Manuals+ — User Manuals Simplified.

Tektronix 46W-74051-0 Test Automation User Guide

Tektronix®

**Getting Started
with C++ for Test Automation
GETTING STARTED GUIDE**

Contents

What is a Programmatic Interface?

A programmatic interface (PI) is a boundary or set of boundaries between two computing systems that can be programmed to execute specific behaviors. For our purposes, it is the bridge between the computer that runs on every piece of Tektronix test quipment and the application written by the end user. To narrow this even further, it is a set of commands that can be sent remotely to an oscilloscope and the system on that oscilloscope that processes and executes them. The PI Stack (Figure 1) shows the flow of information from the host controller down to the instrument. The application code written by the end user defines the behavior of the target instrument. This is usually written in one of the development platforms popular in the industry, such as Python, MATLAB, LabVIEW, C++, or C#. This application will send SCPI (Standard Commands for Programmable Instrumentation) commands, which are the standard format for almost all test and measurement equipment, to the VISA layer. In some cases, the application will call a driver, which will then send one or more SCPI commands to the VISA layer.
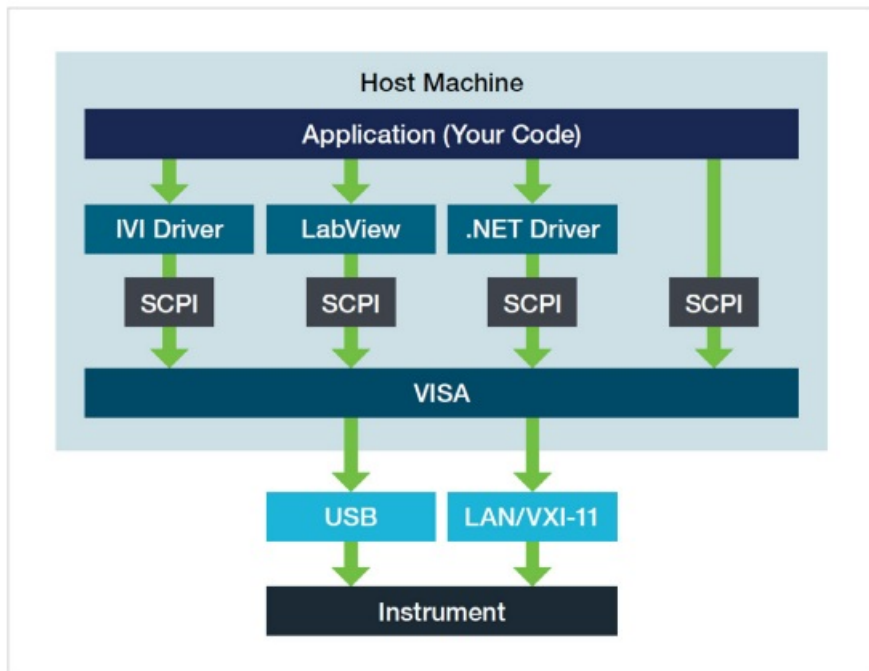
Figure 1: The programmatic interface (PI) stack shows the flow of information between a host controller and instrument.

Installation and Prerequisites Overview

This guide will provide you with a source code file to build a C/C++ project with to connect to your oscilloscope to save a screenshot to its local storage and send a copy to your local computer.

The steps outlined in the guide will walk you through the basic installation of free tools, ending with the execution of the code to yield a screenshot specifically on our MSO 5-Series. This code is also compatible with the 2, 4, 5, 6-Series scopes.

A note on the script: We have included code that should be compatible with C as lower common denominator, surely you could do this easier with more modern C++, but we wanted to demonstrate with basic C where possible.

Please download or copy our script from the Tektronix GitHub: **Hard Copy Example**

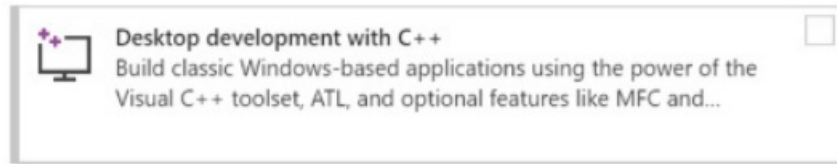We will go through the following:

1. IDE (Integrated Development Environment) Installation – Visual Studio Community Edition – Free

    a. Note: This guide will be using the VS compiler, not a true C compiler like mingw

2. VISA Installation

    a. NI-VISA

3. Starting a project with Visual Studio

4. Linking VISA into the project

5. Running the example code

Visual Studio Community Edition
Install Visual Studio Community Edition

1. Download & Install **Visual Studio Community Edition**

   a. Make sure you install the C/C++ Compiler from the Workloads tab.

Desktop development with C++
Build classic Windows-based applications using the power of the
Visual C++ toolset, ATL, and optional features like MFC and...

