

Using MCC GPIB Products with LabVIEW

* This application note applies to PCI-GPIB-1M, PCI-GPIB-300K, PCM-GPIB, as well as to ISA- and PC104- boards

How NI Compatibility Works

National Instruments (NI) provides the NI-488.2 GPIB package for use only with NI hardware. NI-488.2 contains a library of functions—such as `ibfind`, `ibrd`, and `ibwrt`—that are used to control the device. NI-488.2 provides an interface between the programmer and NI’s GPIB hardware driver.

To use MCC GPIB hardware, install the MCC GPIB Library instead of NI-488.2. The MCC GPIB Library contains similar functions as those found in NI-488.2. The MCC GPIB Library, however, provides a link to MCC GPIB hardware rather than to NI GPIB hardware. This relationship is illustrated below.

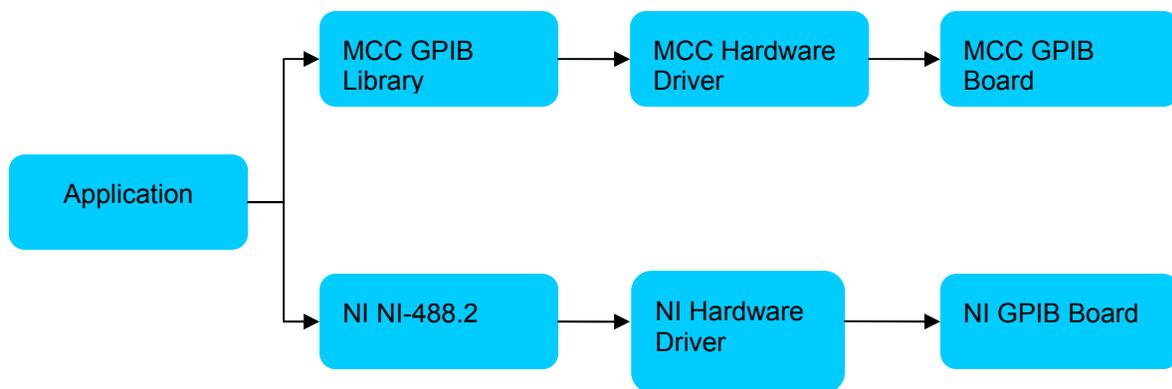


Figure 1. How an application communicates with GPIB hardware

NI-488.2 and the MCC GPIB Library use some files with identical names, and cannot be installed on the system at the same time; therefore, NI and MCC GPIB hardware cannot co-exist on the same machine. If the NI 488.2 package is installed, MCC GPIB hardware does not function, and if the MCC GPIB package is installed, NI GPIB hardware does not function.

LabVIEW GPIB VIs

The GPIB VIs found on the LabVIEW Function Palette under All Functions ► Instrument I/O ► GPIB and All Functions ► Instrument I/O ► GPIB ► 488.2 are used to access the NI-488.2 driver and communicate with NI GPIB hardware. In order to use MCC GPIB hardware, NI-VISA should be used as the I/O interface.

NI-VISA and MCC GPIB Hardware

VISA stands for **V**irtual **I**nstrument **S**oftware **A**rchitecture. Its main component is a file called `visa32.dll` that contains a list of functions used to communicate with instruments through various interfaces, including GPIB, VXI, PXI, Serial, Ethernet, and USB. The VISA functions are not intended for a specific interface; but provide a way for applications to communicate with instruments regardless of the interface used. Each instrument attached to the computer is assigned a VISA identifier which describes its interface type, bus address, device address, and function.

For example, a GPIB oscilloscope might be described as “GPIB0::5::INSTR”, meaning the instrument on GPIB bus 0 at device address 5.

When VISA functions communicate with GPIB devices, they use the same basic GPIB functions described earlier. This means that VISA functions can communicate with MCC GPIB hardware through the MCC GPIB Library as shown below.

