Tech Tip: BeamGage Profiling with .Net Automation Interface and LabVIEW®

BeamGage Professional and BeamGage Enterprise version 5.7 are supported with Automation via .Net components. Both include a LabVIEW example that can be run with the LabVIEW Run-Time Engine that is provided with the BeamGage software CD or available for free download from National Instruments.

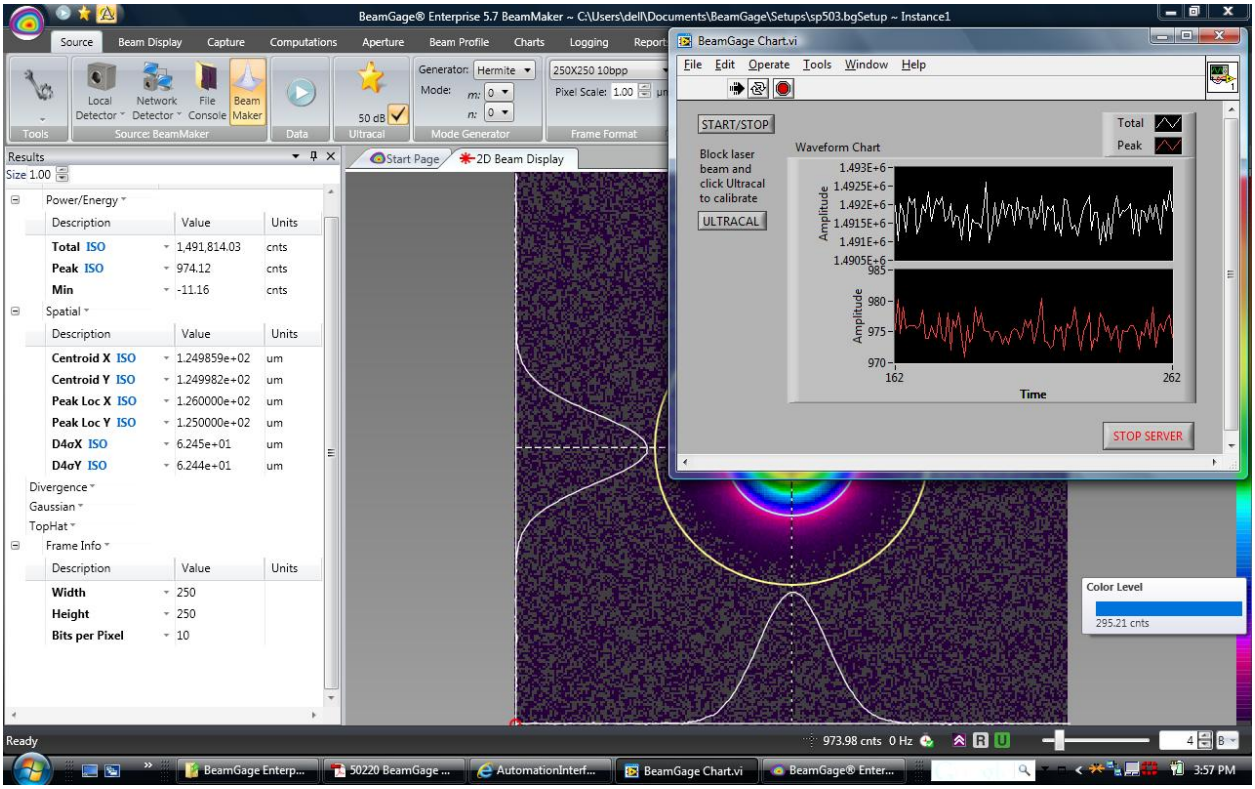


Figure 1. BeamGage and LabVIEW.

Ophir Photonics Group
3050 North 300 West
North Logan, UT 84341
Tel: 435-753-3729
www.ophiropt.com/photonics

The attached pages from the BeamGage User Guide include information for downloading and installing the LabVIEW Run-Time Engine and accessing and starting the example.

To run the example you will need to have BeamGage Professional or BeamGage Enterprise version 5.7 installed along with the LabVIEW Run-Time Engine.

You will need to browse to the actual installed BeamGage User Guide to use the links. Once the BeamGage software is installed, it is available at; C:\Program Files\Spiricon\BeamGage Enterprise or Professional and titled 50220 BeamGage User Guide.pdf

To run the example there are additional steps required. You will need to create a .NET exe.config file and place it in the LabVIEW directory. i.e. C:\Program Files\National Instruments\LabVIEW 2010\LabVIEW.exe.config

From the Automation documentation:

"This version of BeamGage has been built to target the .NET 4.0 CLR. By default LabVIEW does not know how to handle .NET 4.0 assemblies. To make LabVIEW work with .NET 4.0 compiled assemblies, you will need to place a .NET config file next to the LabVIEW.exe; to tell it to load the .NET 4.0 CLR (instead of the default CLR). The easiest way to do this is to copy and rename the file "BGTotalPeakExample_XXX.exe.config" found in the BeamGage Automation examples directory to "LabVIEW.exe.config" and then move the new file to the same directory where LabVIEW.exe is installed. LabVIEW 2009 is installed to the path below by default:"

