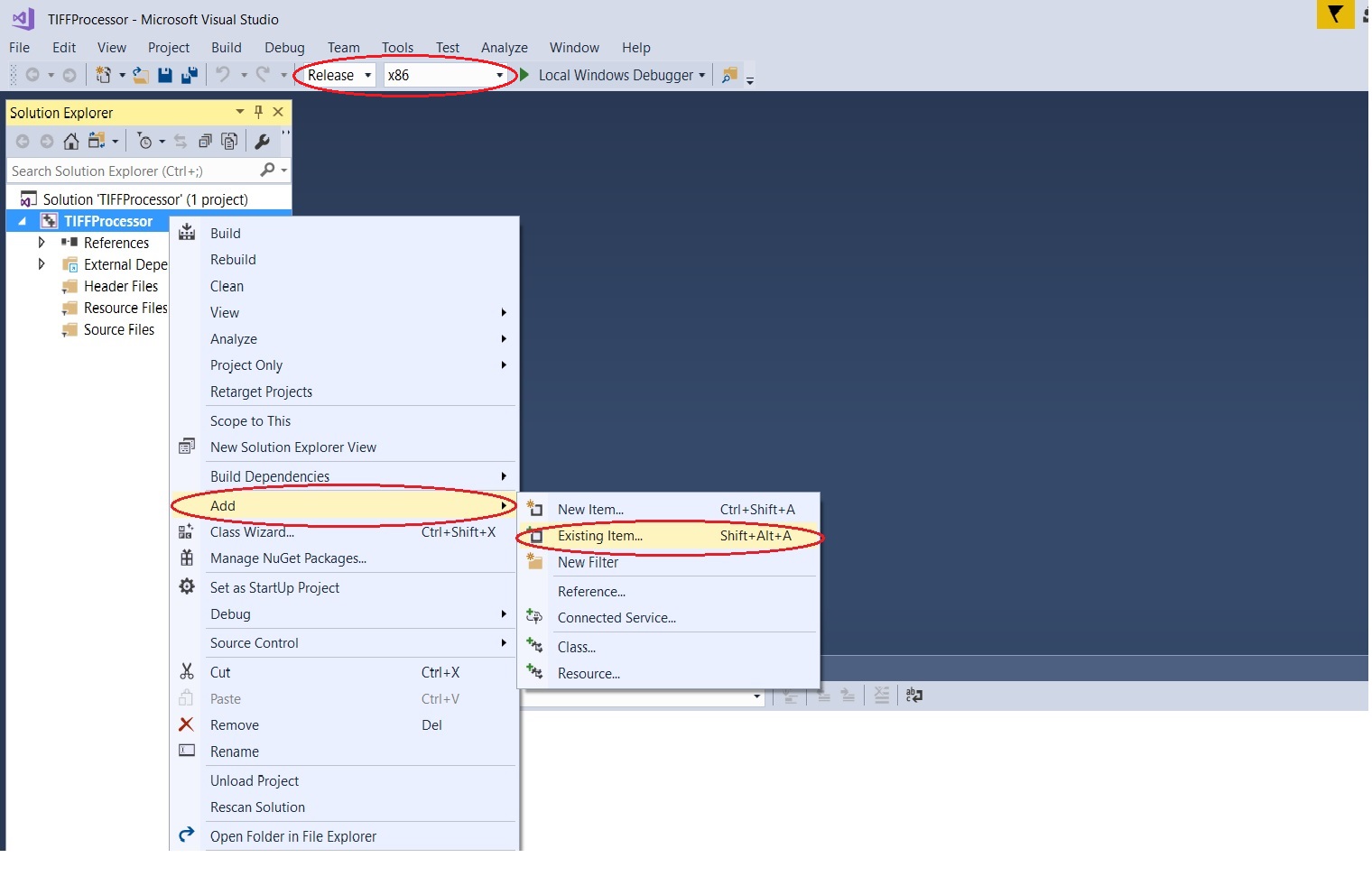
**STEP 1: Creating a project in Visual Studio**

* Open Visual Studio and select “File->New Project”.
* Select “Installed->Visual C++” on left pane and choose “Empty Project” from the right pane. Choose “TIFFProcessor” directory (which contains the above mentioned three source files) and click OK. Leave other settings to default values.