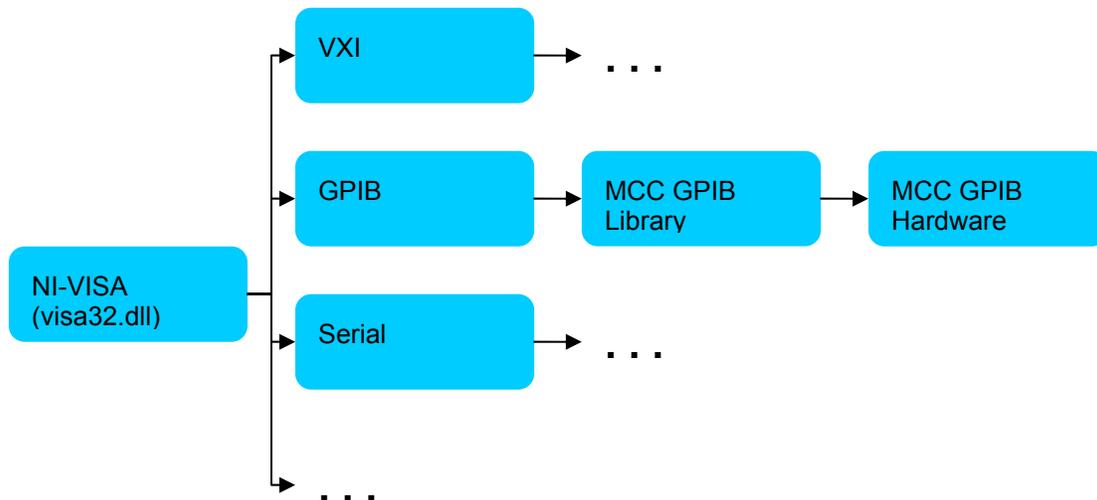


Figure 2. How NI-VISA communicates with MCC GPIB Hardware

With NI-VISA, National Instruments includes a utility called VISA Interactive Control that detects and tests any instruments connected to your computer. This utility is available from the Windows Start menu under **Programs ► National Instruments ► VISA**. To view and test instruments connected to MCC GPIB hardware with this utility, perform the steps below.

1. Make sure that you have a valid development or deployment license for NI-VISA. For more information on obtaining such a license, check the National Instruments website at www.ni.com/visa.
2. Install the MCC GPIB software and controller board.
3. Connect at least one GPIB instrument to the MCC GPIB controller board.
4. Configure the MCC controller board in the MCC CBCONF32 utility in accordance with the *GPIB Programming Library User's Guide* (www.mccdaq.com/PDFmanuals/GPIBSoftware.pdf).
5. While you are configuring the controller board, change the **Disable Auto Serial Polling** setting to "NO".
6. Configure each instrument attached to the GPIB board in CBCONF32. Make sure that the primary (and secondary, if applicable) GPIB address for the instrument is configured correctly. Save the settings and exit CBCONF32.
7. Start NI's VISA Interactive Control utility. The GPIB controller board and instrument are listed under the GPIB heading on the VISA I/O tab.

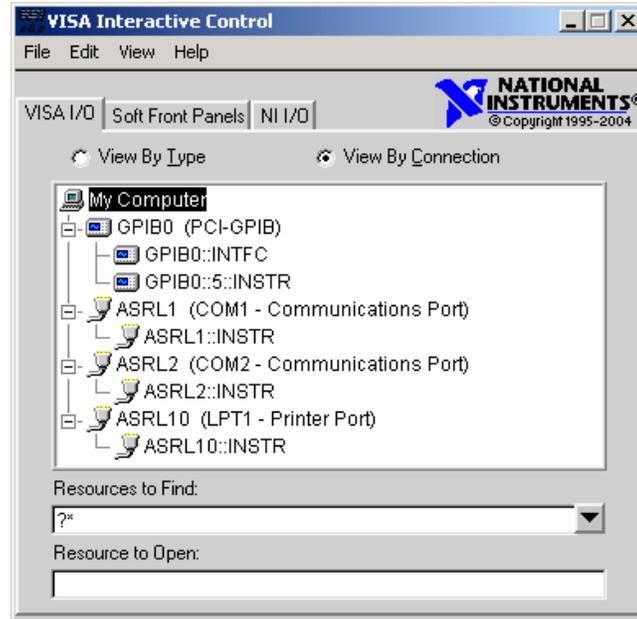


Figure 3. VISA Interactive Control

8. To test communication, double click on the instrument address to bring up the instrument session window. You can send commands and receive data from the device under the **Basic I/O ► Read** and **Basic I/O ► Write** tabs.

