

Dll's and Interfacing for DataRay Products

1. What you need to know

DLL stands for Dynamic Linked Library. These instructions assume that you are a software *programmer*, familiar with the use of DLLs and/or LabView. If you do not qualify, then find somebody who does.

We cannot teach you how to use Dlls. We will do our best to assist you in using our Dlls.

The normal use of a DLL is to allow other applications the ability to reuse a piece of software WITHOUT requiring the **user** to understand how the functions are implemented, and also without **giving** anyone the source code. Another advantage to using DLLs is that the user's program does not need to be updated (recompiled) when the DLL is updated, so that upgrades to the DLL are implemented simply by replacing the DLL.

2. DLL related files

The following files are installed in **c:\Program Files\Dataray** by **iDataRay.exe** versions 3.05T and later:

Name	Size	Type
DataRayData.txt	2 KB	Text Document
DataRayFormats.doc	24 KB	Microsoft Word Doc...
DataRayWin2K.inf	1 KB	Setup Information
DOWNLD.LOG	1 KB	Text Document
Dri_dll.cpp	40 KB	CPP File
DRI_DLL.dll	392 KB	Application Extension
DRI_HEADER.h	9 KB	H File

If you cannot find these files in Windows Explorer, check under **View** or **Tools** for **Folder Options ...**, choose the **View** tab and verify that **Hide file extensions for known file types** and **Hide protected operating system files [recommended]** are unchecked. (Windows Explorer often checks them as the default setting.)

3. File printouts

Because file contents change as the software is upgraded, file printouts are not given here.

You may open and print the files using WordPad, found as **Start, Programs, Accessories, WordPad** and choosing **Files of type: , All Documents(*.*)**. Then find these files in the **c:\Program Files\DataRay** directory, open them and print them out as required.

4. File Contents & Application

Whilst the DataRay software is written in **C++**, the DLLs are written strictly in **C**.

All installed versions are release versions for programmers not running a Microsoft debugger. [**DebugDll** versions are available upon request to programmers running a Microsoft debugger and attempting to debug the use of these DLLs. The debugger will point to the specific line where the problem arises.]

DRI_DLL.dll is the minimal file to install for extracting data. [**DRI_DLL.lib** is available upon request. DLL stands for Dynamic Linked Libraries, but if you are working in C and wish to statically link the library, then this is what you use to do it. i.e. You make **DRI_DLL.lib** part of your program, thus there is no need to link it later.]

DRI_DLL.cpp is the C++ source code of the dll. It allows you to see how a specific call is made.

DRI_HEADER.h is the main header file and declares the ‘exposed’ module parameters for the DLL. i.e. things that the outside world can do. It defined the explicit exportable parameters. It defines how a Parameter is set and declares how a Function (i.e. a sub-routine) is called. It returns the result of the call in double precision.

5. Sample Visual Basic Software

The following Visual Basic sample directories are installed in **c:\Program Files\Dataray** by **iDataRay.exe** versions 3.05T and later:

Name	Size	Type
DataRayBasic.frx	1 KB	FRX File
Declares.bas	7 KB	BAS File
JobFile.job	20 KB	Task Scheduler Tas..
WinCamD_Basic.exe	36 KB	Application
WinCamDBasic.frm	15 KB	FRM File
WinCamDBasic.frx	1 KB	FRX File
WinCamDBasic.vbp	1 KB	VBP File
WinCamDBasic.vbw	1 KB	VBW File

Name	Size	Type
BeamMap.frm	13 KB	FRM File
BeamMap.frx	1 KB	FRX File
BeamMap_Basic.exe	36 KB	Application
BeamMapBasic.vbp	1 KB	VBP File
BeamMapBasic.vbw	1 KB	VBW File
Declares.bas	8 KB	BAS File
JobFile.job	20 KB	Task Scheduler Tas..

The **WinCamD_VBasic.exe** and **BeamMap_Basic.exe** files may be run when the normal **dataray.exe** software is not running. Double-click the one appropriate to your product to see how it works. **Init** or **Initialize** it and then **Start** it to see data collection actually working.

In these directories:

***.frm** contains the sample code which runs under the ***.exe**.

Declares.bas is what the dll calls are referenced from.