VISA

Install a VISA Client

1. Download and install NI-VISA

    a. Follow the steps outlined in our "**VISA Getting Started Guide | Tektronix**" guide for downloading and installing NI-VISA.
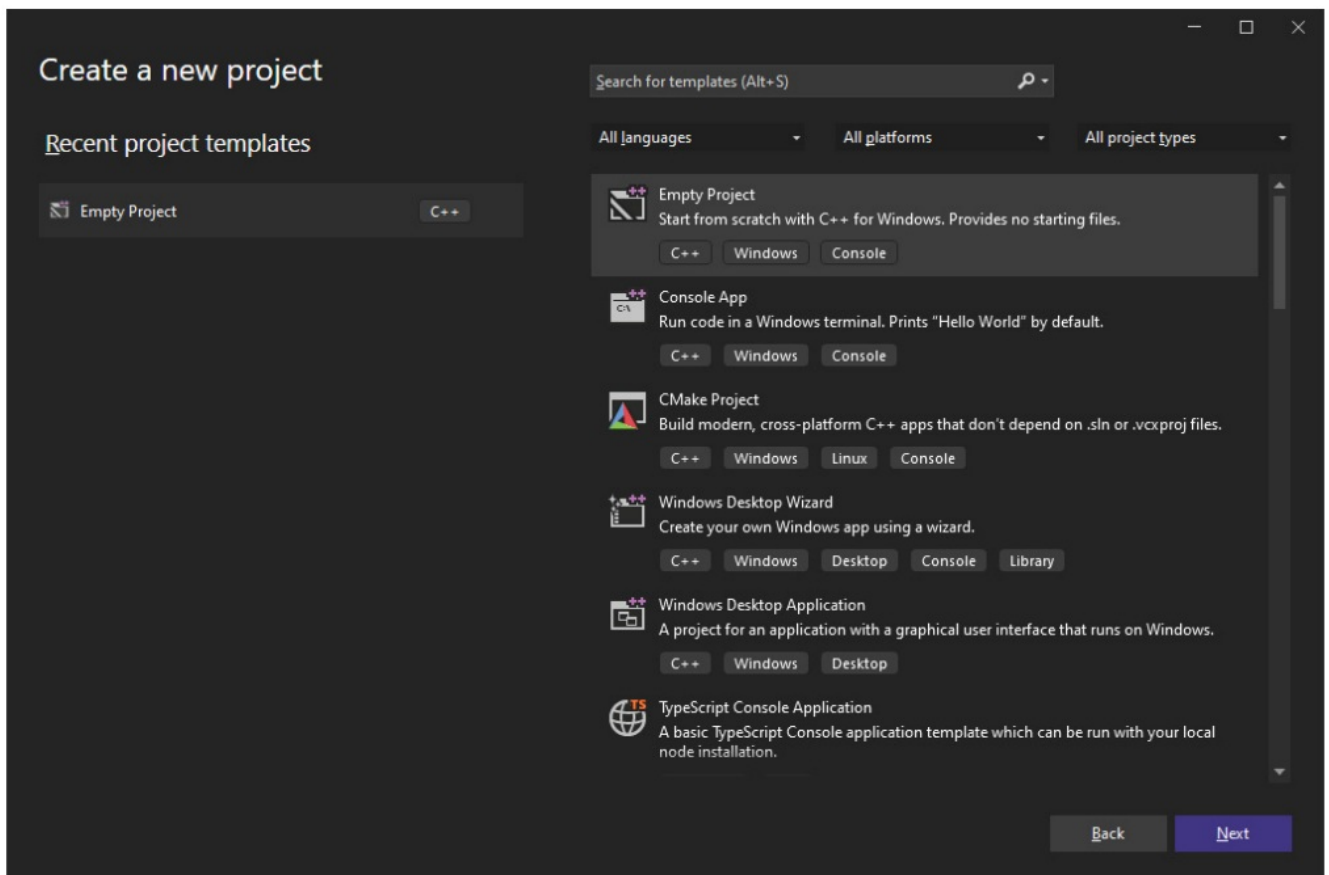
Example Code: Saving a Screenshot
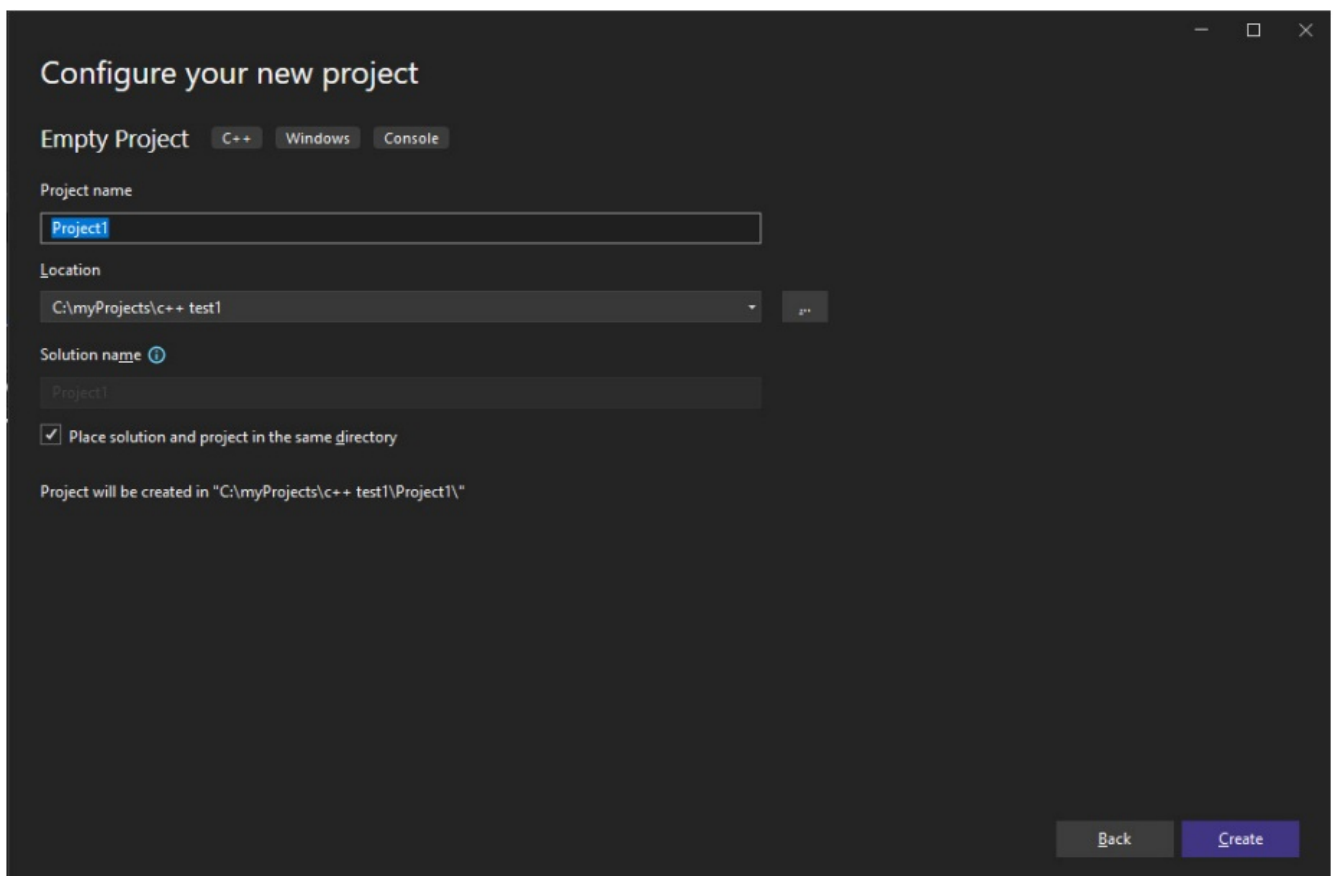
Setting up the project

