Document Type: Tutorial
NI Supported: Yes
Publish Date: Oct 24, 2011

USB Instrument Control Tutorial

Overview

This tutorial is meant as a starting point for using NI-VISA to communicate with a USB device. It is not intended as a starting point for learning about USB architecture or the various protocols used in USB communication. After reading this tutorial, you should be able to install a USB device and use NI-VISA to communicate with that device, as long as you understand the device communication protocol.

Table of Contents

1. USB and VISA Background

This document is part of the Instrument Control Fundamentals Series

VISA is a high-level API used to communicate with instrumentation buses. It is platform independent, bus independent, and environment independent. In other words, the same API is used regardless of whether a program is created to communicate with a USB device with LabVIEW on a machine running Windows 7 or with a GPIB device with C on a machine running Mac OS X.

USB is a message-based communication bus. This means a PC and a USB device communicate by sending commands and data over the bus as text or binary data. Each USB device has its own command set. You can use NI-VISA Read and Write functions to send these commands to an instrument and read the response from an instrument. Check with your instrument manufacturer for a list of valid commands for your instrument.

Starting with version 3.0, NI-VISA supports USB communication. Two classes of VISA resources are supported: USB INSTR and USB RAW.

USB devices that conform to the USB Test and Measurement Class (USBTMC) protocol use the USB INSTR resource class. USBTMC devices conform to a protocol that the VISA USB INSTR instrument control resource class can understand. No configuration is necessary to communicate with a USBTMC device. To communicate with a USBTMC instrument, refer to section 3. For more information about the USBTMC specification, refer to the USB Implementers Forum Web page linked below.

USB RAW instruments are any USB instrument other than those instruments that specifically conform to the USBTMC specification. If you are using a USB RAW device, follow the instructions in section 2 to configure NI-VISA to control your device. Contact your instrument manufacturer for details about the communication protocol and the command set your instrument uses.

For specific information about the NI-VISA API, refer to the *NI-VISA Help*. This document is included with NI-VISA and is available through the link at the end of this tutorial.

2. Configuring NI-VISA to Control Your USB Device

Watch a 3 minute video on Getting Started with Instrument Control using USB

This section walks through the steps for configuring a USB RAW device to be controlled by NI-VISA 5.0 on a Windows-based computer. If you are using a USBTMC-compatible device, connect your device and skip to the third step, Test the device with NI-VISA Interactive Control.

At this point, NI-VISA already should be installed on your computer, and your USB device should not be connected. Furthermore, you should not have a driver for your USB device installed. There are three steps to configure your USB device to use NI-VISA:

1. Create the INF file using the Driver Development Wizard.
2. Install the INF file and the USB device using the INF file.
3. Test the device with NI-VISA Interactive Control.

For the purposes of this tutorial, a Creative Webcam is used as an example USB device and is installed on a Windows 7 system. Because this tutorial is intended to explain the configuration of a generic USB device, details specific to the Creative Webcam are not discussed.

2.1. Create the INF File Using the Driver Development Wizard

To use NI-VISA, you must first tell Windows to use NI-VISA as default driver for the device. In the Windows environment, you can do this with an INF file. NI-VISA 3.0 and higher includes the NI-VISA Driver Wizard to create an INF file for your USB device.

1. To open the NI-VISA Driver Wizard, select **Start»All Programs»National Instruments»VISA»Driver Wizard**. Figure 1 shows the open screen.

