CONEX-AGP

Agilis-P Controller with Encoder Feedback

LabVIEW Samples Manual

V2.0.0

Newport®
Experience | Solutions

For Motion, Think Newport™

Table of Contents

CONEX-AGP
Agilis-P Controller with Encoder Feedback

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide instructions on how to use the CONEX-AGP LabVIEW software.

This document shows:

- How to connect with a selected instrument
- How to use a function from the Command Interface library
- How to disconnect the connected instrument.

1.2 Overview

The CONEX-AGP LabVIEW software provides sample VIs that demonstrate how to develop a LabVIEW program for the CONEX-AGP controller.

2.0 Prerequisites

2.1 Install LabVIEW 8.x or higher

LabVIEW version 8.x or higher from National Instruments must be installed on your computer.

2.2 Connect Newport Instrument

Before connecting your Newport instrument, install its application software. This installation will also install the communication driver.

3.0 LabVIEW Software

3.1 Finding the Newport Instrument LabVIEW Software

The default installation folder for the CONEX-AGP software is: "C:\Program Files\Newport\Piezo Motion Control\Newport CONEX-AGP Applet". The LabVIEW software is located in the \Samples folder beneath the default installation folder.

3.2 Instrument Communication

3.2.1 Instrument Initialization

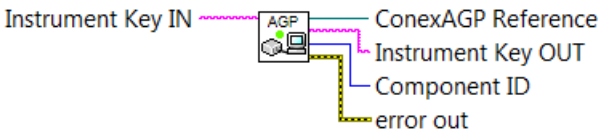
ConexAGP_Open.vi creates an instance of the command library and initializes communication with the selected instrument.

Inputs

- **Instrument Key IN**: represents the serial COM port. If it's empty, a dialog box displays a device list. Then the user can select one instrument from this list.

Outputs

- ConexAGP **Reference**: represents the command library instance.
- **Instrument Key OUT**: represents the selected serial COM port.
- **Error**: communication I/O error (0 = success)
- **error out:** contains LabVIEW error information.



3.2.2 Instrument Shutdown

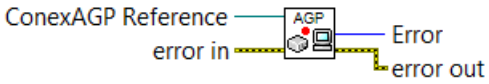
ConexAGP_Close.vi is used to close communication with the selected instrument.

Inputs

- ConexAGP **Reference**: represents the instrument instance.
- **error in:** describes LabVIEW error conditions that occur before this node runs.

Outputs

- **Error**: communication I/O error (0 = success)
- **error out:** contains LabVIEW error information.



3.2.3 Instrument Commands

The LabVIEW Invoke Node is used to call a method in the CommandInterface DLL. The commands available are described in the Command Interface Manual.

The command syntax is "AA". The command name is the beginning of the function name. That allows the user to refer to the controller's manual to get the description of the command to select.

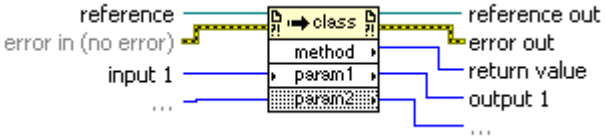
 **Invoke Node** (from the Connectivity → .NET function panel provided by LabVIEW)

Inputs

- **reference** is the refnum associated with the instrument object on which you want to invoke a method or perform an action.
- **error in** describes LabVIEW error conditions that occur before this node runs.
- **input 1..n** are example input parameters of a method.

Outputs

- **reference out** returns **reference** unchanged.
- **error out** contains LabVIEW error information.
- **return value** is an example return value of a method.
- **output 1..n** are example output parameters of a method.



