**Driver Documentation**

The agx2k3k.txt file can be found in the <*Program Files*>\IVI Foundation\IVI\Drivers\agx2k3k directory. It contains notes about the driver’s level of compliance with the IVI specifications and lists the optional IVI features supported in the driver.

The agx2k3k.chm file can be found in the <*Program Files*>\IVI Foundation\IVI\Drivers\agx2k3k directory. It contains descriptions of all functions, parameters and their valid values or ranges.

The agx2k3kAttributeInfo.html file can be found in the <*Program Files*>\IVI Foundation\IVI\Drivers\agx2k3k directory. It contains names and descriptions of all attributes and their valid values.

**Driver Source Code and Examples**

The C driver source code and example(s) can be found in the <*Program Files*>\IVI Foundation\IVI\Drivers\agx2k3k directory. For instructions on rebuilding a driver, refer to the following Knowledgebase documents: [Rebuilding an IVI Specific Driver in NI LabWindowsTM/CVITM](http://digital.ni.com/public.nsf/allkb/753FCBD48E366B7C862570A00048AFFF?OpenDocument) and [Rebuilding an IVI Specific Driver in Microsoft® Visual Studio®](http://digital.ni.com/public.nsf/allkb/6E74B81158955BEA86257A1600799CEB?OpenDocument) .

The NI LabVIEWTM wrapper VIs and examples can be found in the <*LabVIEW*>\instr.lib\agx2k3k directory.

**Connecting to the Instrument**

The IVI resources page (<http://ivifoundation.org/resources/default.aspx>) has documents and videos that explain how to get started with an IVI-C driver in different development environments:

IVI Getting Started Guide for LabVIEW

IVI Getting Started Guide for LabWindows/CVI

IVI Getting Started Guide for Microsoft Visual C++

IVI Getting started Guide for MATLAB®

**Configuring Instrument Settings**

An IVI instrument driver implements each readable or writable setting on the instrument, such as the vertical voltage range on an oscilloscope, as an attribute.

IVI instrument drivers export high-level functions that allow you to set the value of multiple attributes in one call. This can be useful when it is necessary to send settings to an instrument in a particular order.

IVI instrument drivers also allow you to modify and get a value of individual attributes. You do this by calling agx2k3k\_GetAttributeVi<*data type*> and agx2k3k\_SetAttributeVi<*data type*> functions. To use these functions correctly you need to know the data type, valid values, and ID of the attribute you want to access. How you do so depends on the development environment you are using.

NI LabWindows/CVI Users

Open the agx2k3k.fp file, expand the Configuration class, and select one of the Set/GetAttribute functions. From the function panel window, click on the **Attribute ID** control to display a dialog box containing a hierarchical list of the available attributes.

If you do not see the attribute you want, click on the **All Data Types** option in the Data Type pane of the dialog box.

NI LabVIEW Users

Use a Property Node to access the specific properties of a driver. A property node can be found in the **Connectivity»ActiveX** functions palette.

Microsoft® Visual Studio® Users

Refer to the agx2k3kAttributeInfo.html file in the <*Program Files*>\IVI Foundation\IVI\Drivers\agx2k3k directory.

**Configuring Driver Settings**

IVI instrument drivers implement inherent capabilities such as simulation, range checking, state caching, coercion recording, interchangeability checking, and instrument status checking. A user can enable/disable these features either programmatically using an attribute or statically using a configuration utility.

**Direct I/O**

IVI instrument drivers for message-based instruments export agx2k3k\_viRead and agx2k3k\_viWrite functions, which enable you to perform direct I/O with the instrument**.**

**Instrument Command Coverage**

IVI instrument drivers for message-based instruments typically implement the full functionality of the instrument available via the commands and queries with a few exceptions.  The agx2k3k.chm file lists the instrument commands that the driver implements for each function and parameter. The same information can also be found in the agx2k3k.fp file.

Some commands and queries are not suitable for an instrument driver. The commands from the following nodes are NOT implemented in this driver:

* *<DIAGnostic>*
* *<FORMat (may be used internally but not exposed to users)>*
* *<SYSTem:COMMunicate>*
* *<Anything dealing with Service or Factory Calibration functionality>*
* *<Undocumented SCPI (factory use only)>*
* *<Other features not normally accessed through the programmatic interface, for example:*
  + *DISPlay*
  + *HARDCopy*
  + *MEMory:STATe*
  + *CURSor>*

Driver users can send any commands to a message-based instrument using the driver’s Direct I/O functions.

**Known Issues**

*None*

**Contacting Support**

If you have feedback or need help using this driver, contact National Instruments at *instrument.drivers@ni.com*.

**Driver Revision History**

Refer to the agx2k3k.c file for the revision history of this driver.

**Trademarks**

CVI, LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments. Refer to the *Trademark Information* at ni.com/trademarks for other [National Instruments trademarks](http://digital.ni.com/express.nsf/bycode/rdtrad).

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

MATLAB® and Instrument Control Toolbox™ are trademarks of The MathWorks, Inc.

Microsoft and Visual Studio are registered trademarks of Microsoft Corporation.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.