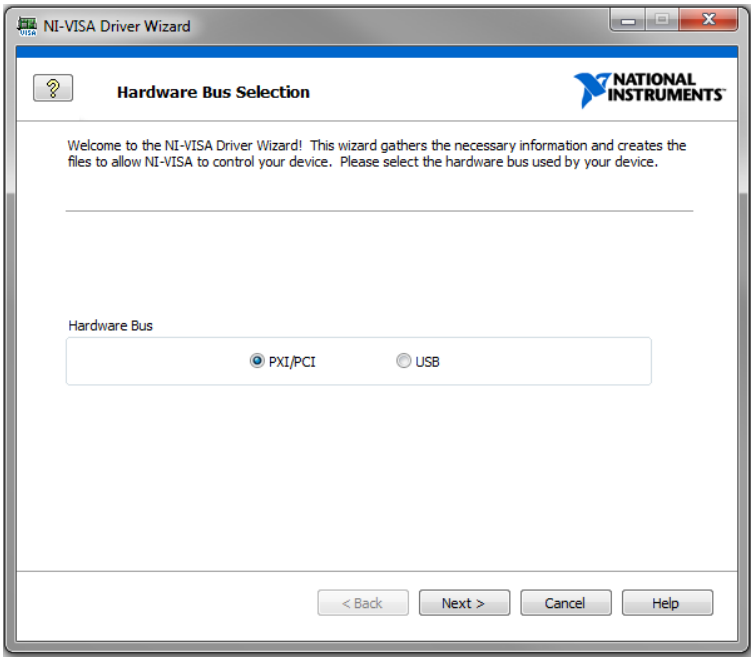

Figure 1. NI-VISA Driver Wizard Hardware Bus Selection Window

You can use this wizard to create an INF file for a PXI/PCI, or USB device. Because you are creating the driver for a USB device, click **USB** and **Next**. The NI-VISA Driver Wizard USB Device Selection window opens as shown in Figure 2.

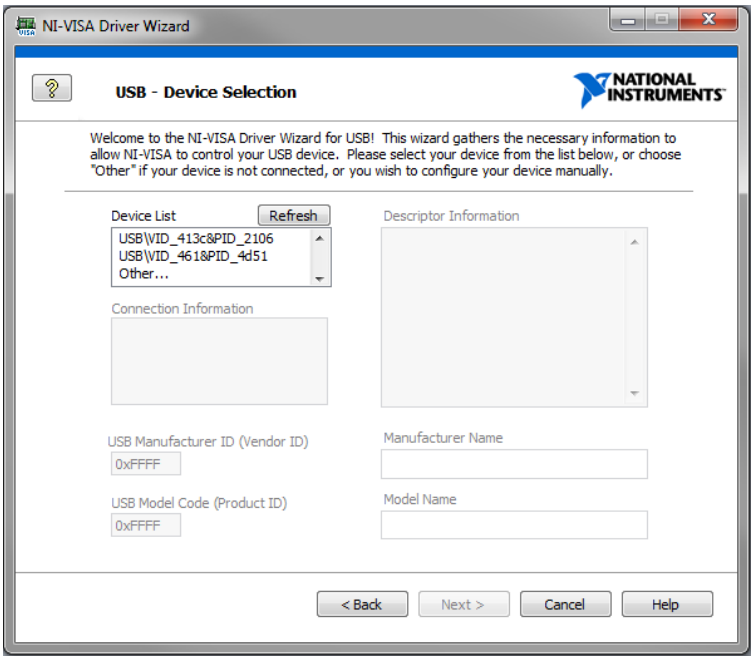

Figure 2. NI-VISA Driver Wizard USB Device Selection Window

2. For this step, you must know the USB vendor ID and product ID for your USB instrument. These numbers identify your USB device when you install it and address your device when you want to communicate with it. According to the USB specification, both numbers are 16-bit hexadecimal numbers and should be provided by the device manufacturer.

If you have access to your USB device, go ahead and connect it to your computer at this time, but cancel out of the Found New Hardware Wizard if it starts. Complete step 2a.

If you do not currently have access to your USB device, but know your USB vendor ID and product ID complete step 2b.

If you do not currently have access to your USB device and do not know your USB vendor ID and product ID, obtain that information from the manufacturer, then complete step 2b.

2a. Open the Device Manager from the Control Panel and find your device on the list, usually under "Other Devices." It may show a yellow exclamation mark indicating it is an unknown device. Double-click the device to open the properties. Select the Details tab and ensure that "Hardware Ids" shows in the attribute drop-down box. A string of characters will be displayed similar to Figure 3. The four characters to the right of "VID_" and "PID_" are your vendor ID and product ID, respectively. Write down the characters for your device, close the Device Manager, and unplug the device from the computer.