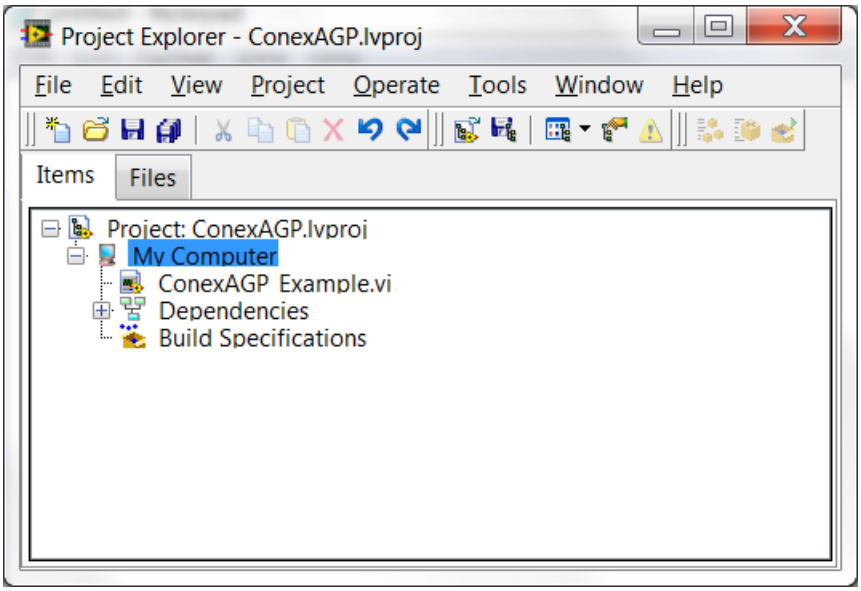
4.0 LabVIEW Example

4.1 LabVIEW Project Creation

Create a labVIEW project (refer to the National Instrument manuals for more information on how to create a LabVIEW project). It's a necessary step to develop with the Newport LabVIEW software. The labVIEW development must be done from this opened project.

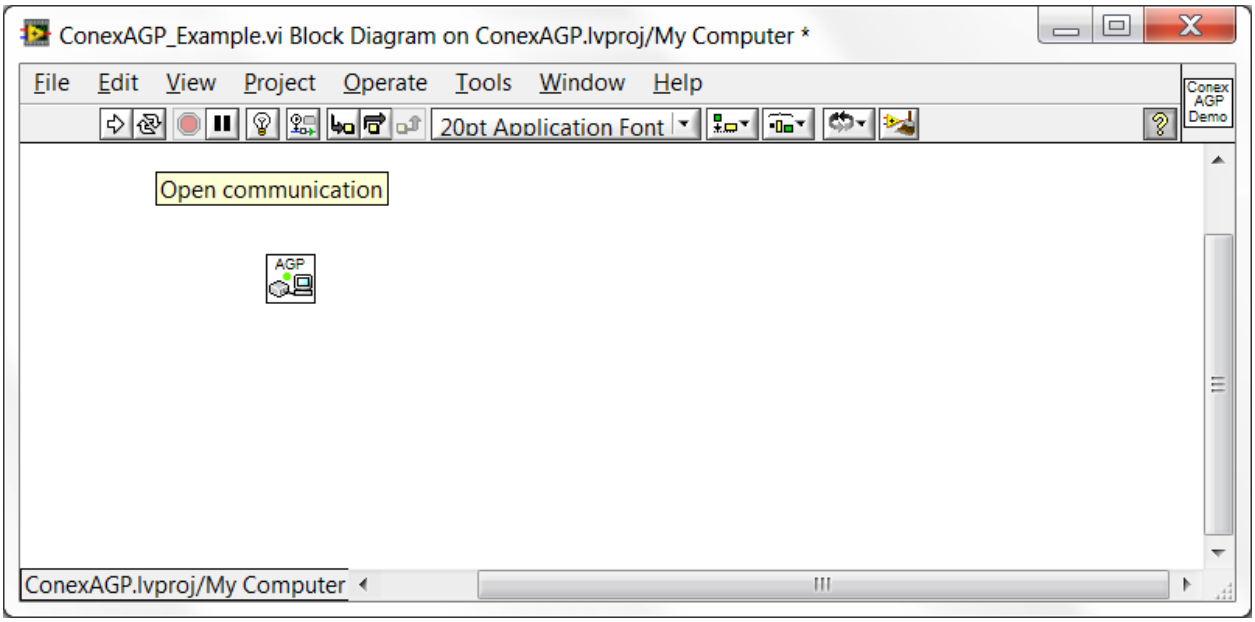
NOTE:

Project must be opened when you do the LabVIEW coding work.