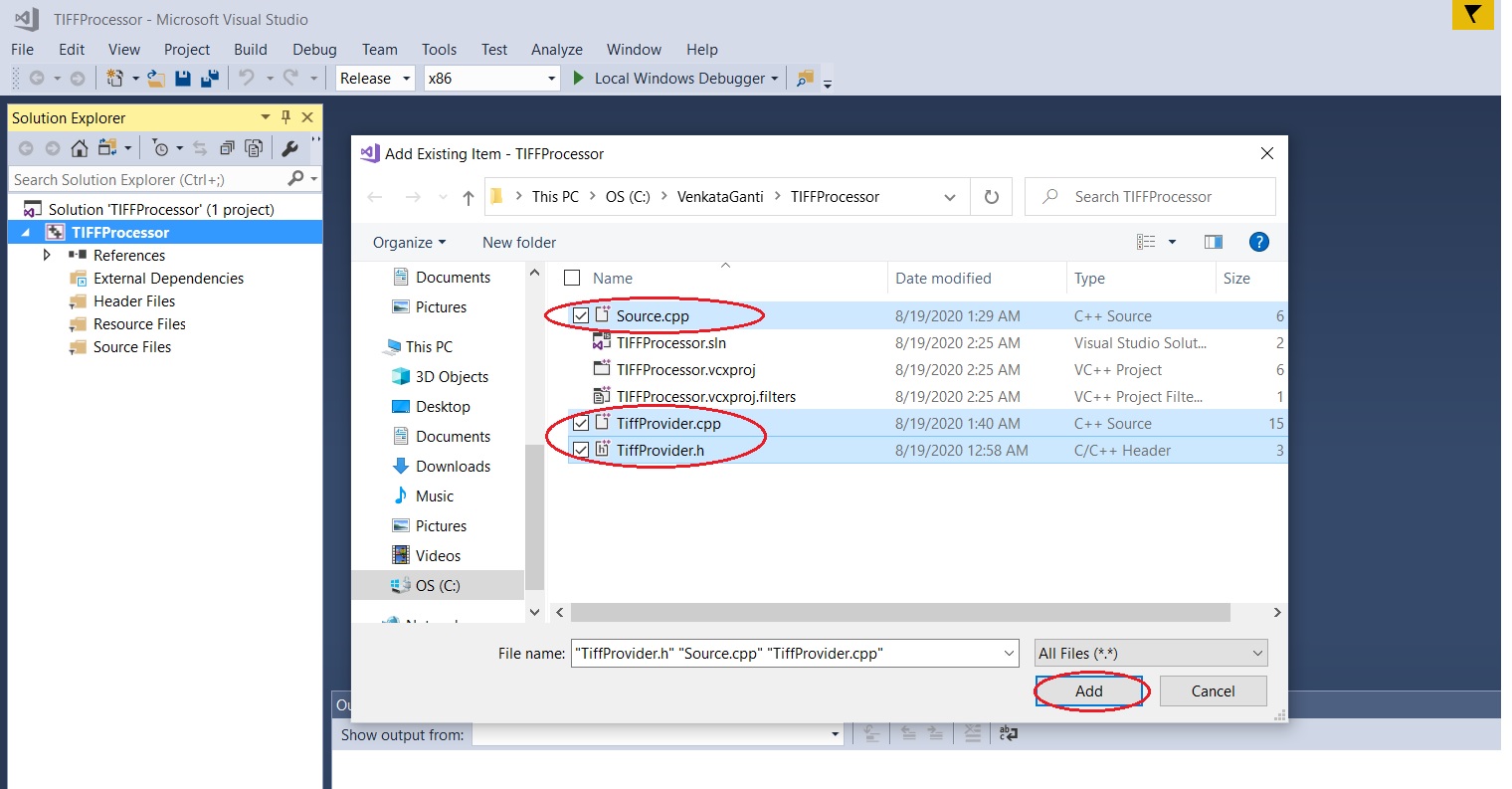


**Step 2: Adding three source files to the project**

Right click on the project “TIFFProcessor” and select “Add->Exisiting Item” option. Make sure you select the “Release” “x86” option from the top of the window, shown in the below image.