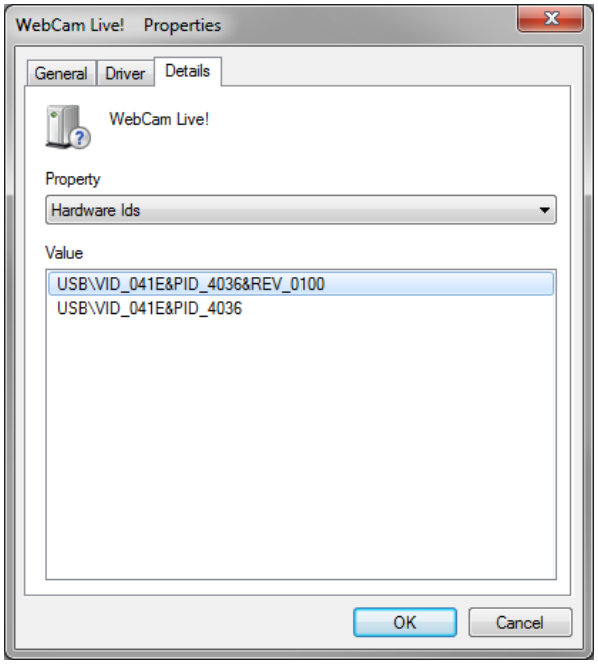

Figure 3. Finding the Hardware IDs from the Device Manager

For this Creative Webcam, the vendor ID and product ID are 0x041E and 0x4036, respectively. Note that the vendor ID and product ID will be different for your device depending on the manufacturer and model.

In the NI-VISA Driver Wizard USB Device Selection window use your vendor ID and product ID to match to one of the items in the Device List. Once you've clicked on this device, you should see the other fields in the window populate with information about your device. Verify that this information looks correct. If you do not see a match for your device under Device List, first try clicking **Refresh**, and if that still does not work then skip to step 2b.

Click **Next**. The Output Files Generation window is displayed as shown in Figure 6.

2b. In the NI-VISA Driver Wizard USB Device Selection window, choose **Other** from the Device List.

Click **Next.** The following prompt shown in Figure 4 will recommend you to connect your USB device.

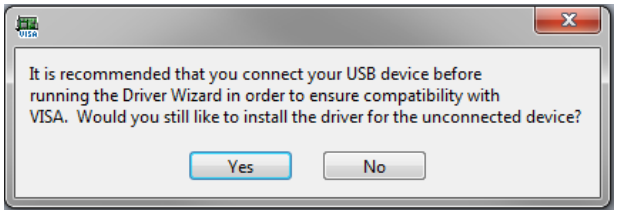

Figure 4. Prompt recommending you to connect your USB device

Click **Yes.** The USB - Device Information windows is displayed as shown in Figure 5.

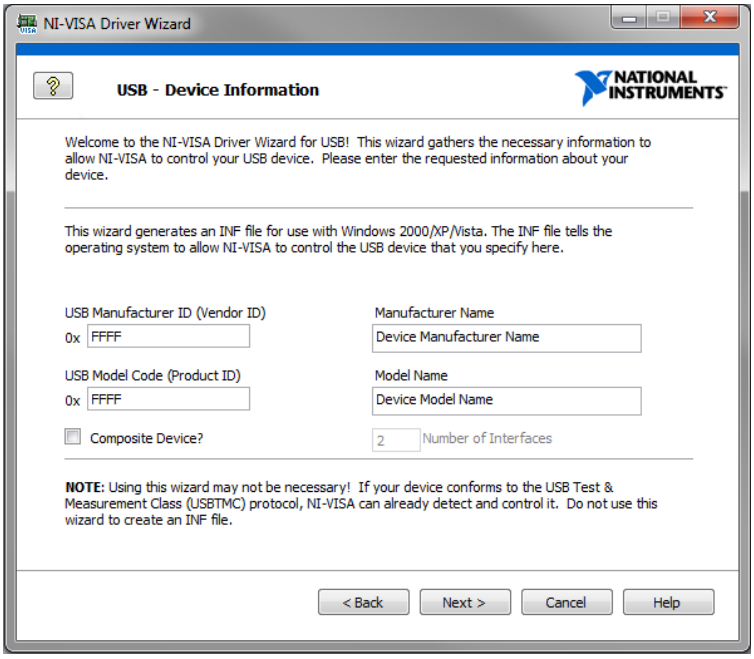

Figure 5. NI-VISA Driver Wizard USB - Device Information Window

Enter the information you've collected about your device into the four fields shown in Figure 5.