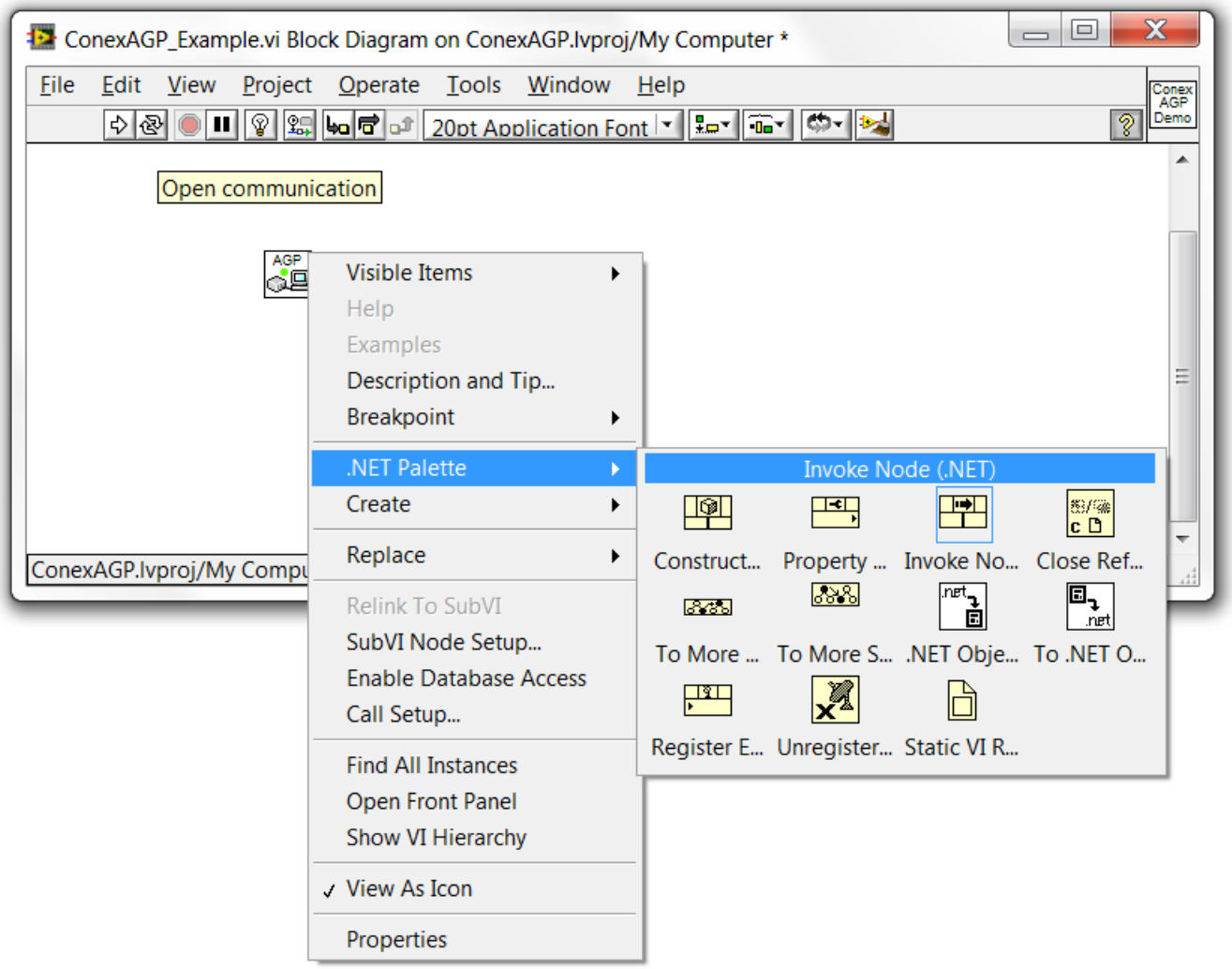
4.2 First Step: Initialize Instrument Communication

In your LabVIEW project, create a New VI and show the Block Diagram Window. Right click on the Block Diagram Window to open the "Functions" panel and choose "Select a VI…" Then select "ConexAGP_Open.vi" to create a VI call that initializes communication with the instrument.