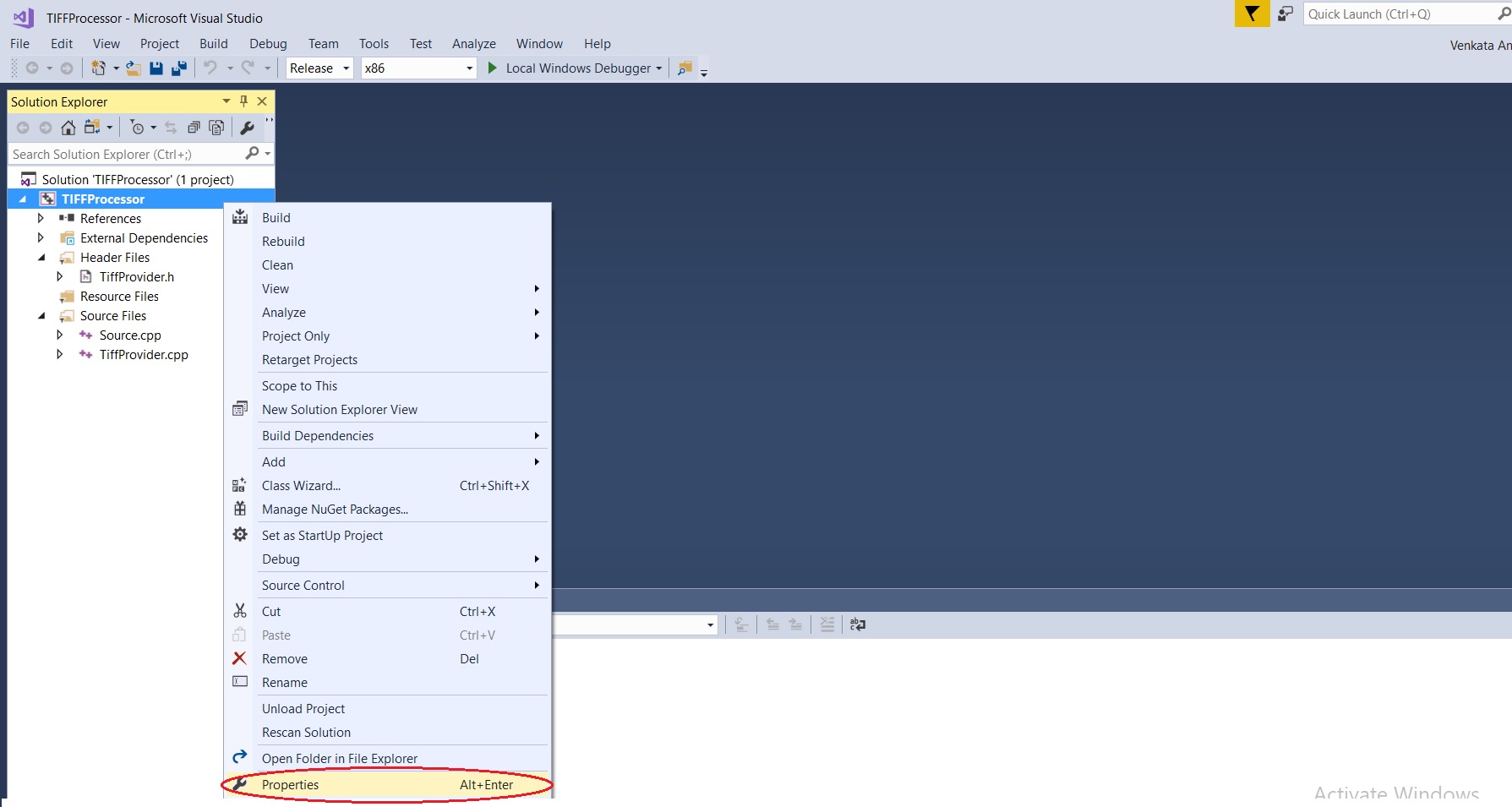


Select the three source files from the “TIFFProcessor folder” and click “Add” button.



