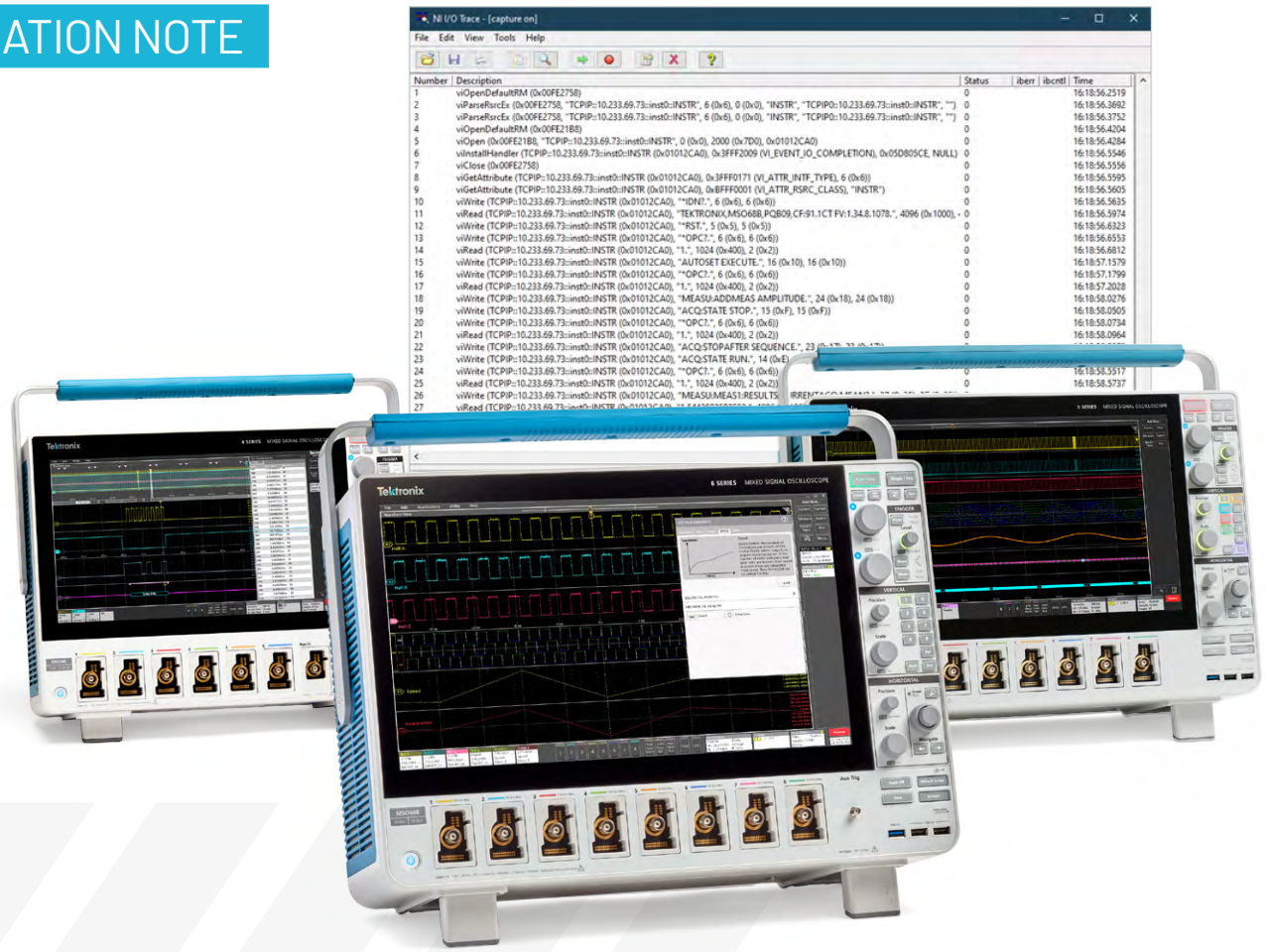
Tektronix®

Getting Started Controlling Instruments with VISA

APPLICATION NOTE



Introduction

Most modern test and measurement instrumentation today can be configured and controlled via a remote Programable Interface (PI). These remote PIs are accessed over physical interfaces such as Ethernet, USB, GPIB or Serial Port and many instruments will include two or more of these interfaces on them. Each of these physical interfaces is very different in how data is sent over them and the use of these interfaces in a remote instrument control application requires interacting with a different driver or application programmable interface (API) within the host operating system of the system controller. Even though each of these physical interfaces is unique and has its own way of communicating with it, the remote Programmable Interface on the instrument is the same, no matter which physical interface is in use.

For most instruments, interaction with the Remote Programmable Interface consists of sending text-based commands to the instrument and reading back the response. If you were to write a program to control an instrument via GPIB, you would write a program that uses a GPIB library to write these text commands via the GPIB interface. However, if a new model of your instrument came out that no longer included a GPIB interface, but had an Ethernet interface instead, you would now need to re-write your code to send those same text commands via Ethernet. How inconvenient and tedious!

This problem of having many physical interfaces that each require their own API to use, even though all those interfaces are being used to access the same remote Programmable Interface on an instrument, was recognized long ago and the Virtual Instrumentation Software Architecture (VISA) Library was created to address the issue. VISA is an instrument control library that allows you to write programs and scripts to control instruments without having to deal with the specific libraries of the physical interface being used. VISA allows you to send text-based commands strings to instruments and read-back the responses using a unified API that works with all of the common physical interfaces available on instrumentation. When an application is written using VISA, all that is needed to switch the physical interface being used is usually little more than to change the VISA resource address.

In this guide we will show you how to get started using VISA to interact with and control instrumentation.