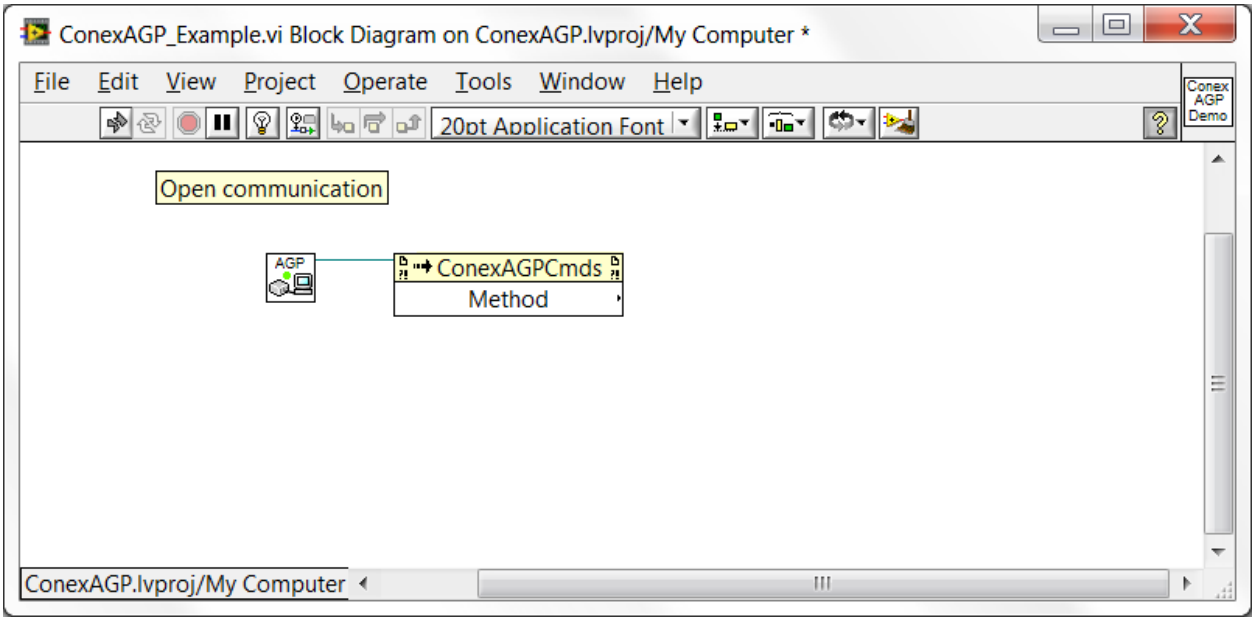


4.3 Second Step: Invoke an Instrument Command

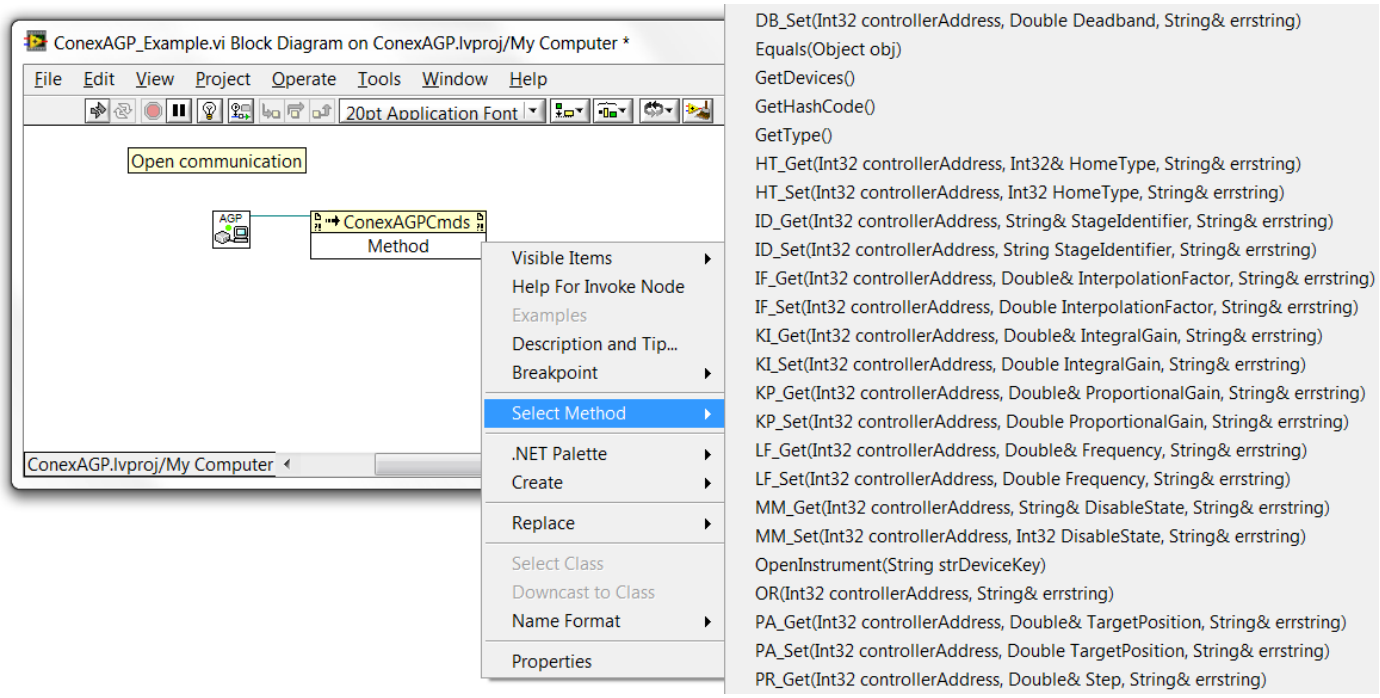
Right click the upper right wire terminal of the ConexAGP_Open.vi to display a context menu. From the context