Please see the following NI link for more details.
http://digital.ni.com/public.nsf/allkb/32B0BA28A72AA87D8625782600737DE9

Ophir Photonics Group
http://www.ophiropt.com/photonics

Ophir Photonics Group
3050 North 300 West
North Logan, UT 84341
Tel: 435-753-3729
www.ophiropt.com/photonics

8.2 Evolution of Spiricon's Automation Interface

Spiricon's older products used COM and ActiveX to provide an automation interface. More recent technologies, like .NET, provide more fully featured functionality. Recent developments in remoting technologies allow nearly transparent interaction between machines on the same domain. This allows the user to leverage more than one machine while using BeamGage for analysis. COM remote operation is more difficult to use and setup when compared to its .NET counterpart. For this reason, BeamGage's automation interface was developed using Microsoft's .NET infrastructure. Any .NET application should easily integrate, and be able to interact with the core functionalities provided by BeamGage.

8.3 Introduction

The BeamGage automation interface was designed to achieve two main goals. First, to allow the BeamGage user to programmatically do what they could otherwise do via the graphical user interface (GUI). Second, to expose stable interfaces to the user that will not change, causing breaks to their dependant code. In order to facilitate these goals, it is important that the user be given stable abstractions to program against. It is likewise important to allow BeamGage to evolve as new features are added. Spiricon is dedicated to protect users from changes in underlying implementation as BeamGage evolves. To this end BeamGage's automation interface is presented as a set of interfaces that collectively expose the functionality of the application. Access to these various interfaces is provided by creating one concrete class known as BeamGageAutomation. From BeamGageAutomation the user can create and destroy several instances of the BeamGage application.

BeamGageAutomation exposes a method (via the IV5AppServer interface it implements) called GetV5AppServer. This method will launch BeamGage instances and return a reference to an IAutomationCLT object (CLT stands for Core Logic Trunk). Each instance of BeamGage can be independently controlled via its respective core logic trunk. The number of active BeamGage applications available is controlled via BeamGageAutomation while, control of a single BeamGage instance is provided via its core logic trunk.

The IAutomationCLT exposes a method called GetInterfaceX; this method is used to access all API interfaces. Each interface is accessed via its name. Once the desired interface is acquired, the user can control its respective functionality for the given instance of BeamGage.

8.4 Documentation

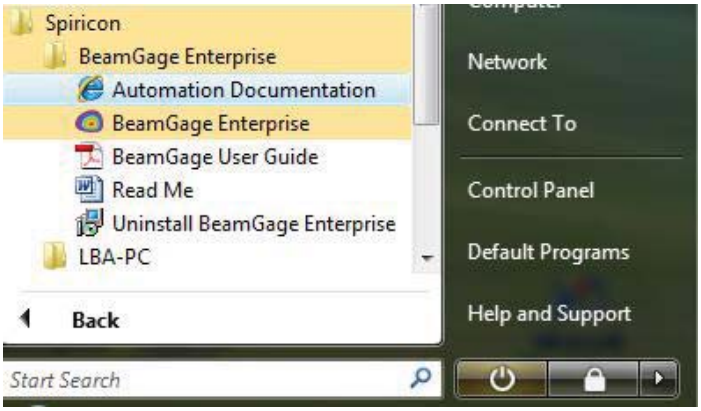
The Automation API reference is presented via html. The html reference provides cross-referenced access to all interfaces and functionality provided for automation application development. The **BeamGage Automation API** may be accessed via the following link:

Automation Documentation

-or-

Via the start menu shortcut as shown below:

Start, **All Programs**, **Spiricon**, **Automation Documentation**



8.4.1 Examples

Examples of simple automation applications in LabView and Visual Basic .NET are provided. For step-by-step walkthroughs click on the appropriate link:

Setup (LabView)
Setup (VB)

Also provided is an executable version of a LabView example. The LabView executable requires that you have the appropriate LabView run-time engine installed on your PC.

Note: If you already have LabView 2009 w/SP1 and f1TREpatch installed then the run-time engine will already be present.

8.4.2 LabView Run-Time Engine

The appropriate run-time engine must be installed in order to run the supplied LabView example. If LabView and its run-time engine are not already installed on your PC, then one will need to be installed. A copy of the LabView run-time engine is provided on the BeamGage CD. To install the runtime engine use **Windows Explorer** to view the folders on the Spiricon supplied CD. Open the folder shown below and then start the run-time installer.

<CD Drive>:\Resources\Software\<number> BeamGage <edition> 5.x (5.5 or higher)

LV2009SP1f1RTEpatch.exe (LabVIEW 2009 SP1 Run-Time Engine (Standard) (32-bit) for Windows)

Follow the instructions that appear on the screen to complete the installation.