IMPORTANT: Make sure that the **Async** checkbox on these tabs is unchecked. The MCC GPIB driver does not support asynchronous communication.

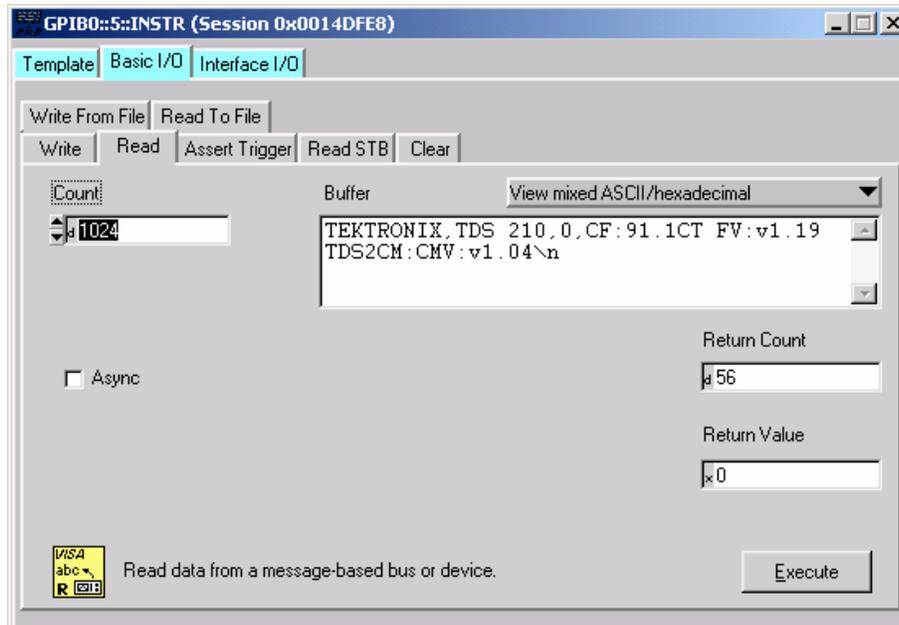


Figure 4. Testing I/O in VISA Interactive Control

LabVIEW VISA VIs

The LabVIEW VISA VIs—available on the LabVIEW Function Palette under **All Functions ► Instrument I/O ► VISA**—use the `visa32.dll` to communicate with and control connected instruments. The VISA VIs can be used to communicate with MCC GPIB hardware as shown below.

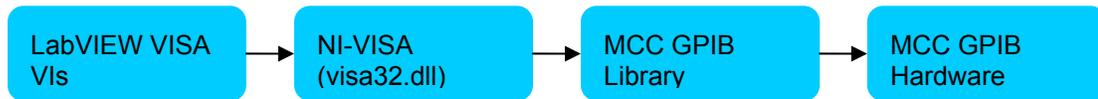


Figure 5. How LabVIEW VISA VIs communicate with MCC GPIB hardware

The VISA “Locking.vi” example included with LabVIEW shows how to use the VISA VIs with GPIB hardware. It is available from the LabVIEW Example Finder under **Hardware Input and Output ► VISA** in the task list.

MCC GPIB Hardware and the Measurement and Automation Explorer

The NI Measurement and Automation Explorer (MAX) configures and tests NI’s GPIB hardware. It provides essentially the same functionality that MCC’s CBCONF32, CBTEST32, and CBIC programs do. MAX cannot detect or configure MCC GPIB hardware. Likewise, CBCONF32 cannot detect or configure NI GPIB hardware.