1. Open visual studio community edition and start a new project.
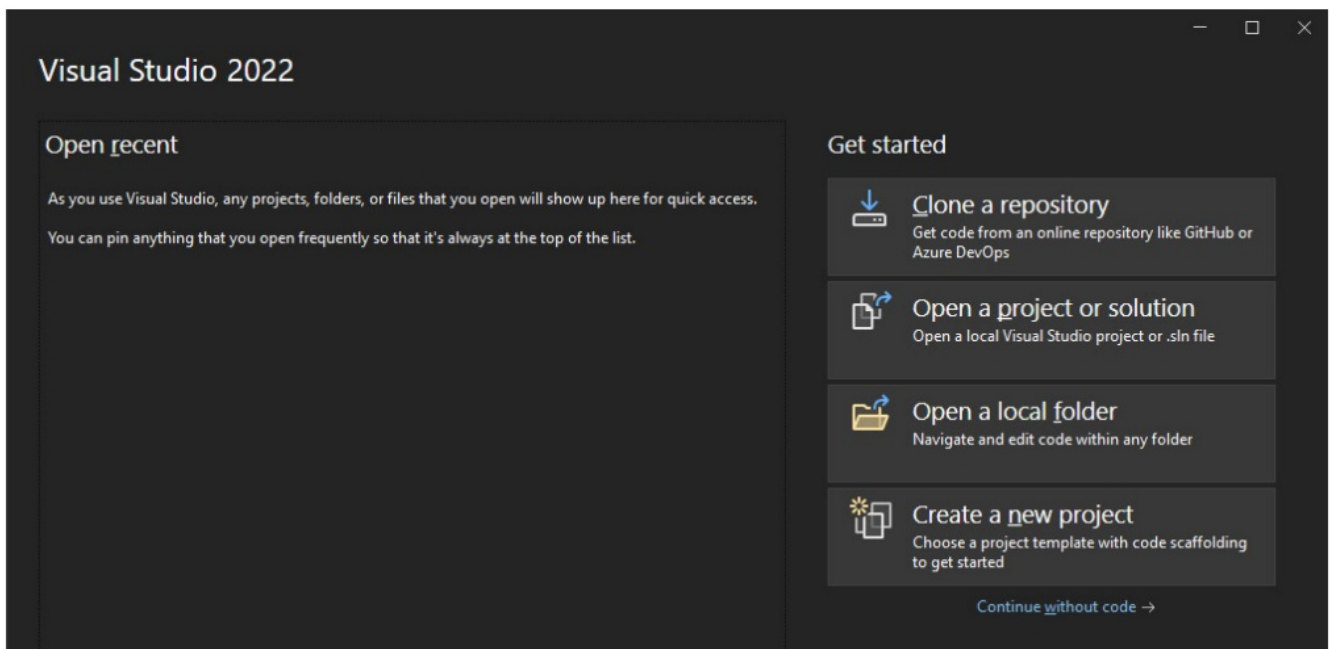

2. Select "Empty Project."

And in the next screen, enter your project name and folder.

3. If you do not see the solutions explorer on the right, you can display it with CTRL+ALT+L.

4. Download and import source.c you got from **the Tektronix GitHub** into your project by either 'right clicking