**Step 3: Adding file paths and dependency libraries**

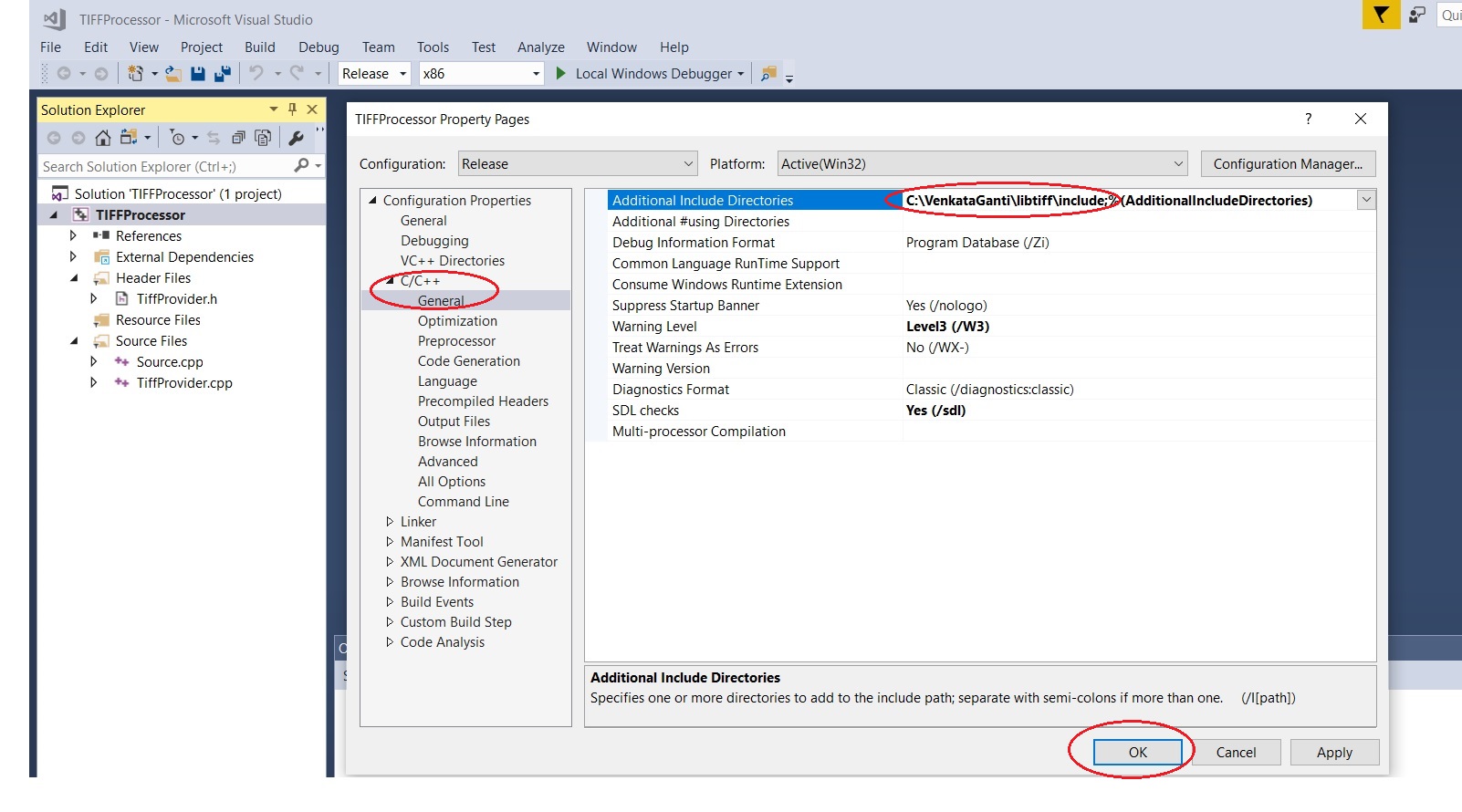
Right click on the “TIFFProcessor” project and select “Properties” option from the popup menu.



A window with name “TIFFProcessor Property Pages” will appear.