menu, select ".NET Palette" and then select the "Invoke Node" block.



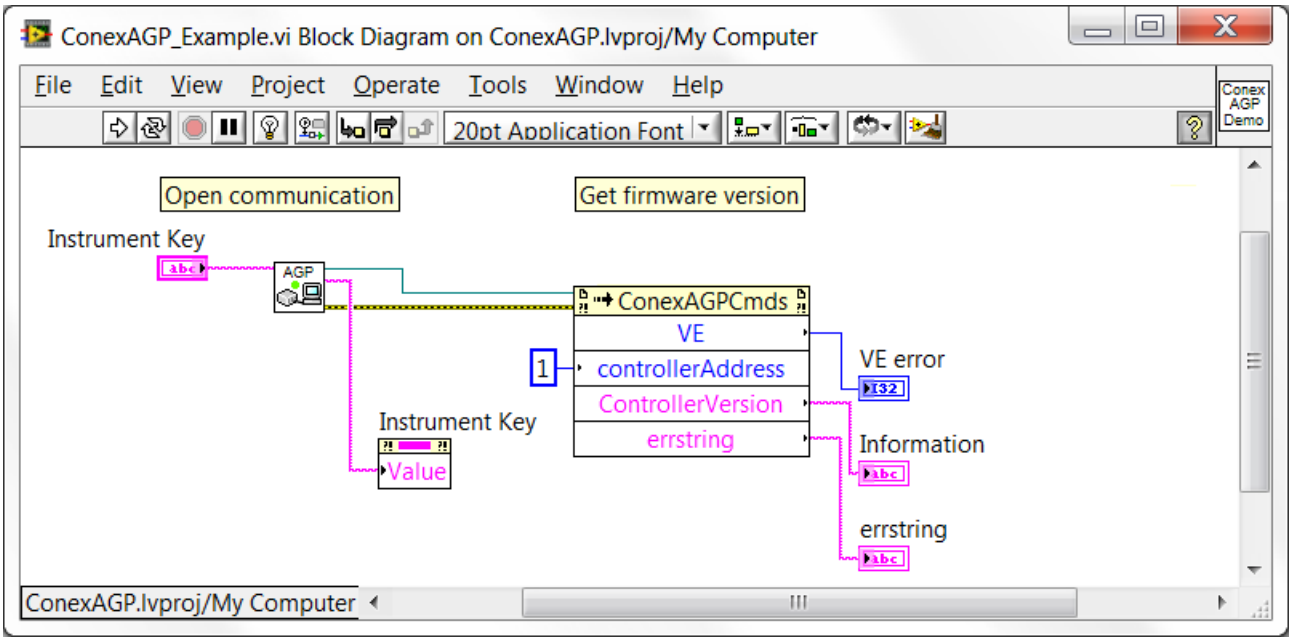
Draw a wire from the upper right wire terminal of the ConexAGP_Open.vi to the upper left wire terminal of the Invoke Node.