Note: You must have Administrator privileges in order to install the above run-time engine and run the following executable example.

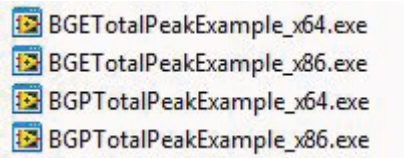
Important: ***If you downloaded BeamGage v5.5 or later from our web site, you will not receive the above LabView run-time files. In this case you will need to go to the LabView web sight and manually download and install them.***

8.4.3 To Run the LabView Example

Go to the following folder to locate the LabView executable example:

C:\Program Files\Spiricon\BeamGage<Edition>\Automation\Examples\LabVIEW2009\TotalPeakChartExample\

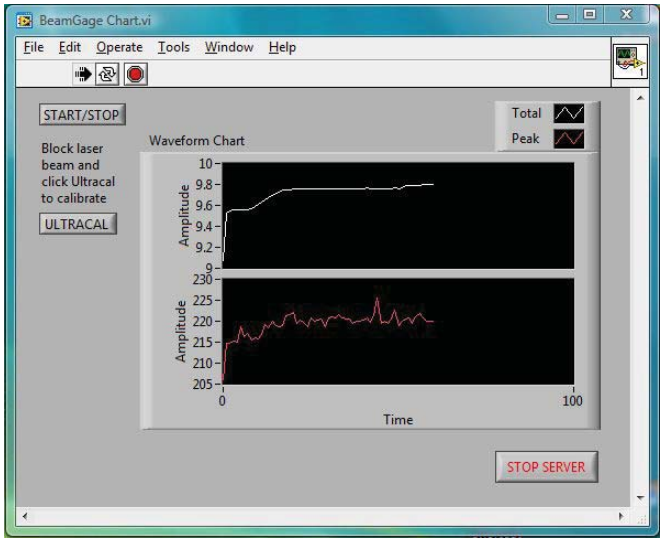
One of the following LabView executables will be present:



Run the one that is provided to start the LabView example.

8.4.4 Operating the LabView Example

The **TotalPeakExample** will open with the following LabView VI:



… and it will open, connect to, and run BeamGage. When it opens BeamGage, BeamGage will start with the setup file last saved in BeamGage. If that setup connects to an available source, the Total Power and Peak Fluence values will plot in the displayed charts.

The **START/STOP** button will cause BeamGage to start and stop collecting data. The **ULTRACAL** button will cause BeamGage to perform an **Ultracal** cycle.

To close the above LabView example and BeamGage you must, in the following order:

1. Click on the **Stop Server** button to close BeamGage.
2. Close the VI.

Important: ***BeamGage must be closed using the Stop Server button. It can not be closed normally once this VI is closed. If you close the VI without first closing BeamGage with the Stop Server button, then the only way you will be able to close BeamGage will be by rebooting the PC.***

Important: ***When BeamGage is launched from an automation client, it can only be properly closed by that same client. Thus it must be closed before the client is closed. Keep this in mind when designing your own client application.***

APPENDIX A ISO Computations Table

This table of ISO computations follows the labeling and notions found in the ISO standards. Where differences exist within the standards no particular preference is given to any one notation. The information contained here may not always agree with the latest releases of the individual standards, however Spiricon will update it from time to time as the ISO standards evolve. The Section numbers in the list may also break when ISO standards are reformatted. The ordering of the items follow no particular sequence and no importance should be attached to the item #.

Note: The symbols and names listed here may or may not agree with the notations used in BeamGage. These notations may be slightly modified for the purpose of maintaining consistency in labeling within BeamGage and standard industry and legacy usage. Not all listed results are currently provided in BeamGage.

( ) shall contain one of the following symbols [1]
u = a measurement based upon a percentage of power/energy, usually an encircled amount or contained by a smallest slit . The value u shall be replaced with an amount, such as 90, to mean 90%.
σ = a measurement based on second moment definitions
k = a measurement based on a Knife Edge measurement, usually a 10/90 percent of energy [2]

Item #	Symbol	Name	ISO 11145	ISO 11146-	ISO 11670	ISO 13694	ISO 15367-	Section	Definition
1	$A_{()}$ $A_{\sigma}(z)$	beam cross-sectional area	x			x		3.2.1 3.2.2 3.2.6	Area bounded by the beam width definition ( ) for circular beams $A_{\sigma} = \pi \frac{d_{\sigma}^2}{4}$ for elliptical beams $A_{\sigma} = \pi \frac{d_{\sigma x} d_{\sigma y}}{4}$
2	$d_{()}$	beam diameter	x	1				3.3.1 3.3.2 3.3	Smallest aperture or second moment diameter of a circular beam. beam diameter (second moment) $d_{\sigma}(z) = 2\sqrt{2\sigma(z)}$ where :