on the project name > Add > Existing Item' and browse to the included source.c file. You can also use the keyboard shortcut SHIFT+ALT+A. This step will also create the 'Source Files' folder if it is not present by default.
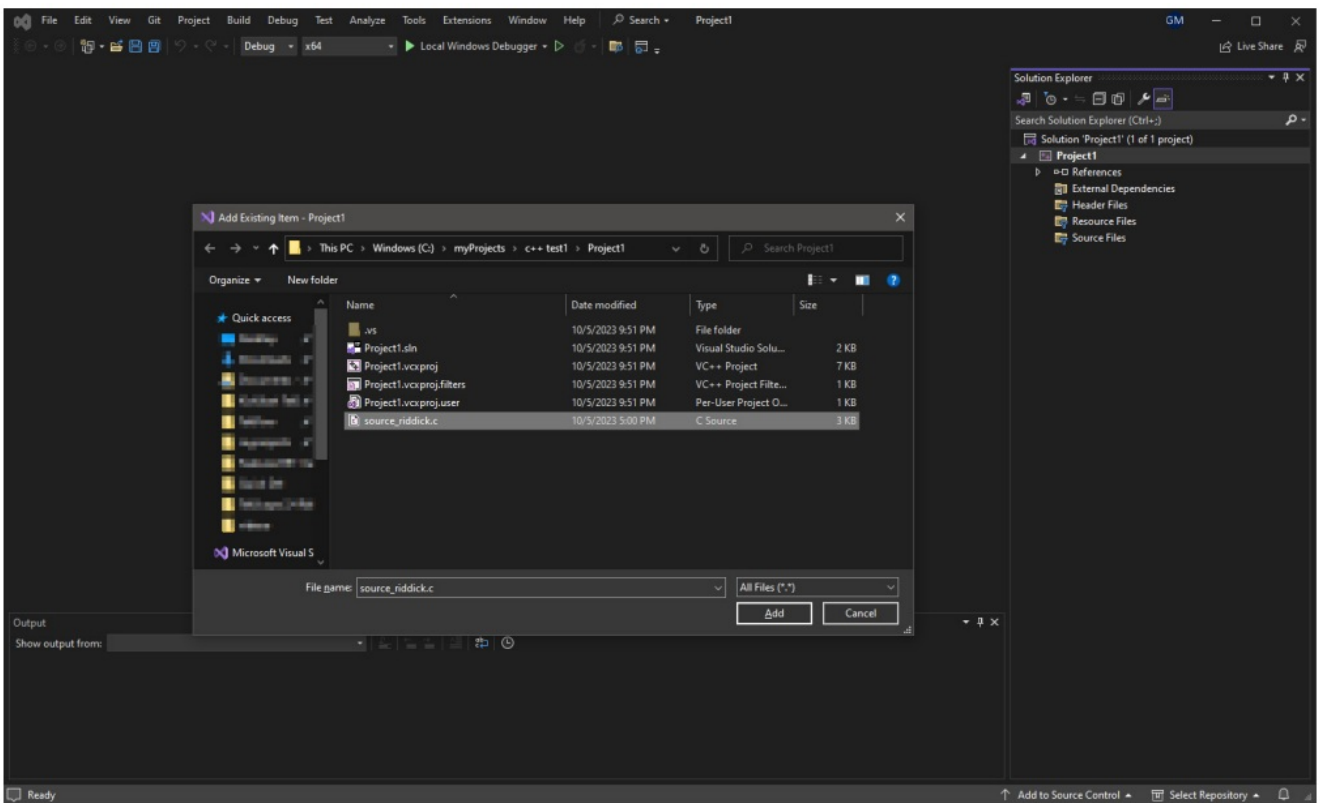
5. Open the source.c file. You should see many errors since the VISA libraries are not linked yet.
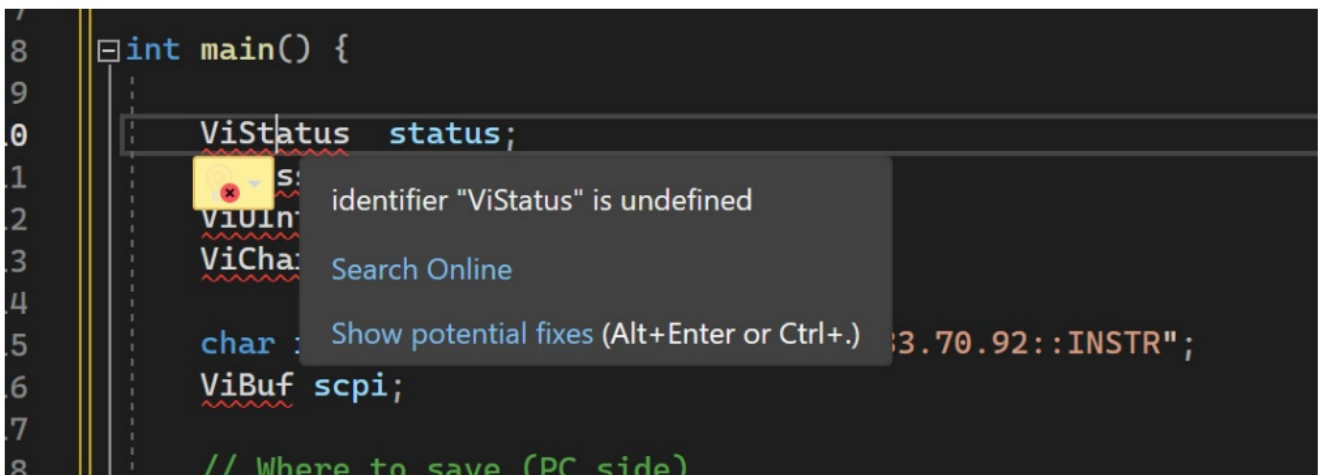