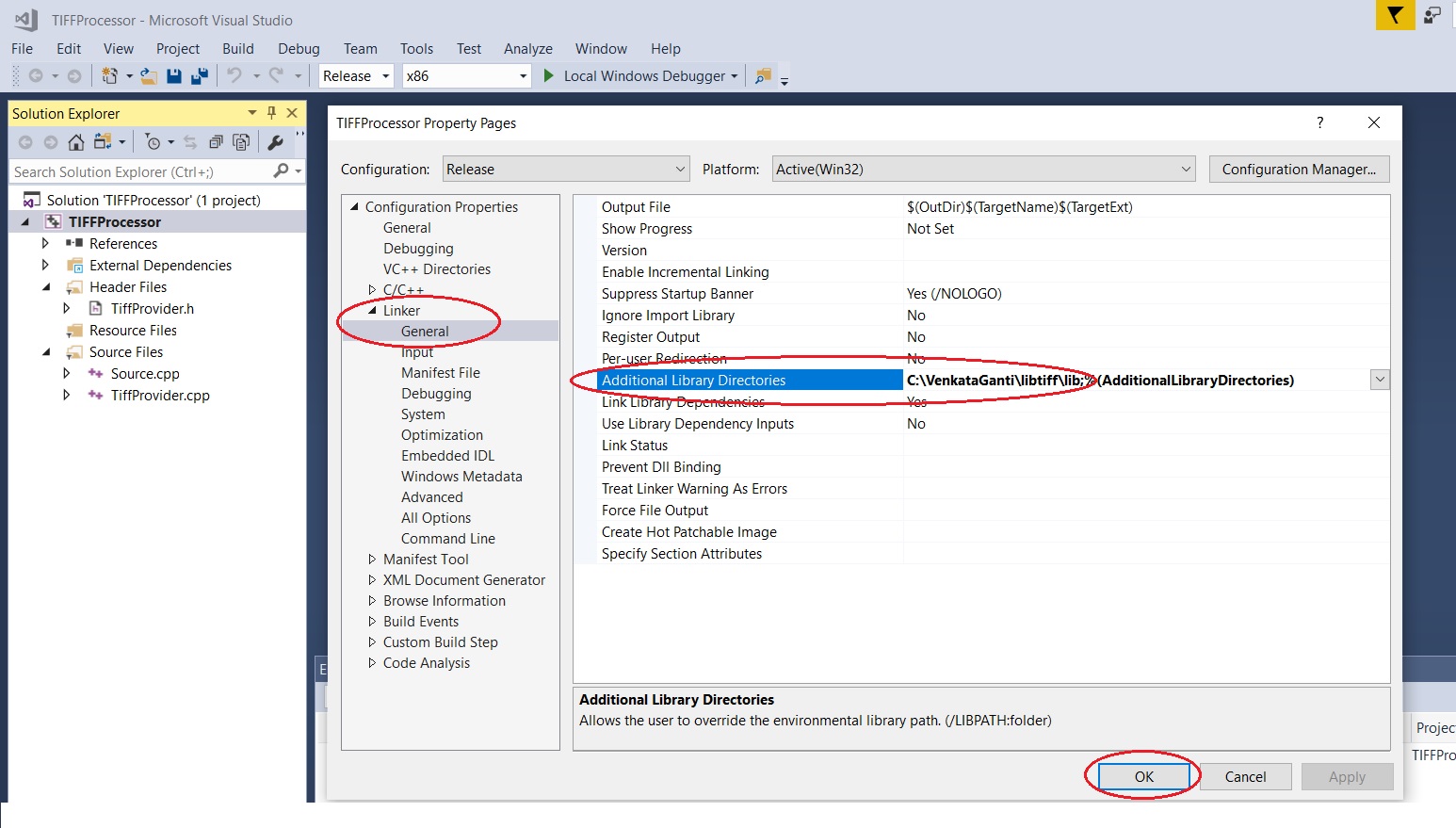
On the left side of the window select “Configuration Properties->C/C++->General” option.

In the “Additional Include Directories” text box, enter the path for “libtiff include” files. “libtiff” include files followed by a “;” (semicolon) to separate the path from other paths in the same text box, as shown in the picture. Or alternatively you can add the relative path. Ex: ..\libtiff\include.



On the left side of the window select “Configuration Properties->Linker->General” option.