Click **Next**. The Output Files Generation window is displayed as shown in Figure 6.

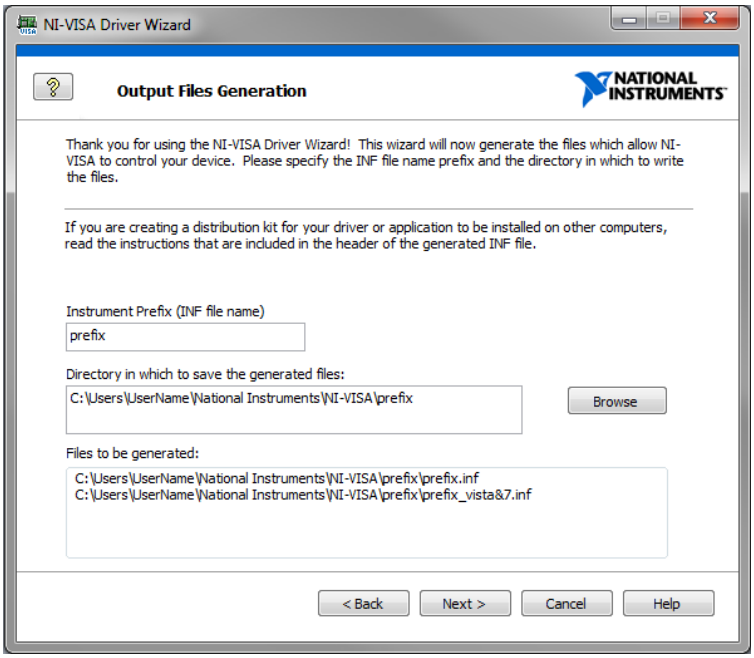

Figure 6. NI-VISA Driver Wizard Output Files Generation Window

The USB Instrument Prefix is simply a descriptor that you will use to identify the files used for this device, and you can choose any prefix you would like. Enter a USB instrument prefix, select the desired directory in which to place these files, and click **Next**. The Installation Options window is displayed as shown in Figure 7.

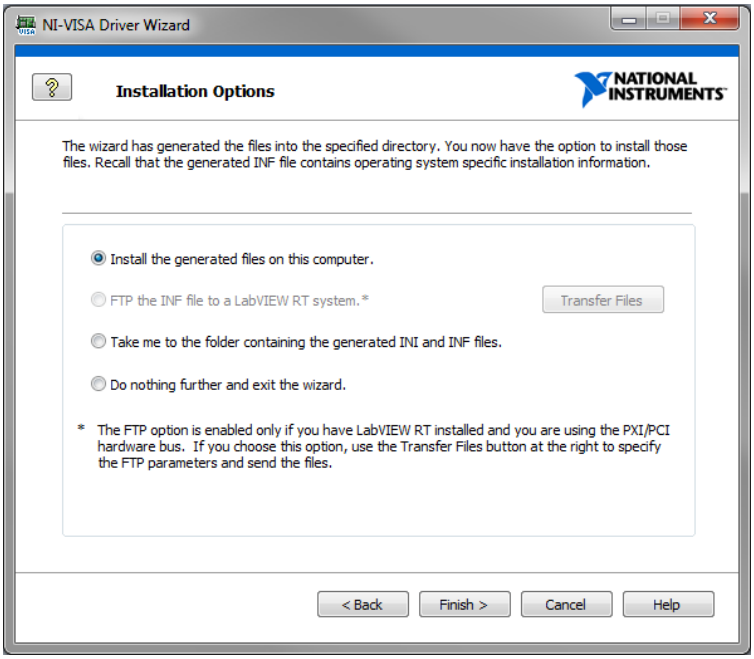

Figure 7. NI-VISA Driver Wizard Installation Options

The default selection is to install the generated files on your computer and is usually the best option. Once you select an option, click **Finish** to exit the wizard. The INF file is created in the directory you specified in the output file directory field in the previous window.