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's the bridge between the computer that runs in every piece of Tektronix test equipment 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 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.

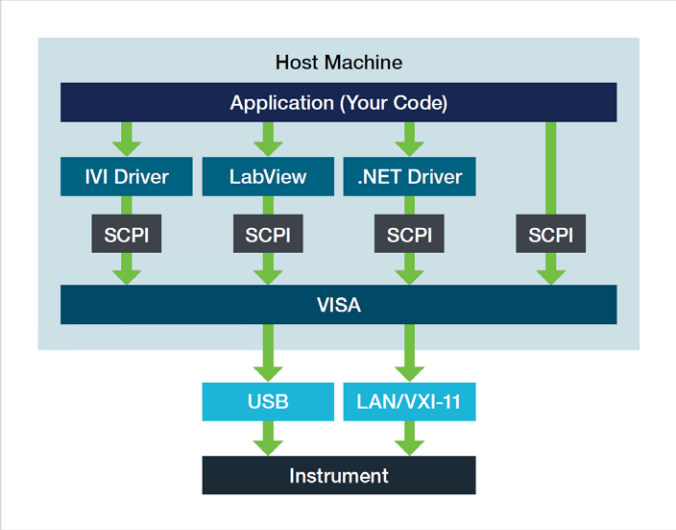


Figure 1: The programmatic interface (PI) stack shows the flow of information between a host controller and instrument

The VISA layer negotiates the connection with the instrument over the physical interface, such as USB or Ethernet. VISA opens and closes the remote session with the instrument, writes and reads data on the bus, and maintains the health of the connection with synchronization, flushing and other miscellaneous functions. This layer abstracts a lot of the technical minutia required for communication, like resource addresses and buffer management.

The instrument will then read and write information directly on the physical interface. Depending on what commands are issued, it may change settings in the scope application or provide the response to a query.

In all of the systems above the physical communication interface are also mirrored in the scope, but we can consider it a "black box" since we can't modify those systems. Each major instrument vendor provides an implementation of the VISA standard to use with their products. All of the basic functionality is included in the standard requirements, so any vendor's VISA should work well enough to control an instrument.

Installing VISA

Recommended System Requirements

The follow list contains the recommend system requirements for following along with this guide.

- Personal computer running Windows 10 or Windows 11
  - Core i5-2500 or newer processor
  - 8 GB of RAM or greater
  - > 3 GB of free disk space

Install VISA

Note: If you are planning to develop instrument control applications using a .NET language (C#, VB.NET) and the Visual Studio or Visual Studio Code development environment, it is recommended that you install the development environment before proceeding to install NI-VISA. The installer for NI-VISA will detect that Visual Studio is installed and will automatically make sure that the correct components are selected and installed for use in code development.

1. Download NI-VISA:
   You can find the NI-VISA installer by searching for "NI-VISA download" on the web. The link to the NI-VISA download page on ni.com should be the first item in the search results. Click on it to go to the download page and proceed to download the installer.

Note: In this guide we will be using NI-VISA 2023 Q2. Other versions of NI-VISA as early as version 17 will work but the setup process may vary from what is shown in this guide and a separate installation of the IVI Compliance Package may be required to gain support for the IVI VISA. NET application programming interface. NI-VISA 2023 Q2 contains all needed packages and will be the only file you need to download and install.

Note: When downloading and installing NI-VISA, if there is an option between a Full version and a Run-time version, be sure to get the Full version. The Full version has additional tools and libraries that are needed for code development.

2. Install NI-VISA:
   Run the installer you downloaded in the previous step to begin installing NI-VISA.
   If you downloaded the off-line installer for NI-VISA 2023 Q2, you should have downloaded an ISO image file. If you are installing NI-VISA on an up-to-date version Windows 10 or 11, then you should be able to open this file by right-clicking on it and selecting "Mount". This will "Mount" the ISO image file to a new Virtual DVD Drive.

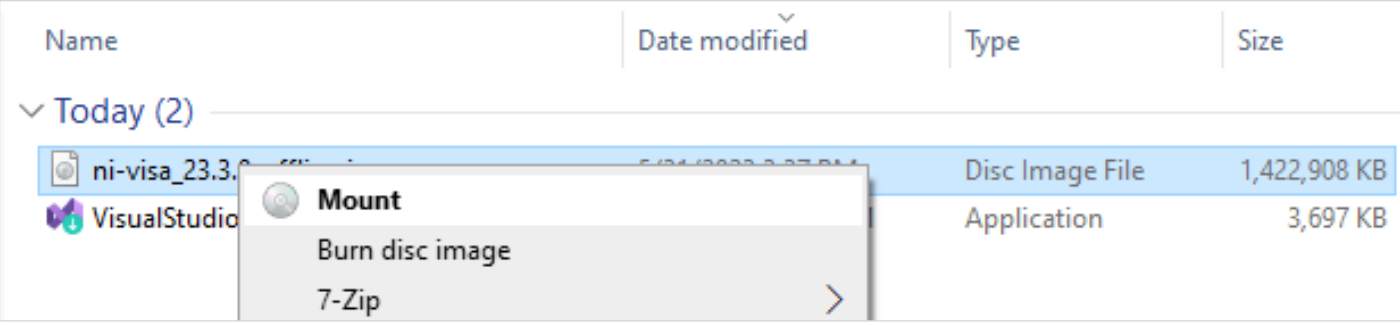


Figure 2: Mount the .iso file as a Virtual Disc Drive in Windows Explorer

Note: If you are using Windows 10 then the Mount option may not be available by default. To make this available in Windows 10 you will need to go to Default Apps in Windows Settings and change the app associated with .iso files to Windows Explorer.