MCC GPIB Hardware and IVI-based Instrument Drivers

IVI stands for **I**nterchangeable **V**irtual **I**nstruments. It is an industry-standard originally developed by National Instruments for writing instrument drivers. An instrument driver is a collection of higher-level functions that can communicate with a specific instrument. The higher-level functions shield the user from having to know the library of commands for each specific device. IVI enables programmers to write an application generically for a type of instrument and use the code interchangeably with any device of that type.

An IVI-compliant instrument driver consists of two parts: a class driver and an instrument-specific driver. The class driver contains a standard list of functions for a particular device class. The IVI-standard driver defines eight different instrument classes: digital multimeter, oscilloscope, DC power supply, switch, power meter, spectrum analyzer, function generator, and RF signal generator. The instrument-specific driver provides implementation for functions associated with a specific device and also provides an interface between the class driver and lower-level functions.

IVI drivers communicate with instruments through VISA. Since this is the case, some compatibility between IVI drivers and MCC GPIB hardware is expected. However, the drivers are not developed or tested by MCC, so compatibility is not guaranteed. The IVI control path is shown below.

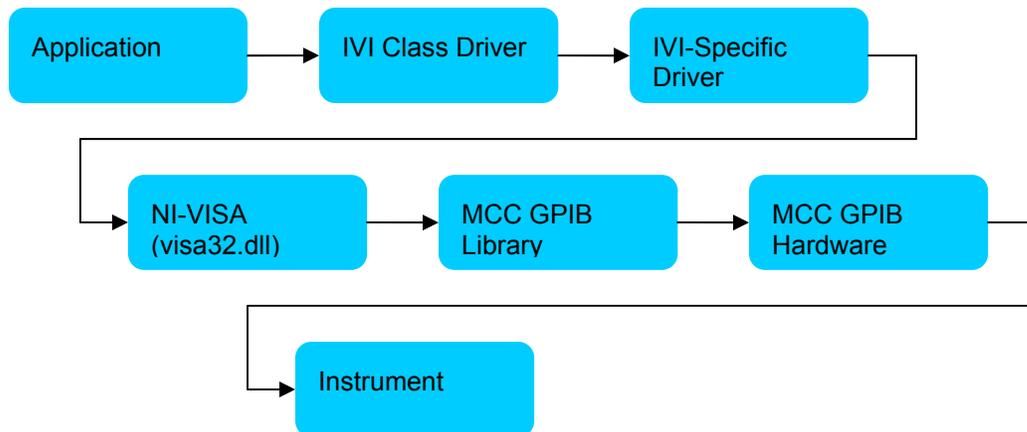


Figure 6. How IVI communicates with instruments through MCC GPIB hardware

IVI devices can be configured with logical names through MAX that can be used to refer to the device in LabVIEW applications. To configure an instrument connected to a MCC GPIB board for use with an IVI driver, perform the following steps:

1. Follow the steps outlined above under *NI-VISA and MCC GPIB Hardware* for configuring the board for use with VISA.
2. Ensure that the latest version of NI's IVI Compliance Package is installed
3. Download and install the IVI compliant instrument driver for your instrument from NI's website (http://sine.ni.com/apps/we/niid_web_display.driv_search).
4. Start the Measurement and Automation Explorer and open the IVI Drivers node under **My System**.
5. Right click on **Logical Names** and click **Create new (case sensitive)**.... Type the logical name you want to use for your instrument, for example "Fluke Multimeter."
6. In the **Driver Session** drop-down box on the **General** tab in the right pane, choose the instrument driver you downloaded in step 3. Then click the **Go To** button.