Linking VISA Libraries

You should see numerous errors like 'cannot find Source file,' 'identifier undefined,' etc.

1. Right click on the Project Name and navigate to properties.
2. In properties, navigate to Linker > Input > Additional Dependencies, drop down option> Edit and add the path to your NI Visa Library (**default: C:\Program Files\IVI Foundation\VISA\Win64\Lib_x64\msc\nivisa64.lib**)
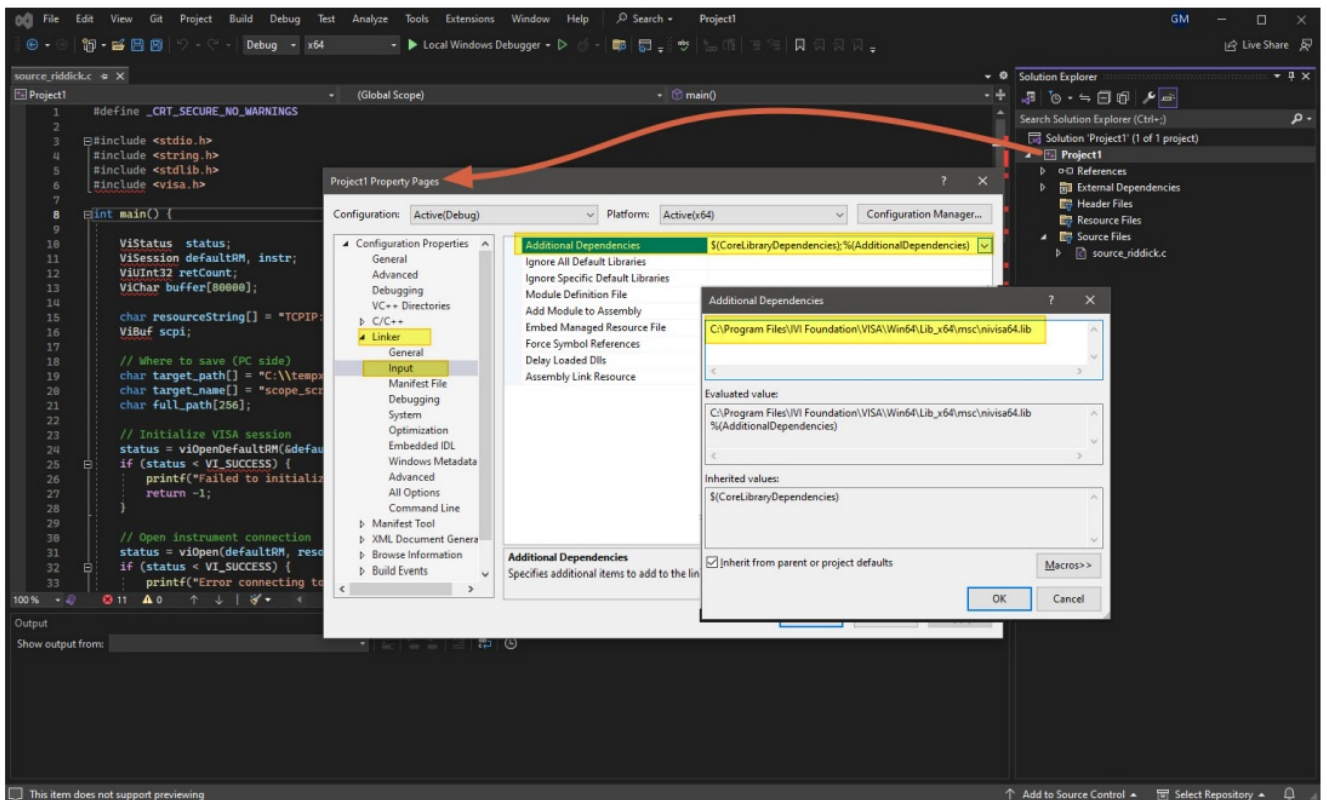
3. In the same project properties window, go to C/C++ > General > Additional Include Directories> Add a new line > Browse to your NI VISA headers folder (**default: C:\Program Files\IVI Foundation\VISA\Win64\Include**)