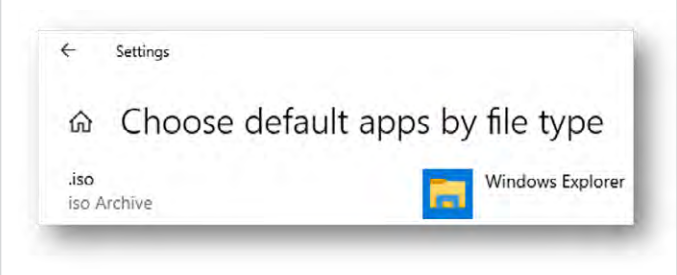


Figure 3: Associate .iso files with Windows Explorer to gain access to Mount option for .iso files in Windows Explorer

3. Browse into the Virtual DVD Drive in Windows Explorer and double-click “Install.exe” at the root of the drive to begin the installation process.

Note: Alternatively, or if mounting the ISO image file is not an option, you can use a third-party utility such as 7-Zip to extract the contents of the .iso file to the local hard disc, much like you would a .zip file. Once extracted, run “Install.exe” from the extracted files.

4. As you proceed through the NI-VISA installer, you will reach a screen where you will select “Additional items you may wish to install.”

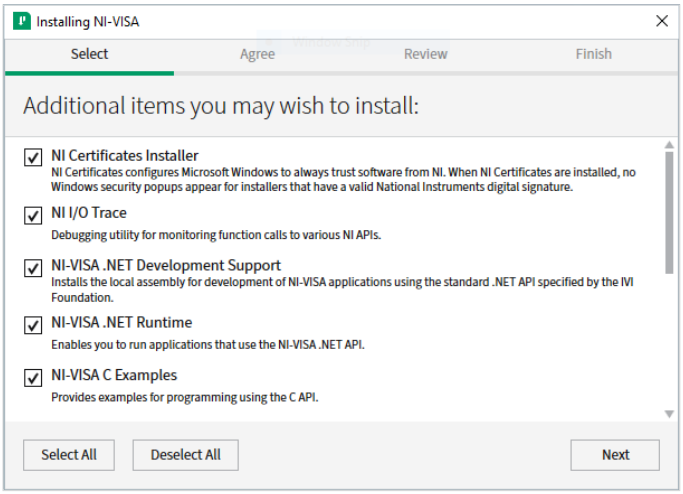


Figure 4: The development tools should be selected during the NI-VISA installation

The default options selected should include everything you need to perform development work. However, if you will be developing in a .NET language (C#, VB.NET) please ensure that at minimum the following options are also checked.

- NI-VISA .NET Development Support
- NI-VISA .NET Runtime

Click Next and proceed through the NI-VISA installer until it has completed. At the end of the installation, you will be prompted to reboot your computer. Click Reboot Now.

Identifying the Instrument's VISA Resource Address

To connect to an instrument using the VISA communications library, you will need to know the VISA Resource Address of the instrument you want to communicate with. The VISA Resource Address is a text string identifier that tells VISA the location of the instrument. The exact form of the VISA Resource Address will depend on the type of physical connection to the instrument, whether it is Ethernet, USB, GPIB, Serial, etc.

Ethernet Instruments

Many instruments today come with Ethernet as an interface for remote control. With Ethernet, the VISA resource address will usually be "TCPIP::<ip address>::inst0::INSTR" where <ip address> is the IP Address of the instrument's Ethernet port. Usually, an instrument will have its IP address displayed either right on its front panel or in a menu accessible from the front panel. For example, on Tektronix 2/3/4/5/6 Series Oscilloscopes, you can find the IP Address by going to the Utility menu, clicking I/O… from the menu and the IP address will be displayed on the LAN page.

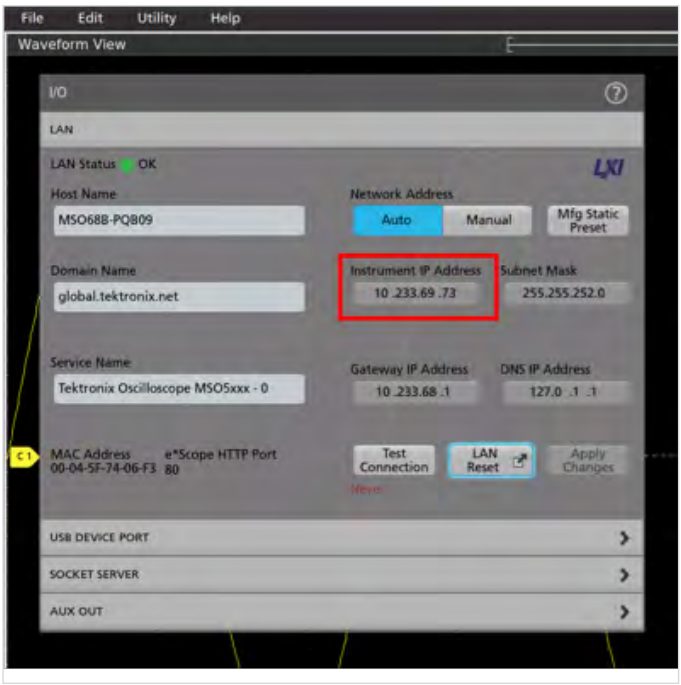


Figure 5: The instrument's IP address can be found in the I/O menu on 2/3/4/5/6 Series Oscilloscopes

For instruments running Windows, such as Tektronix MSO/DPO70000DX and DPO70000SX Series Performance oscilloscopes, you can find the IP address of the instrument in any of the following ways:

1. In Windows Control Panel go to *Control Panel\Network and Internet\Network Connections* and find the network adapter. Right-click on the adapter and click Status. In the network adapter's status window click Details… to open the Network Connection Details window.