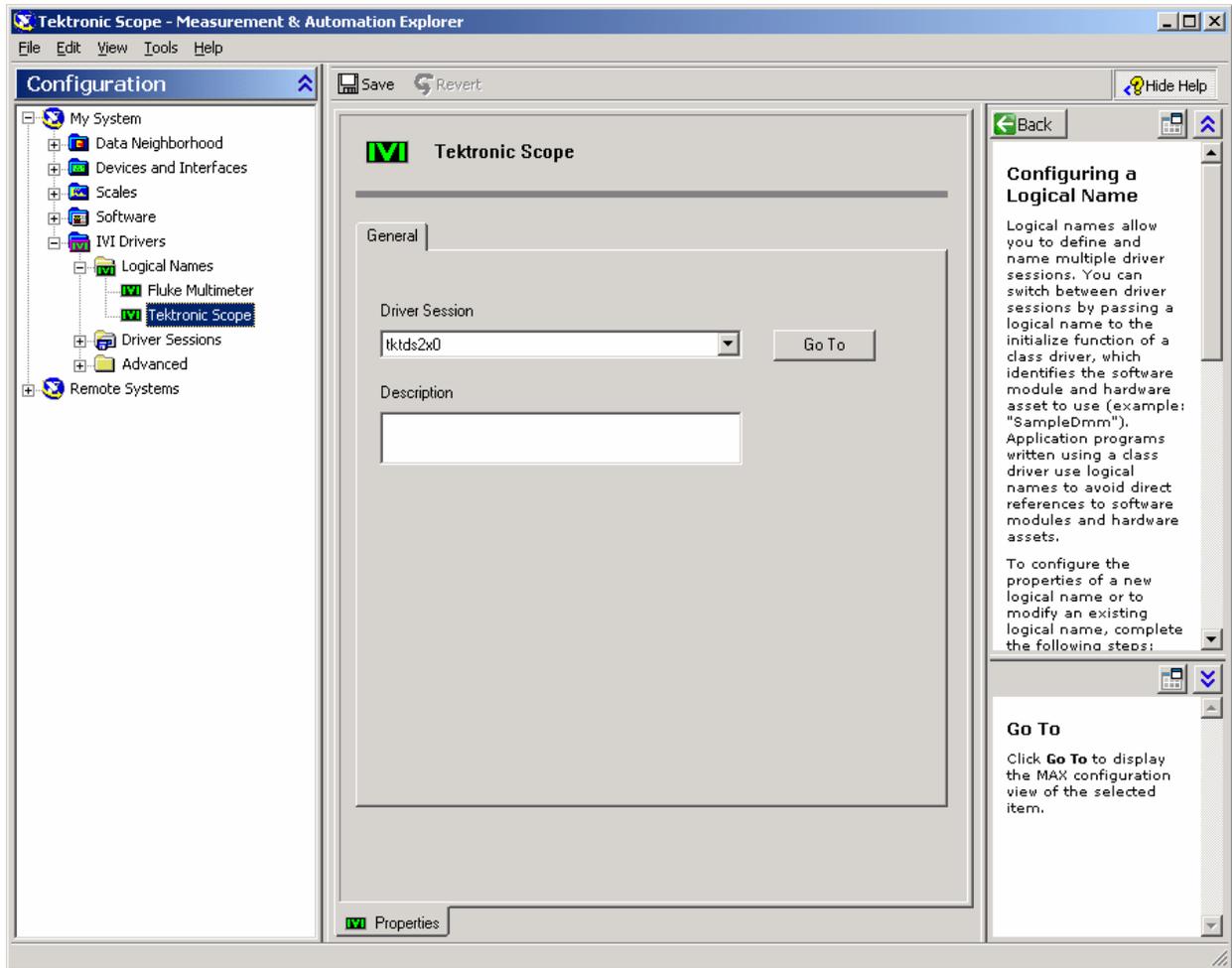


Figure 7. MAX IVI Logical Name configuration

7. On the **General** tab, in the **Simulate With** drop-down box, choose *Don't Simulate*.
8. On the **Hardware** tab, click **<<Create New** and type the name you used in Step 5. Press **Enter** to enable the other two text boxes on this tab.