   If you do not see C/C++ tab, you need to make sure you have a .c file opened in the Visual Studio.
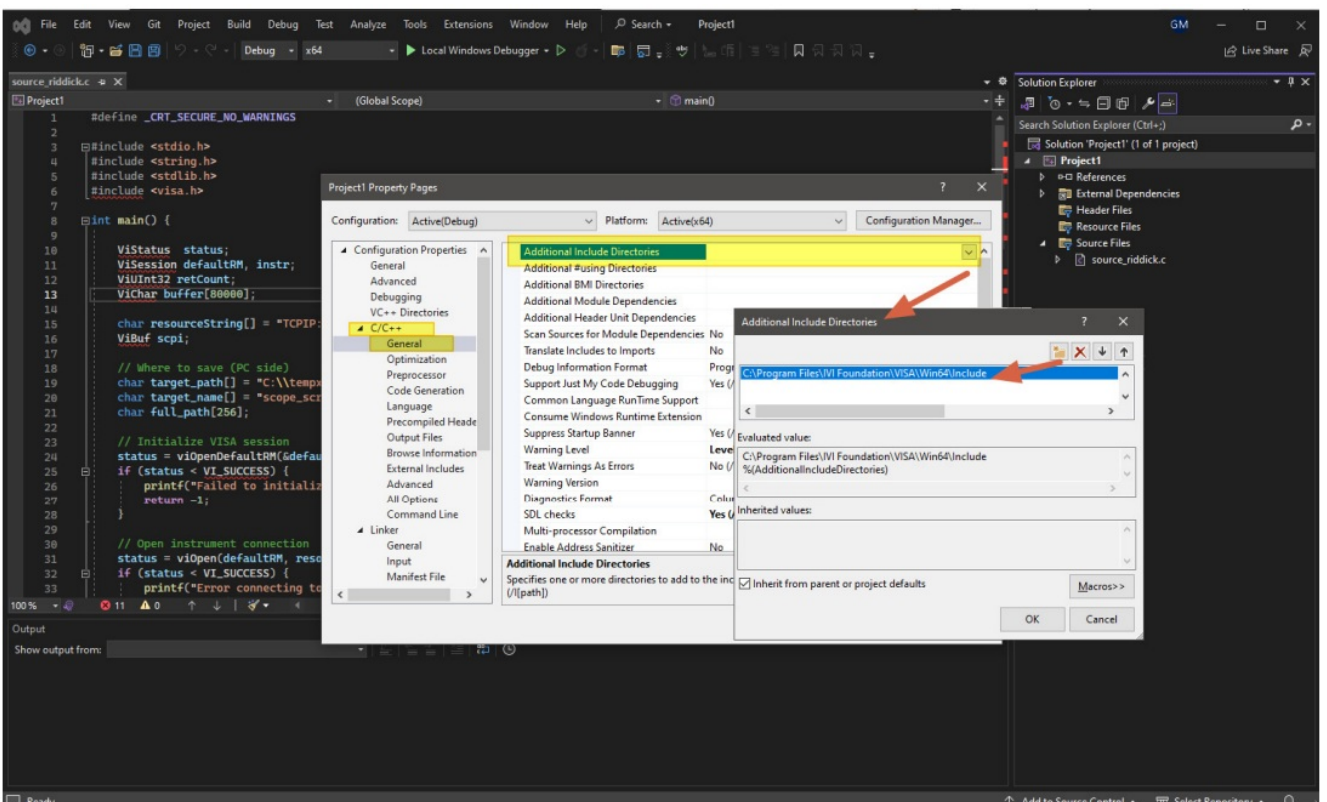
4. Hit Apply / Ok to exit out of the properties window.
5. The errors/syntax highlighting will be cleared now.
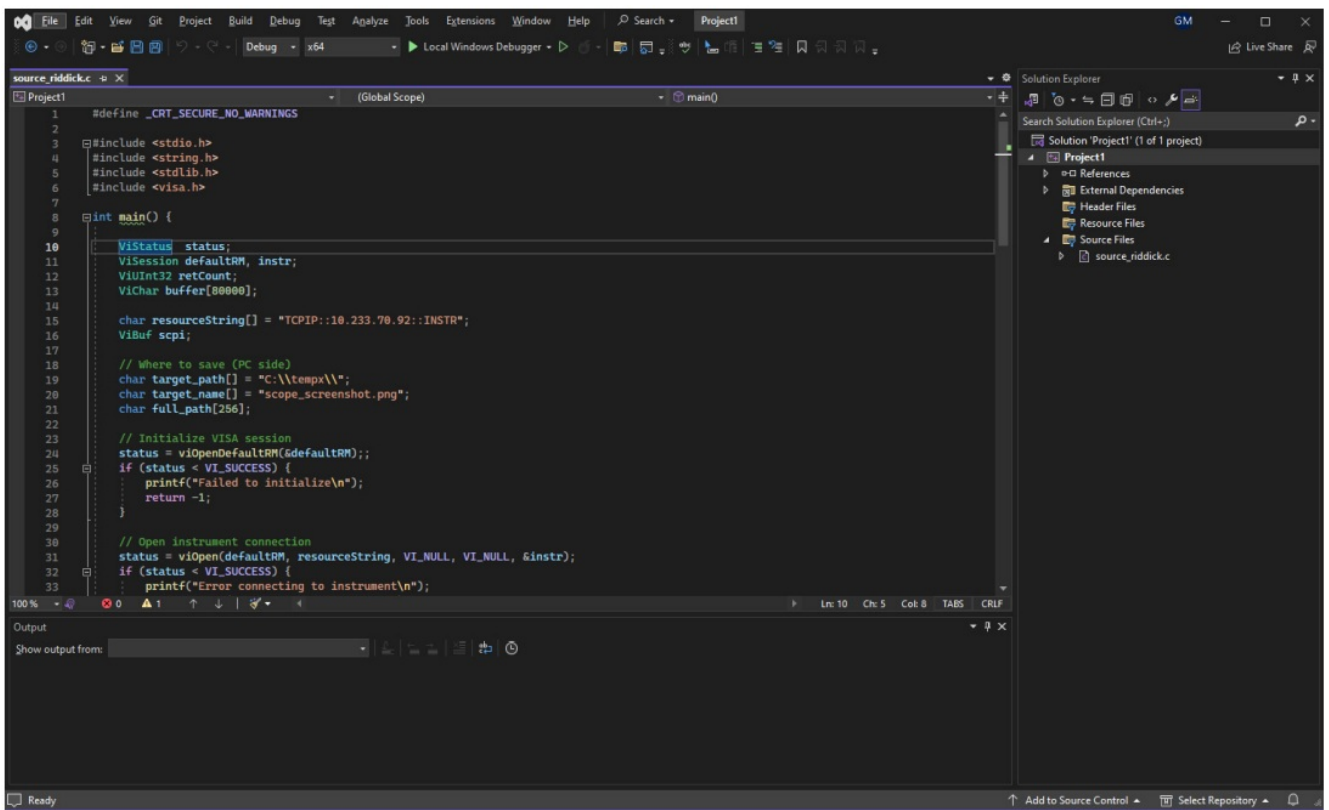
Running the Example