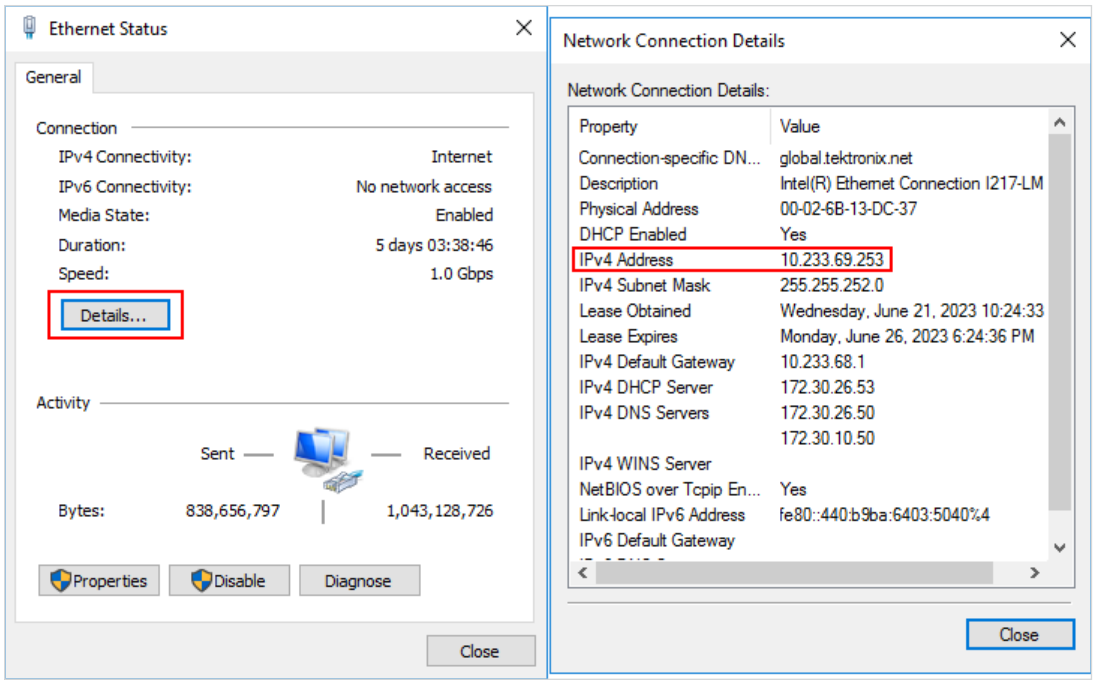


Figure 6: In the Network Adapter Status window click Details... to open the Network Connection Details window and find the IP address.

2. Open a CMD window and run *ipconfig* at the command line.

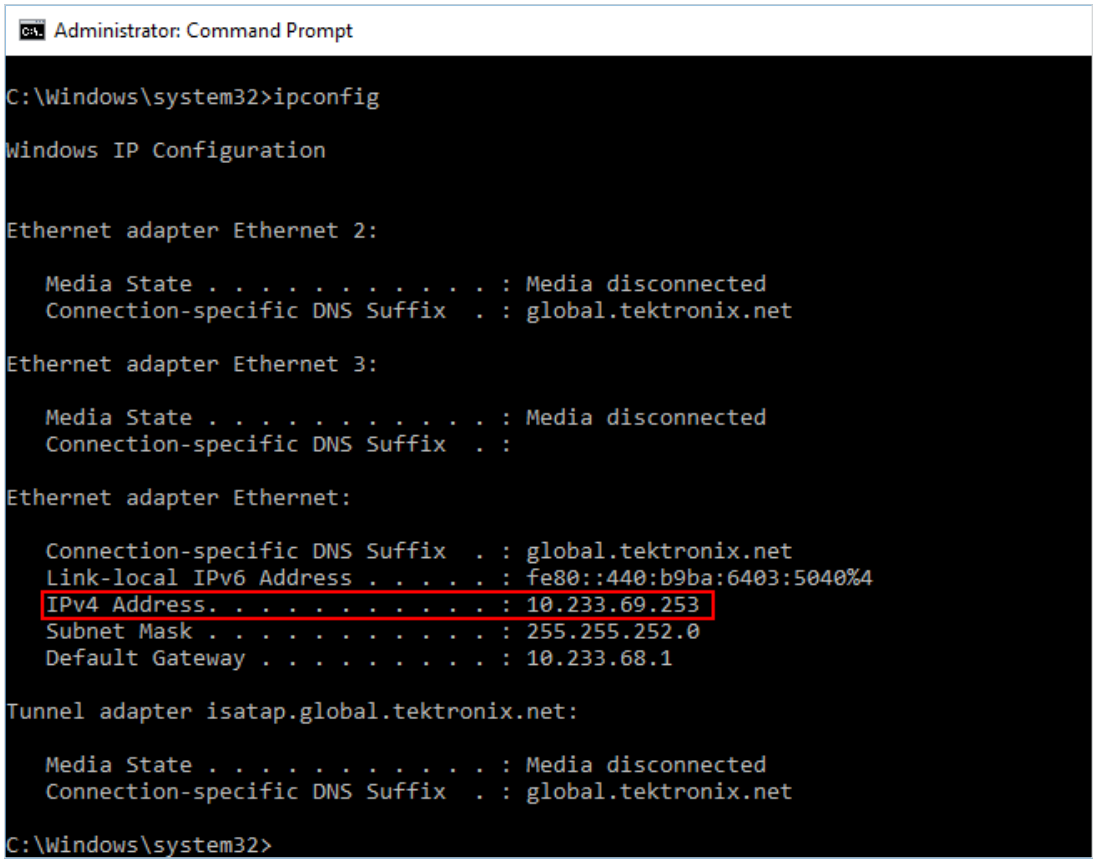


Figure 7: Run *ipconfig* in a CMD window on the instrument to find the IP Address.