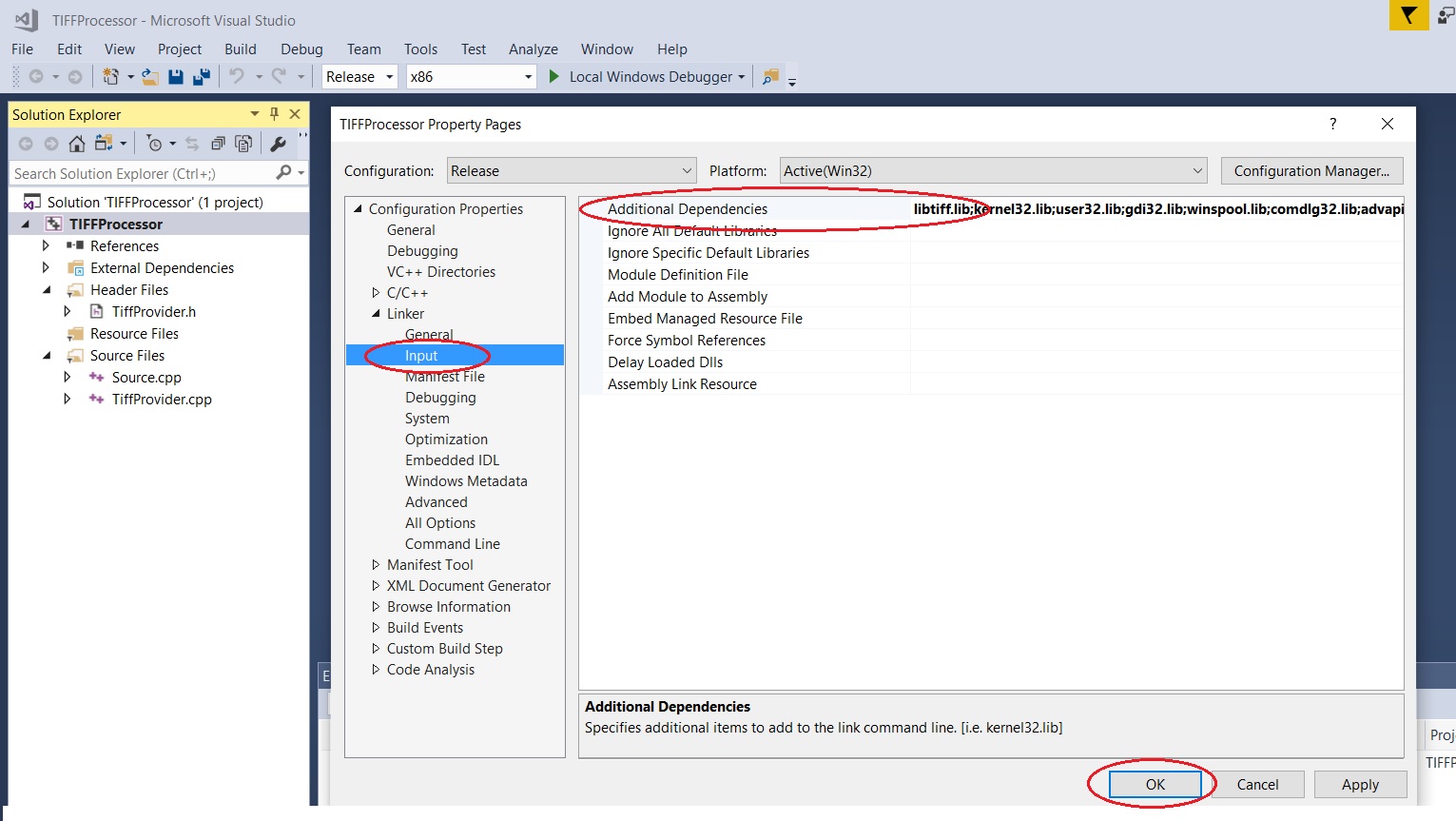
In the “Additional Dependencies” text box, enter the path for the “libtiff” include files, followed by a “;”(semicolon) to separate the path from other paths in the same text box, as shown in the picture. Or alternatively you can add the relative path. Ex: “..\libtiff\include’.



‘

On the left side of the window select “Configuration Properties->Linker>Input” option.

In the “Additional Dependencies” text box, enter the path for the “libtiff” static library file(libtiff.lib) followed by a “;”(semicolon) to separate the path from other paths in the same text box, as shown in the picture. Or alternatively you can add the relative path. Ex: “..\libtiff\lib.



With the above steps the TIFFProcessor project is created and ready to build.

Right click on the TIFFProcessor solution or the project and select build. The project should build without any errors, and creates a “Release” folder in “TIFFProcessor” solution directory.

The solution will create “TIFFProcessor.exe” in the “TIFFProcessor\Release” folder.

Now copy the “tiff.dll” from “libtiff\lib” folder and “jpeg62.dll” from “libjpeg\lib” folder and paste them in “TIFFProcessor\Release” folder.

If you don’t copy and paste the “tiff.dll” and “jpeg62.dll” (dependencies for “libtiff.lib” and “jpeg.lib”), the TIFFProcessor application will crash while trying to execute with the following errors.