2.2. Install the INF files and the USB device.

The installation of INF files is different for each version of Windows. Because of the differences between Windows 2000/XP and Windows Vista/7 the NI-VISA Driver Wizard will create two an INF files, one for each OS group. Installation instructions are included in a header at the top of the INF file. Because INF files are ASCII text files, they can be read in any text editor such as Notepad. For detailed information about installing your INF file, open your INF file in a text editor and follow the instructions at the top of the file. This tutorial assumes you are using Windows 7.

1. To install INF files on Windows 7 you will first need to be logged into an Administrator user account.
2. You will need to unplug the USB device at this time so that the drivers will be properly associated with your device upon install.
3. Copy the INF file to the INF folder. On Windows 7, this folder is usually at **C:\WINDOWS\INF**. This folder may be hidden, so you may need to change your folder options to view hidden files by going to **Tools»Folder Options»View»Advanced Settings** and selecting **Show hidden files, folders, and drives.**
4. Right-click on the INF file in **C:\WINDOWS\INF** and click **Install**. This process creates a PNF file for your device. You are now ready to install your USB device.
5. Connect your USB device. Windows should be able to detect your USB device, and the Add New Hardware Wizard should open automatically as soon as you connect it to the USB port. After an installation period Windows should indicate that has successfully installed your device.

Note: In some cases, Windows may already have a default driver associated with your USB device. If this is the case, Windows will install that driver first. Once you've plugged in your USB device and Windows has installed the default driver, open the Device Manager from the Control Panel. Once Device Manager is open, expand the tree category appropriate to your device, ie: Human Interface Devices. Then find out which instance of "USB Human Interface Device" (shown in Figure 5) corresponds to your USB device by right-clicking and selecting Properties and checking the Details tab for the matching VID and PID as detailed in section 2a above.

Once you've found the "USB Human Interface Device" that has the matching VID and PID of your USB Device, right-click on it and choose **Update Driver Software.**

On the first screen, select **Browse my computer for driver software**. On the second screen, select **Let me pick from a list of drivers on my computer**. On the third screen, shown in Figure 8, click **Have Disk.** Direct the prompt browse to **C:\WINDOWS\INF,** and choose the INF file you copied there and click **OK**. Make sure your device is selected in the window shown in Figure 8, then click **Next.** Windows may issue a warning about not being to verify the driver, choose to install the driver anyways. When the driver has finished installing click Finished.

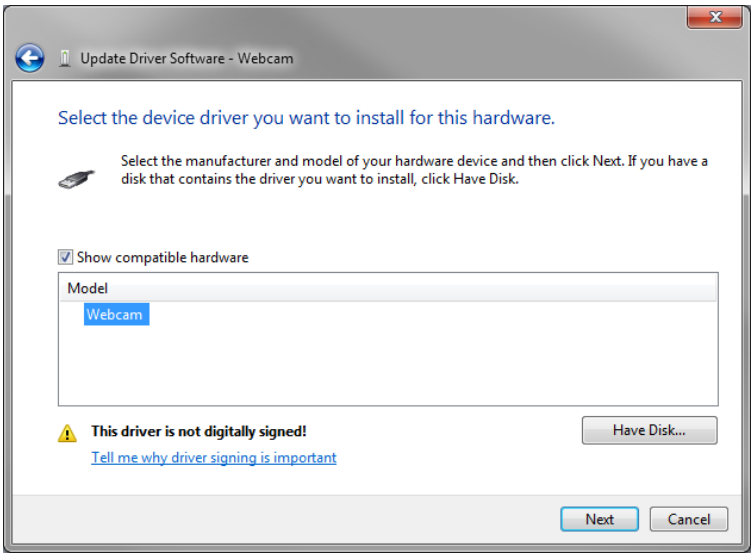

Figure 8. Select the driver for Your USB Device

2.3. Test Communication with VISA Interactive Control.

1. Open Measurement & Automation Explorer (MAX). Select **Tools»Refresh** to refresh the view to ensure your device appears. Your USB device should be listed under **Devices and Interfaces** as shown in Figure 9. Your USB device is now installed and configured to use NI-VISA.