3. Hover your mouse over the TekVISA LAN Server Control icon in the tray. The IP Address will appear in the tool-tip box.

4. Open a web browser on the oscilloscope and go to http://localhost
   The IP address as well as the VISA Resource Address for the instrument can be found on the main LXI webpage of the instrument.

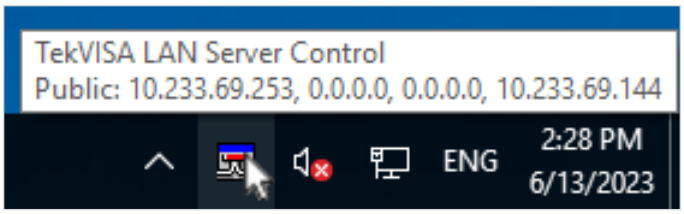


Figure 8: For oscilloscopes running Windows, hovering your mouse over the TekVISA LAN Server Control icon in the tray will show the instrument's IP address.

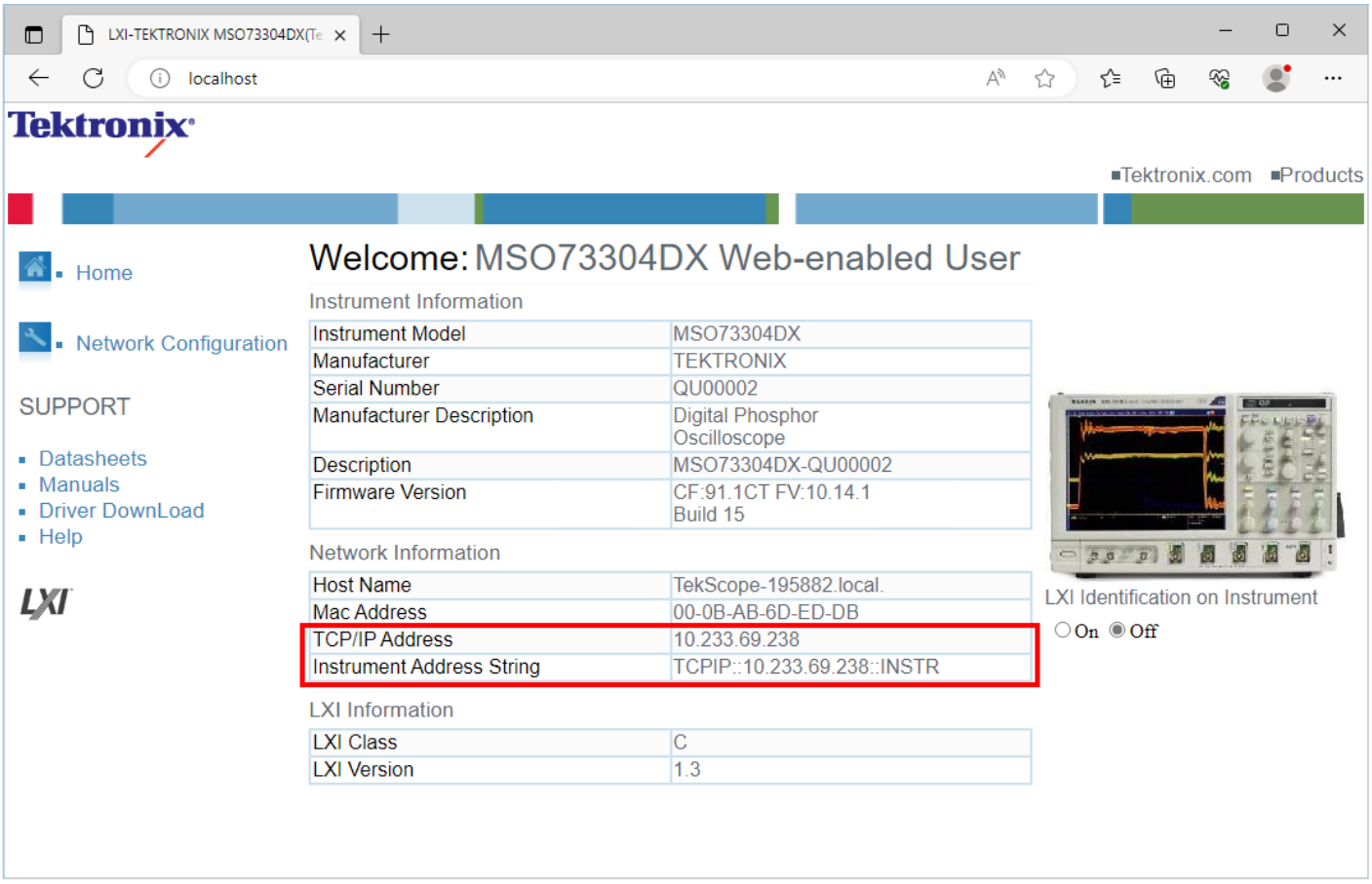


Figure 9: On oscilloscopes running Windows, the IP address and VISA Resource Name can be found quickly by opening a web browser on the oscilloscope and visiting http://localhost.