1. Edit the following lines according to your instrument.

   • Line 15: Resource String. Enter your instrument's IP address.

   • Line 19: Target path where the screenshot will be saved. Please ensure this path/folder exists on your local computer.

   • Line 20: This will be the filename of the saved screenshot on your local computer.

   • Line 60: is a direct hardcode to the scope, make sure this directory exists.

2. When ready, compile and run your code with CTRL + F5 (Or press the open green triangle).

3. If your code compiles correctly your code should run and end with the code 0 (0x0) message.
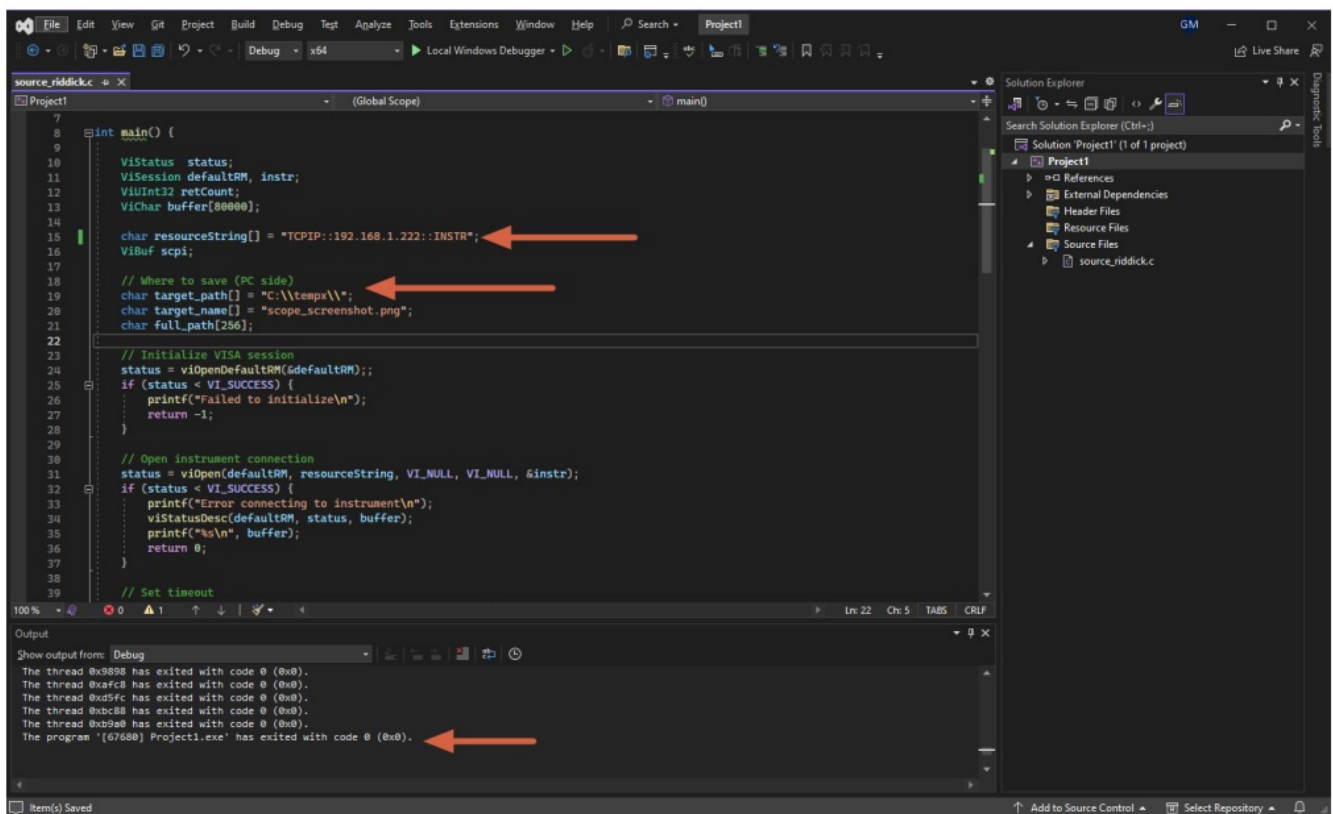
A note on repeat runs:

- You will need to delete the generated files on the scope and the local copy. Alternatively, you can build in some code to add the date and time to the filename to make them unique each run.
- You could also implement changes to increment the file names instead of hardcoding them in this quick example.

With this guide we have covered the basics for a small automation task with C/C++. The linking of VISA libraries is one of the most challenging pieces to getting C based languages working with automation.
If you found this useful, you might also want to check out our other language guides on
www.tek.com/testautomation.

Contact Information:
Australia 1 800 709 465
Austria* 00800 2255 4835
Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium* 00800 2255 4835
Brazil +55 (11) 3530-8901
Canada 1 800 833 9200
Central East Europe / Baltics +41 52 675 3777
Central Europe / Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France* 00800 2255 4835
Germany* 00800 2255 4835
Hong Kong 400 820 5835
India 000 800 650 1835
Indonesia 007 803 601 5249
Italy 00800 2255 4835
Japan 81 (3) 6714 3086
Luxembourg +41 52 675 3777
Malaysia 1 800 22 55835

Mexico, Central/South America and Caribbean 52 (55) 88 69 35 25
Middle East, Asia, and North Africa +41 52 675 3777
The Netherlands* 00800 2255 4835
New Zealand 0800 800 238
Norway 800 16098
People's Republic of China 400 820 5835
Philippines 1 800 1601 0077
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea +82 2 565 1455
Russia / CIS +7 (495) 6647564
Singapore 800 6011 473
South Africa +41 52 675 3777
Spain* 00800 2255 4835
Sweden* 00800 2255 4835
Switzerland* 00800 2255 4835
Taiwan 886 (2) 2656 6688
Thailand 1 800 011 931
United Kingdom / Ireland* 00800 2255 4835
USA 1 800 833 9200
Vietnam 12060128
* European toll-free number. If not
accessible, call: +41 52 675 3777
Rev. 02.2022

Find more valuable resources at **TEK.COM**

Documents / Resources

Tektronix 46W-74051-0 Test Automation [pdf] User Guide
46W-74051-0 Test Automation, 46W-74051-0, Test Automation, Automation

References

- **Test and Measurement Equipment | Tektronix**

- Ⓣ [Test and Measurement Equipment | Tektronix](#)
- Ⓣ [Test Automation | Tektronix](#)
- ⌗ [Programmatic-Control-Examples/Examples/Oscilloscopes/MidrangeScopes/src/SaveHardcopyExample/save_hardcopy_c++.c at master · tektronix/Programmatic-Control-Examples · GitHub](#)
- [User Manual](#)