Once you've selected your USB device, the device information can be displayed by clicking the **USB Settings** tab at the bottom of the right pane. Using this window, you can access information such as the manufacturer ID, model code, and serial number for your device.

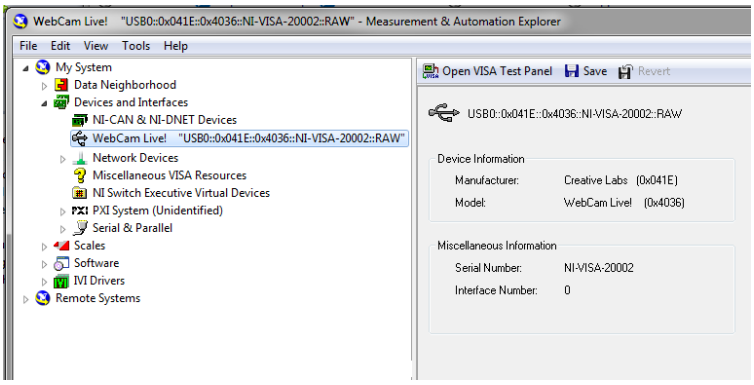

Figure 9. USB Device Shown in MAX

2. To communicate with your device using VISA, use the VISA instrument descriptor for your device. The instrument descriptor format for a USB INSTR device is USB[board]:: manufacturer ID:: model code:: serial number[:: USB interface number]::INSTR. The instrument descriptor format for a USB RAW device is USB[board]:: manufacturer ID:: model code:: serial number[:: USB interface number]::RAW.

According to the USBTMC specification, all USBTMC devices must have a serial number. Some USB RAW devices may not have serial numbers. If your device does not have a serial number, NI-VISA automatically assigns a VISA specific serial number for that device, as shown above in Figure 9. The format for the serial number is NI-VISA-#, where # is an automatically generated number.

Some USB devices have multiple interfaces. This is similar to the way a PCI device can have multiple functions. If your device only supports one interface, you do not need to include the USB interface number.

The Creative Webcam in this example uses the RAW class, and the manufacturer code and model code are 0x041E and 0x4036, respectively. For the Creative Webcam, the instrument descriptor is USB0::0x041E::0x12C0::NI-VISA-20002::RAW.

To test communication with this device, open MAX. Select **Tools»NI-VISA»VISA Interactive Control**. A window similar to that shown in Figure 10 should open.

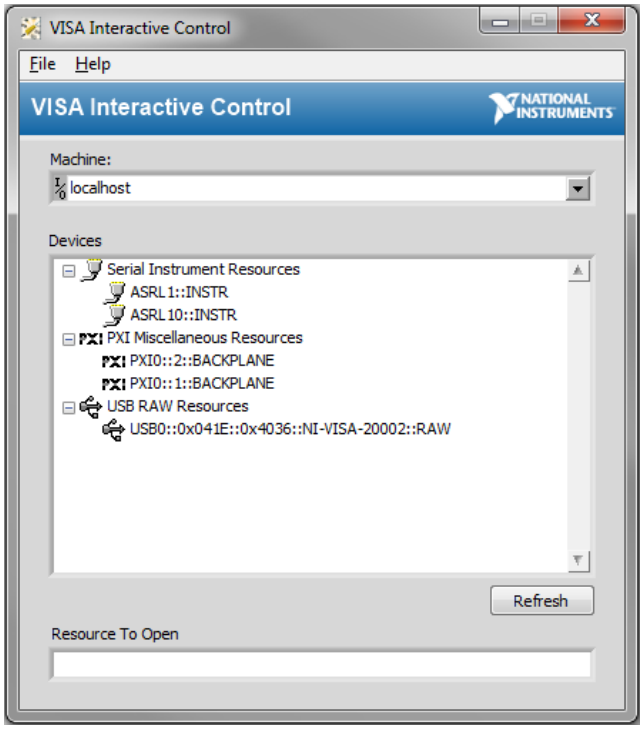

Figure 10. VISA Interactive Control

3. The VISA Interactive Control (VISAIC) is a utility program used to communicate easily with any VISA resource. After your USB device is configured to use VISA, it should be listed in the USB branch. Double-click on your device to open a VISA session to the device. The window shown in Figure 11 should open.