GPIB Instruments

If you are connected to your instrument via GPIB, the VISA Resource Address will be in the form of “GPIB::<GPIB address>::INSTR”, where <GPIB Address> is the address number assigned to your instrument. Typically, you will be able to find and set this address from a menu on the front panel of your instrument and it will be a number between 1 and 30. The address can usually also be auto detected by instrument management software running on the computer containing the GPIB interface adapter that the instrument is connected.

Figure 10: The GPIB address of the instrument can usually be found on the instrument’s front panel or in a menu. On Tektronix MSO/DPO70000DX Performance Oscilloscopes you can find the GPIB address in the Utilities menu under GPIB Configuration...

USB Instruments

If your instrument is connected via USB, the VISA Resource Address will be a long string that includes things like the Vendor ID and Device ID in it and it will be difficult to identify the VISA Resource Address manually. Finding the VISA Resource Address of USB devices is best done using instrument manager software such as NI-MAX.

Finding the VISA Resource Address in NI-MAX

With any VISA implementation, the vendor will include some kind of instrument management utility to identify instruments, assign aliases to them and/or manually add them to the local configuration. The installation for NI-VISA includes a utility called NI-MAX that is used to perform these actions. In this section we will show you how to identify instruments using NI-MAX.

1. From the Windows Start Menu, launch NI-MAX.

2. In the navigation tree on the left, Under My System, expand the navigation tree under the item Devices and Interfaces. Any instruments connected via USB should automatically show up here. Similarly, you should see any auto detected Ethernet Instruments show up if you expand the tree under Network Devices. Any auto detected GPIB instrument should show up if you expand the tree under your GPIB adapter.

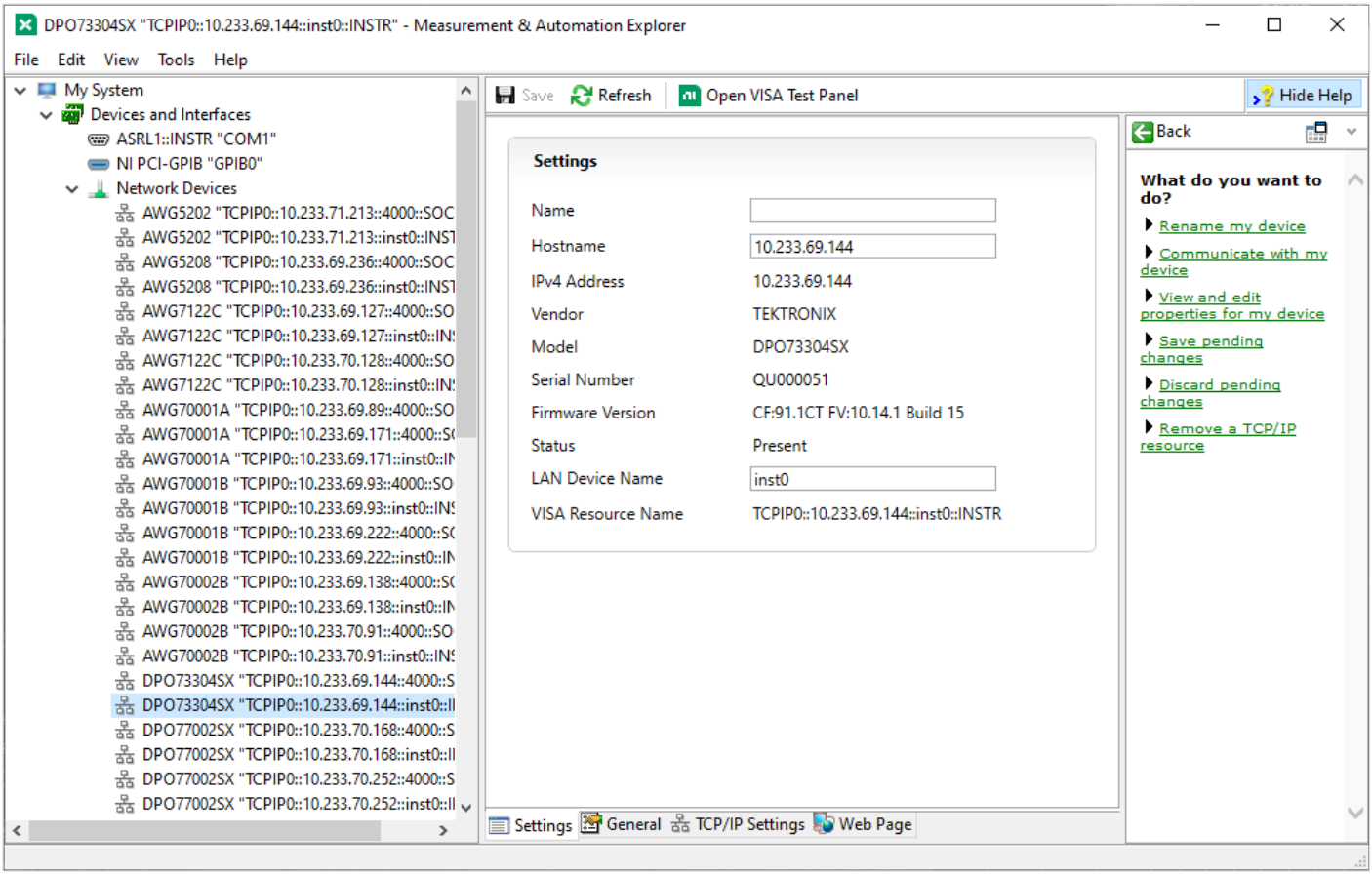


Figure 11: Instruments detected by VISA will show up automatically under Devices and Interfaces in NI-MAX. Click on a device in the tree to view additional details about the device.