Right click the word “Method” in the “Invoke Node” block to display a context menu. From the context menu, select “Select Method” and then choose the method from the command library (ConexAGPCmdLib.dll) that you wish to call.

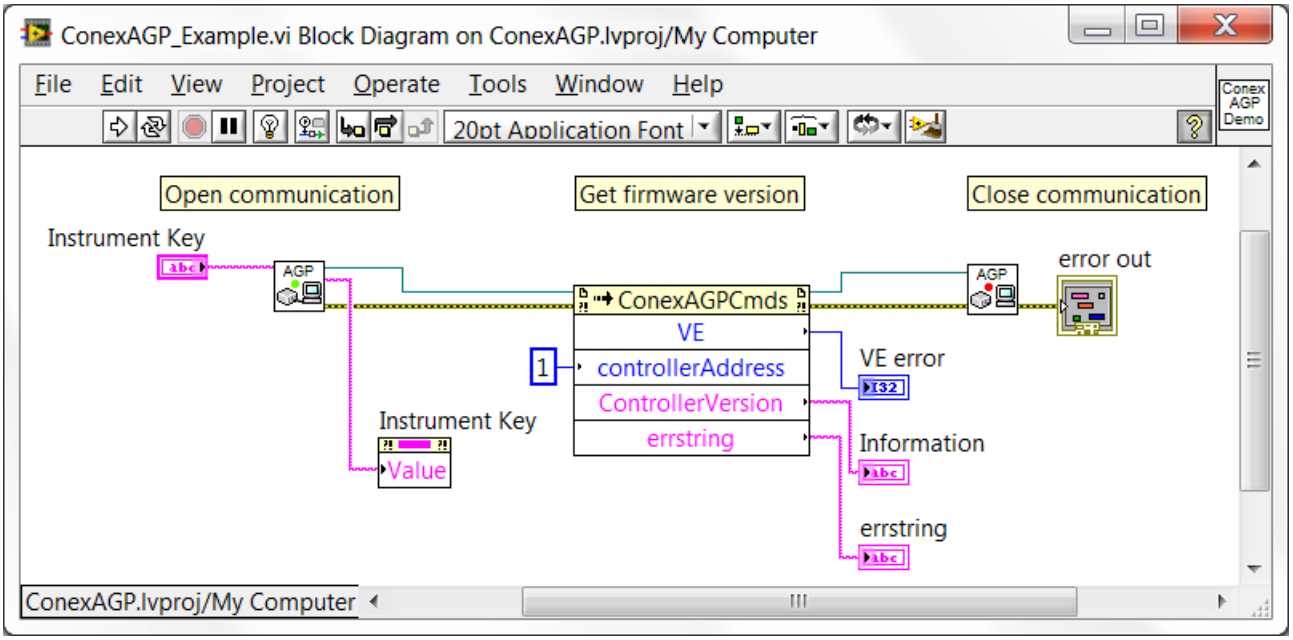


In this example, the selected function is “VE” to get the version of the selected instrument.



4.4 Last Step: Shutdown Instrument Communication

Right click on the Block Diagram Window to open the "Functions" panel and choose "Select a VI…" Then select "ConexAGP_Close.vi" to create a VI call that shuts down communication with the instrument.



5.0 Knowledge from National Instruments

From LabVIEW 2010 Help
Edition Date: June 2010
Part Number: 371361G-01

http://zone.ni.com/reference/en-XX/help/371361G-01/lvconcepts/loading_assemblies/

5.1 Loading .NET Assemblies in LabVIEW

If you reference a .NET object from the front panel or block diagram of a VI, ensure that LabVIEW can load the .NET assembly for that object. The Common Language Runtime (CLR) is responsible for locating .NET assemblies that you call. Refer to the Microsoft Developer Network (MSDN) Web site for more information about how the CLR locates assemblies. If the CLR cannot find the assembly, LabVIEW then searches for the assembly in the same manner it searches for missing VIs. LabVIEW searches for missing VIs in the directories you specify on the Paths page of the Options dialog box. If LabVIEW cannot find the .NET assembly for a .NET object referenced directly on the front panel or block diagram, LabVIEW generates a load-time error. If LabVIEW cannot load a dependent assembly needed during run-time, LabVIEW generates a run-time error.