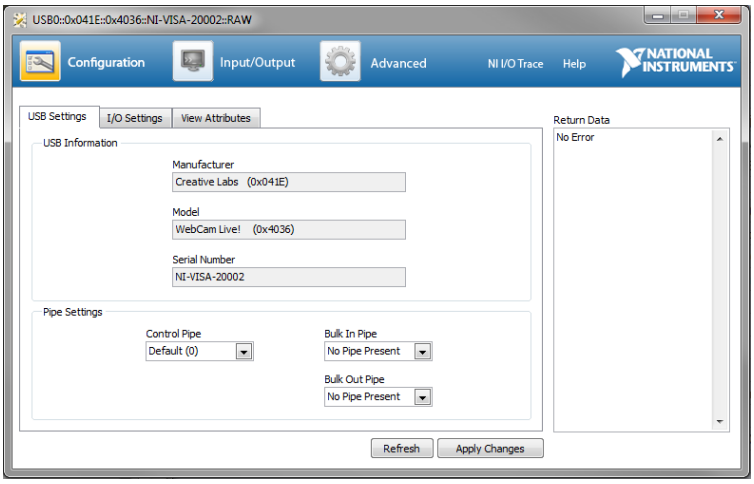

Figure 11. VISA Interactive Control Open VISA Session

When you open a VISA Session with VISAIC, the Configuration section and the USB Settings tab are automatically selected. To read the properties of your device, select the **View Attributes** tab. This lists the different attributes of your device, including information like Resource Name.

For more information about using VISAIC, refer to *Developer Zone: VISA Interactive Control (VISAIC)*. For information about the NI-VISA API, review the *NI-VISA Help*. Both are available through the links at the end of this tutorial. For a list of valid commands for your USB instrument, contact your instrument's manufacturer.

3. Using NI-VISA to Communicate with Your USB Device

This section explains how to communicate with your USB device using NI-VISA 3.0 and above. Recall that there are two classes of USB devices. The method of communication depends on the class of your device.

3.1 USB INSTR Class (USBTMC)

Devices that conform to the USB Test and Measurement Class (USBTMC) use the NI-VISA USB INSTR class. These devices use 488.2 style communication. For these devices, you can simply use the VISA Open, VISA Close, VISA Read, and VISA Write functions in the same way you would if you were communicating with GPIB instruments.

Figure 12 illustrates a LabVIEW VI that communicates with a USBTMC device. In this example, a VISA session is opened to a USB device. A command is written to the device, and the response is read back. In this example, the specific command being sent is the ID query for the device. Check with your device manufacturer for your device command set. After all communication is complete, the VISA session is closed.

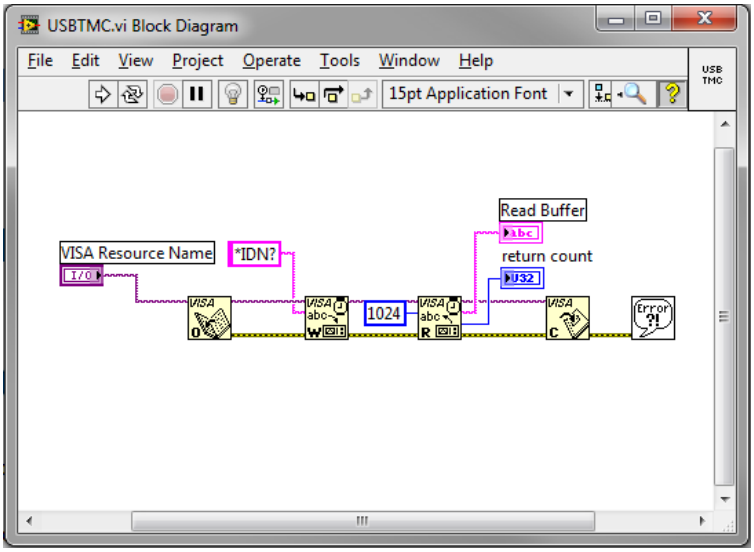

Figure 11. USBTMC LabVIEW Example Block Diagram

3.2 USB RAW Class

Communicating with the USB RAW class is more complicated because each device may use its own communication protocol. Contact your device vendor for details about the communication protocol for your device.