Note: In some network configurations, instruments on the local network may not be detected and show up automatically under Network Devices. You can add manually add instruments to the local configuration by right-clicking on Network Devices and choosing Create New VISA TCP/IP Resource…

3. To find the VISA Resource Address of any instrument in the local configuration, left-click on the instrument and properties for the device will be listed in the panel to the right. At the bottom of the settings tab, you will see the VISA Resource Address listed next to the title VISA Resource Name. This is the address you can use to connect to the instrument using VISA. You can copy and paste this address for later use.

Testing Your VISA Connection

After finding or adding your instrument(s) to your local VISA configuration using instrument manager software like NI-MAX, it is a good time to test the connection to the instrument(s) and verify that communications are working.

To test that communication with the instrument is successful:

1. In NI-MAX, click on your instrument so that its properties load in the main panel to the right. At the top of the screen you will see a button that says, “Open VISA Test Panel”. Click on this button and the VISA Test Panel will open and connect to the instrument.

2. In the VISA Test Panel, click on the Input/Output tab. From this tab panel you can test out sending commands and reading back the responses from the instrument.

3. By default, the VISA Test Panel will have the *IDN? command entered in to the “Select or Enter Command” box. Click the Query button to send this command to the instrument and then read back the response. If the VISA Test Panel displays the response and does not list any errors, then you know that your connection to the instrument is working correctly.

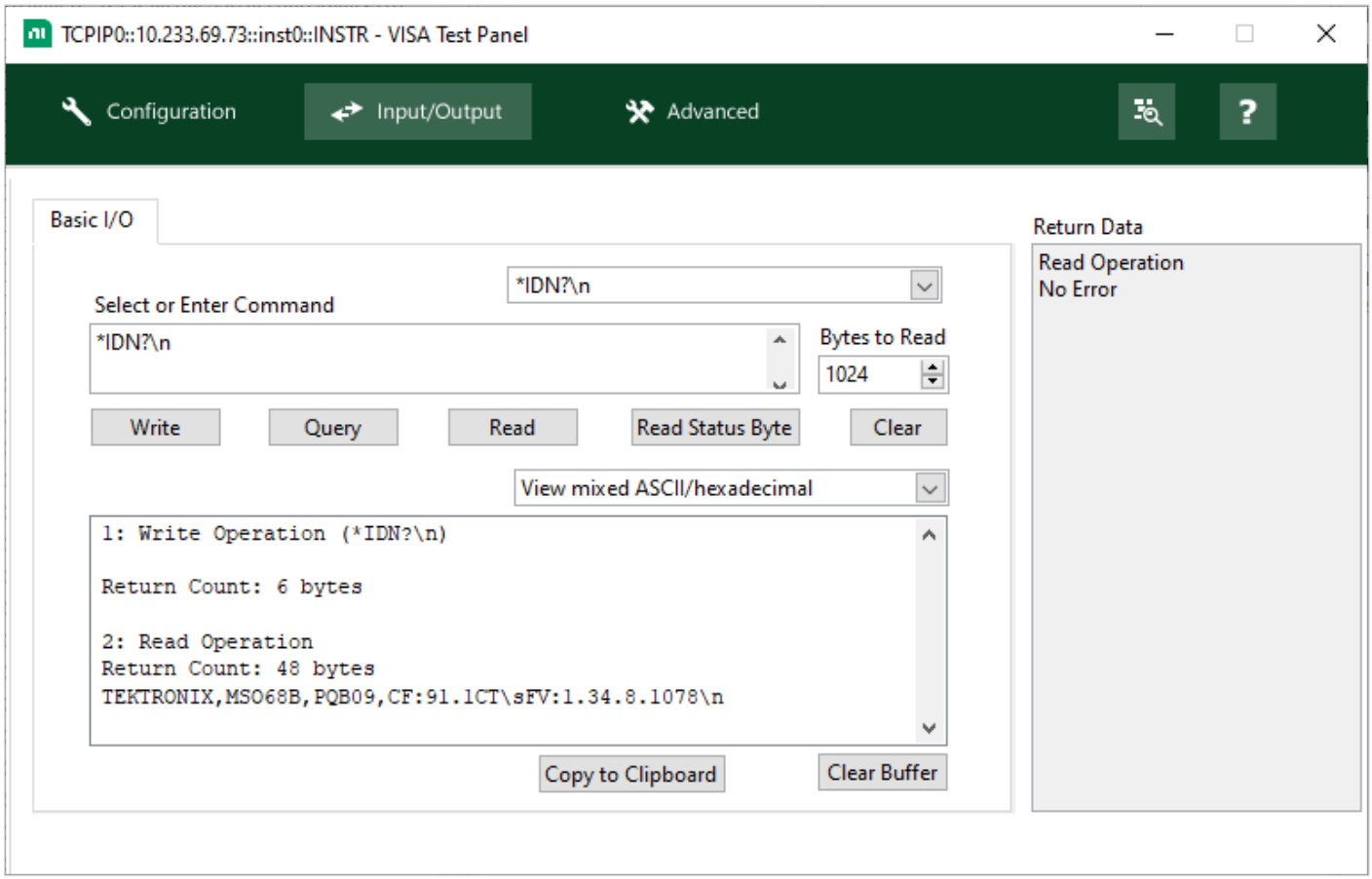


Figure 12: The VISA Test Panel can be used to send commands to instruments and read back responses without writing code. This is useful for testing out commands.

At this point you can continue testing out commands with the VISA Test Panel. Refer to your instrument’s Programmer’s manual for a full list of commands for the instrument and try some of them out using the VISA Test Panel.