The CLR uses the directory of the running executable as the default search path when it loads private .NET assemblies. If you reference a .NET object from a VI that does not belong to a LabVIEW project, the CLR considers LabVIEW.exe to be the running executable. The CLR therefore searches for private assemblies in the directory in which the LabVIEW.exe file is located. **If you reference a .NET object from a VI that does belong to a LabVIEW project, the CLR considers the project to be the running executable. The CLR therefore searches for private assemblies in the project directory.** If you reference a .NET assembly from a VI and the assembly does not belong to the .NET Framework, National Instruments strongly recommends that you store the VI in a project to avoid having to place files in the directory in which the LabVIEW.exe file is located.

If you call a .NET assembly from a VI that does not belong to a project, you technically can save the assembly in the same directory as its calling VI. LabVIEW searches certain VI directories, including the calling VI directory, for assemblies that the CLR cannot load by default. However, calling assemblies stored in this location can result in name conflicts and other unexpected .NET behavior. Therefore, National Instruments does not recommend that you save assemblies in this location.

5.2 Loading VIs with an Updated Assembly

Microsoft Visual Studio .NET and other development tools provided in the .NET Framework SDK can assign strong names to an assembly. Assemblies with the same strong name are expected to be identical.

When you load a VI with a change in the path of a .NET assembly or with a change in the version number or culture string of a strong-named assembly, LabVIEW launches a warning dialog box informing you of the change. Once loaded, the VI includes an asterisk in its title bar and in the list of open VIs displayed in the Window menu. When you save the VI, the asterisk disappears until you make a new change.

When you load a VI with a change in the time stamp of a .NET assembly, LabVIEW does not launch a warning dialog box but does display an asterisk in the title bar of the VI.

Refer to the KnowledgeBase at ni.com for more information about how to select and load specific versions of .NET Assemblies.

Service Form

Your Local Representative

Tel.: _____

Fax:_____

Name: _____

Company:_____

Address:_____

Country:_____

P.O. Number:_____

Item(s) Being Returned:_____

Model#: _____

Return authorization #: _____
(Please obtain prior to return of item)

Date: _____

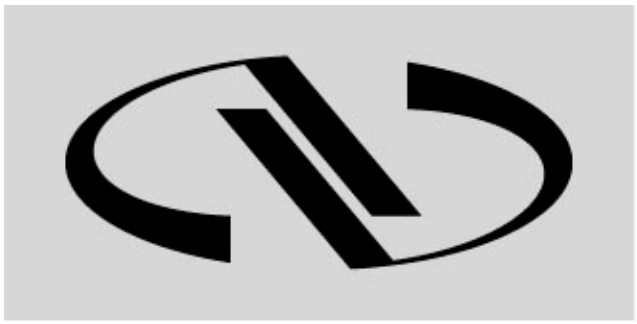
Phone Number: _____

Fax Number: _____

Serial #: _____

Description: _____

Reasons of return of goods (please list any specific problems): _____



Newport®

Experience | Solutions

Visit Newport Online at:
www.newport.com

North America & Asia
Newport Corporation
1791 Deere Ave.
Irvine, CA 92606, USA

Sales
Tel.: (800) 222-6440
e-mail: sales@newport.com

Technical Support
Tel.: (800) 222-6440
e-mail: tech@newport.com

Service, RMAs & Returns
Tel.: (800) 222-6440
e-mail: rma.service@newport.com

Europe
MICRO-CONTROLE Spectra-Physics S.A.S
9, rue du Bois Sauvage
91055 Évry CEDEX
France

Sales
Tel.: +33 (0)1.60.91.68.68
e-mail: france@newport-fr.com

Technical Support
e-mail: tech_europe@newport.com

Service & Returns
Tel.: +33 (0)2.38.40.51.55