USB communicates using four types of pipes or endpoints: control, bulk, interrupt, and isochronous. Each type of pipe transfers a different type of information. Furthermore, any number of

endpoints can be of any endpoint type. Think of an endpoint as a communication socket. For specific details about USB architecture, review the USB specification linked at the end of this article.

NI-VISA supports three types of USB pipes: control, bulk, and interrupt. When NI-VISA detects your USB instrument, it automatically scans your instrument for the lowest available endpoint for each type.

When VISA detects the lowest available endpoint, it assigns that value to the appropriate VISA attribute. The bulk in endpoint and bulk out endpoint are stored in the VI_ATTR_USB_BULK_IN_PIPE attribute and the VI_ATTR_USB_BULK_OUT_PIPE attribute, respectively. The interrupt in endpoint is stored in the VI_ATTR_USB_INTR_IN_PIPE attribute. A value of -1 indicates that a USB device does not support that type of pipe. For the control pipe, only endpoint zero is supported. If you are using the C API, use the viSetAttribute function to change endpoints. In LabVIEW, use a Write VISA Property node.

NI-VISA includes four functions to transfer data through USB pipes. Before you can communicate with your device using these functions, you need to set up the communication protocol using the VISA USB attributes. The following list describes the available functions.

- Use VISA USB Control In and VISA USB Control Out to transfer data using the control pipe.
- To transfer data using a bulk pipe, use VISA Read and VISA Write.

If you are using LabVIEW, VISA includes an additional function to use the interrupt pipe: VISA Get USB Interrupt Data. In the C API, you can do this by accessing the VI_ATTR_USB_RECV_INTR_SIZE and VI_ATTR_USB_RECV_INTR_DATA attributes of the VI_EVENT_USB_INTR event object. See the *NI-VISA Help* for more information about VISA Events.

4. USB on Linux® and Mac

4.1 Linux

NI-VISA relies on a Linux kernel feature for its USB support. This feature is called *usbfs*, and on older Linux kernels was referred to as *usbdevfs*. For NI-VISA to support USB devices, this feature must be present and mounted (like a virtual file system). This is supported in most major Linux distributions such as Red Hat, SuSE, and Mandrake. You may use the mount command to display what file systems are currently mounted to see if your system currently supports this feature.

Also, the VISA user must have write access to the file that represents the USB device, which is typically somewhere in a subdirectory within /proc/bus/usb. If this is not the case, the USB device is not accessible by VISA (it will not be found using viFindRsrc, and viOpen will fail). The default configuration on most systems is that the root user has write access; however, no other user has this access.

There are several options for providing a nonroot user access to a USB device.

- Use the hotplug package. This package is installed by default on most distributions including Red Hat, SuSE, and Mandrake. The hotplug package allows the user to run scripts for a specific USB device based on characteristics such as Vendor ID (VID) and Product ID (PID). If the hotplug package exists, the NI-VISA Installer by default will install scripts to give all users write access to all USB TMC devices and a framework for USB RAW devices. To add write permissions for a specific USB RAW device, run the included script:

  <VXIPNPPATH>/linux/NIvisa/USB/AddUsbRawPermissions.sh

  For more information about the hotplug package, refer to the following Web site: http://linux-hotplug.sourceforge.net/.
- usbfs (formerly known as usbdevfs) may be mounted with the option devmode=0666. This gives all users read and write access to all USB devices.
- The root user may add write permissions to the file that represents the USB device, which is typically somewhere in a subdirectory within /proc/bus/usb. Unfortunately, these permissions are lost if the device is unplugged. Therefore, this approach is not recommended.

4.2 Mac OS X

As long as no other driver on the system claims the USB device, you can use NI-VISA to access it. No special setup is required.

Related Links:
Return to the Instrument Control Fundamentals Homepage
Developer Zone: VISA Interactive Control (VISAIC)
Developer Zone: Using the VISA Driver Development Wizard and NI-VISA to Develop a PXI/PCI Driver in Windows
Products and Services: Instrument Control Software
Products and Services: Ethernet/LXI