Debugging Instrument Communication Using I/O Trace

While developing automated test applications, you will likely encounter bugs in your code that can be difficult to diagnose. Often the issue is caused by sending incorrect commands to the instrument or not getting back the expected response from the instrument. Debugging these problems can be tedious and is often performed by adding series of print statements to the code. Much of this tedium can be avoided by using developer tools that are included with VISA.

When you installed NI-VISA, the installer also installed a piece of software called I/O Trace. I/O Trace is a utility that monitors communication between the PC application and the instrument that is performed using the VISA library. Every command sent to the instrument and every response read from the instrument can be monitored using I/O Trace. The information collected by I/O Trace includes:

- The VISA operation being called
- The parameters of the call including:
  - The address of the instrument being communicated with
  - The read/write data buffer contents
  - The number of bytes read/written
  - Etc.
- The return status of the call (Success, Error, etc.)
- Timing information for the VISA call including the duration of the call

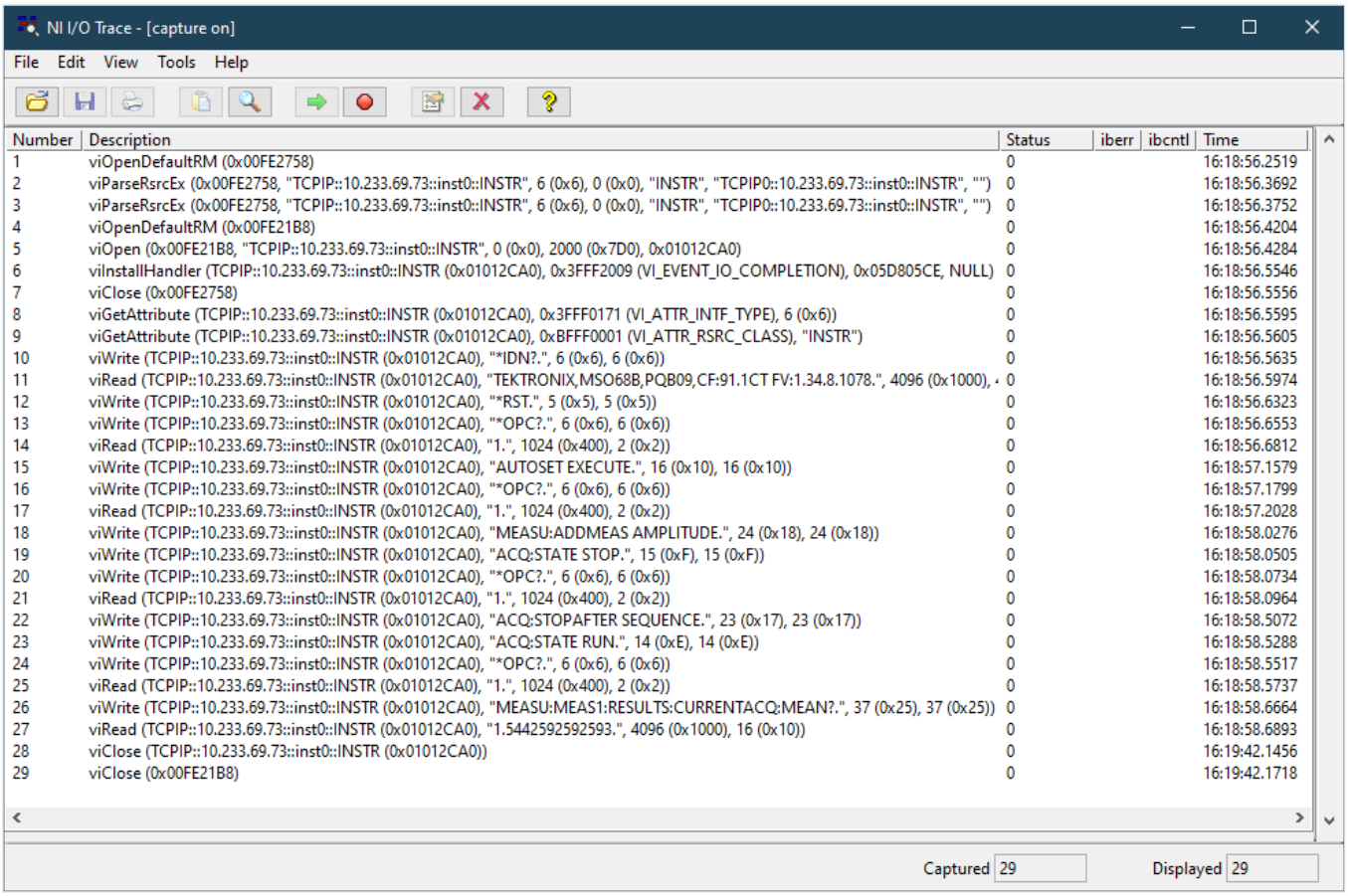


Figure 13: I/O Trace is an invaluable tool for debugging instrument control code.

In your test code you will commonly use string formatting methods to create your command strings from variables within your code or to read values from strings returned by the instrument. The I/O Trace utility is extremely useful to determine not only what your formatted string looks like, but exactly what characters and bytes were sent to the instrument. This utility will also tell you exactly what characters and bytes the instrument is sending back, that is, what your code is receiving. Being able to monitor exactly what is being written and read is extremely helpful to determine if your code is actually sending what you think it is sending and actually receiving what you think it should be receiving.

The I/O Trace utility is also extremely useful for determining where an error in your code is coming from. When VISA errors occur, this utility will show you exactly which call generated the error. Having the trace available makes it easy to determine what calls were made, what commands were sent to the instrument, what data was read leading up to the point where the error occurred. This is very helpful in determining where in your code the error was generated.

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

7/2023 SBG 61W-74019-0