9. In the **Resource Descriptor** text box, type the VISA descriptor for the device. You can retrieve this information from the VISA Interactive Control utility. It should look something like “GPIB0::5::INSTR”

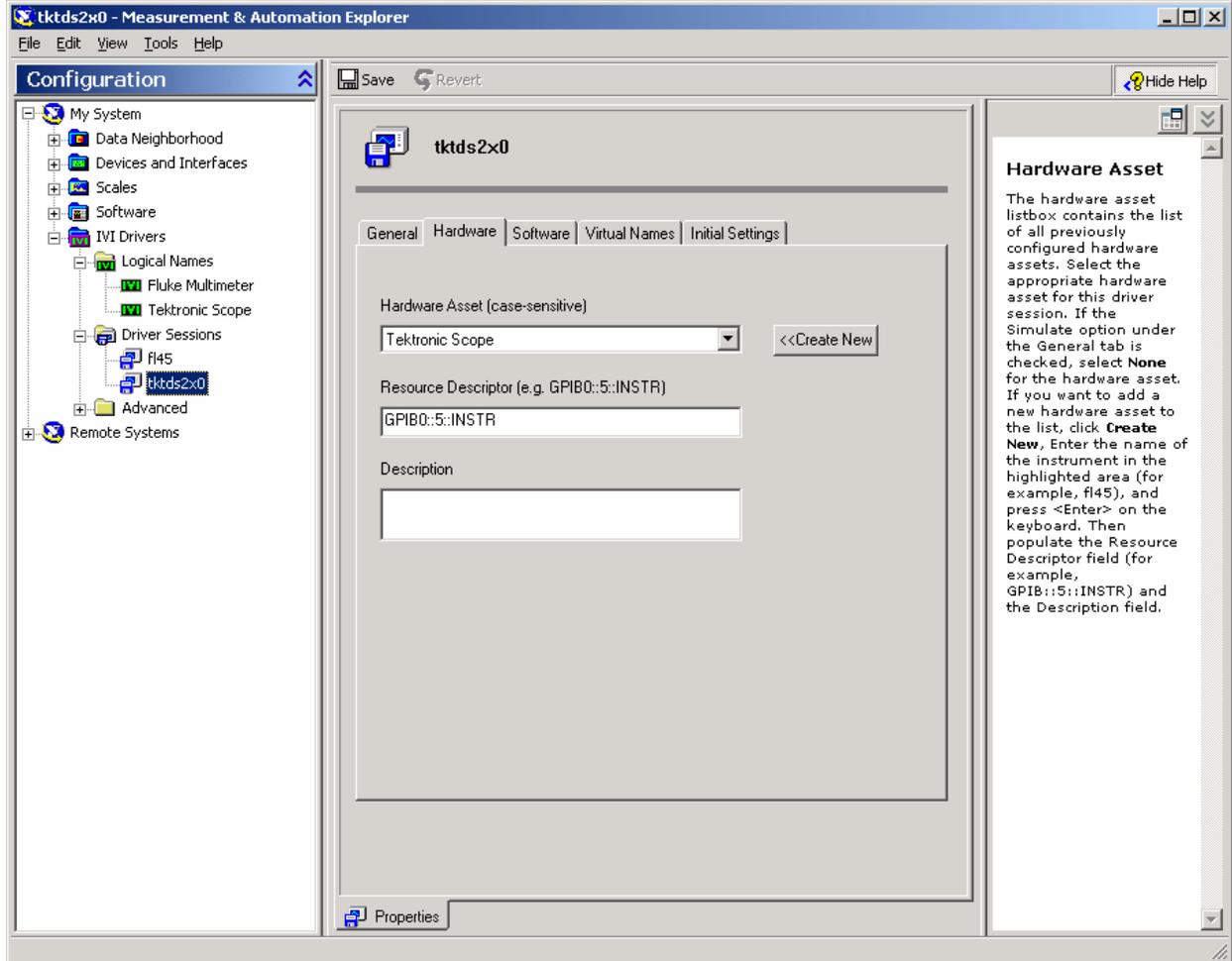


Figure 8. IVI Hardware configuration

10. Click the **Save** button at the top of the pane and exit MAX.

You should now be able to use the logical name you entered to access the instrument. Try one of the examples in the LabVIEW Example Finder under Hardware Input and **Output ► Instrument Drivers & IVI ► General** to test it.

Make sure you choose an example for your instrument's class.