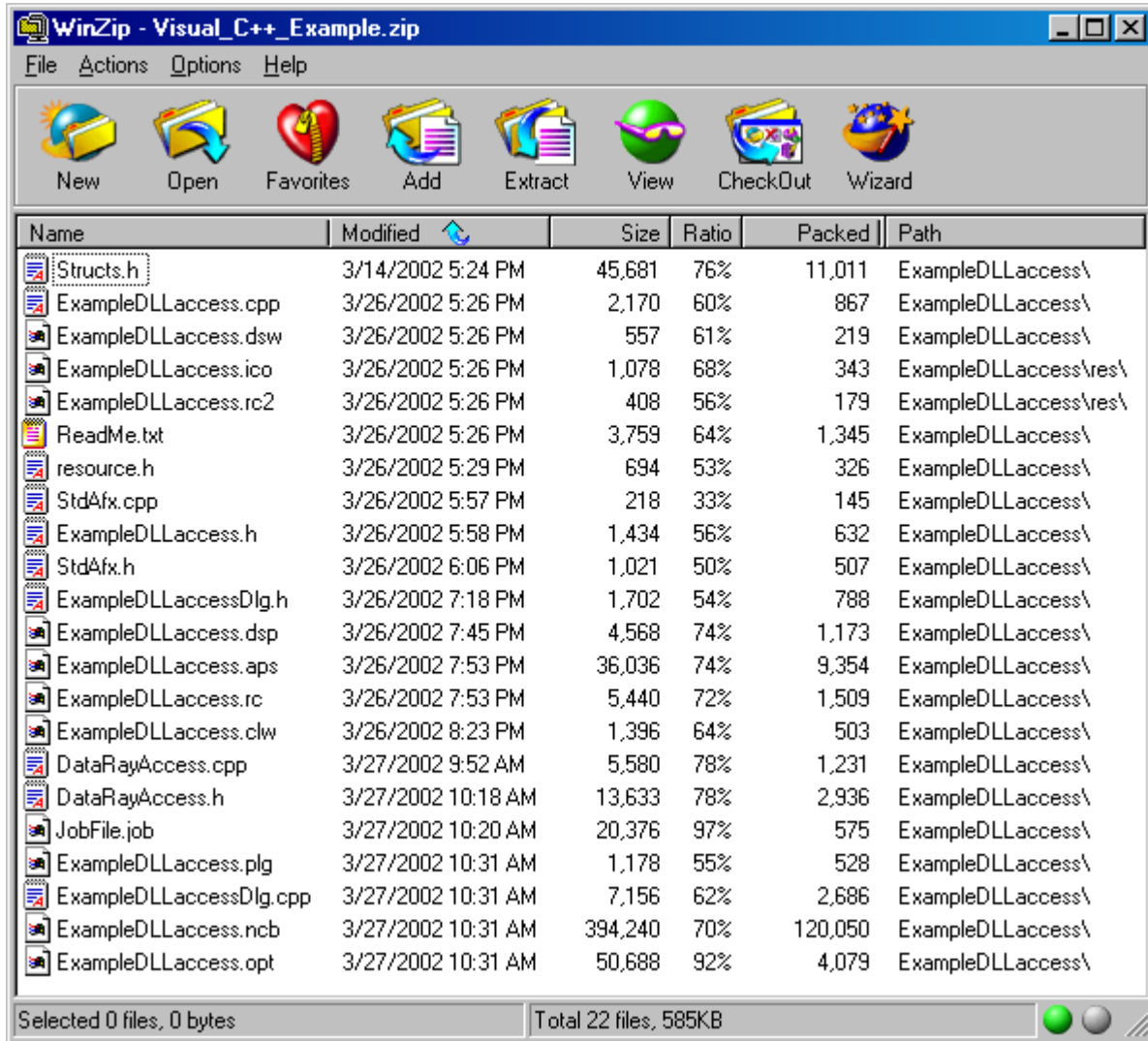
Jobfile.job is a job files created when you open the dll.

***.vbp** and ***.vbw** are the Visual Basic project files.

***.frx** is a file created when the ***.exe** is run.

6. Sample Visual C++ Software

The following Visual C++ files are downloadable from the website as: **Visual_C++_Example.zip**. It contains the following files:



You must be running visual C++ in your PC. It is suggested that you unzip these files into a directory called **C++ Example** within the **c:\Program Files\DataRay** directory, but you may choose to put them somewhere else that is convenient to you.

This set of files allows you to open up a Project in visual C++, which is a standalone program that can access the dll.

Double click **ExampleDLLaccess.dsw** to initialize WinCamD. This has all the functions that you need to create an object that you can bring into your program.

This Visual C++ example is for WinCamD, but the same principles apply to BeamMap, Beam'R, BeamMap ColliMate and BeamScope.

7. Working with LabView

Within LabView: **DRI_HEADER.h** has to be either in the system directory, normally **c:\windows/system**, **c:\winnt/system**, or in some other path that you point to.

Use the **Create dll function** wizard and set up a 'Prototype' for each 'Call' on the dll.

Find the 'prototypes'. For each parameter in the dll, you must install a 'bridge' for each 'call'.

Four calls are essential.

- 'Install BeamMap', 'BeamR', 'BeamMapC', 'BeamScope' or 'WinCamD'. This finds the head, loads the information from the EPROM head, and then returns a '1' if successful.
- Get 'Parameter'
- 'Index to slit, width' [for BeamMap, Beam'R, BeamMapC and BeamScope products only]
- Set 'Parameter'.



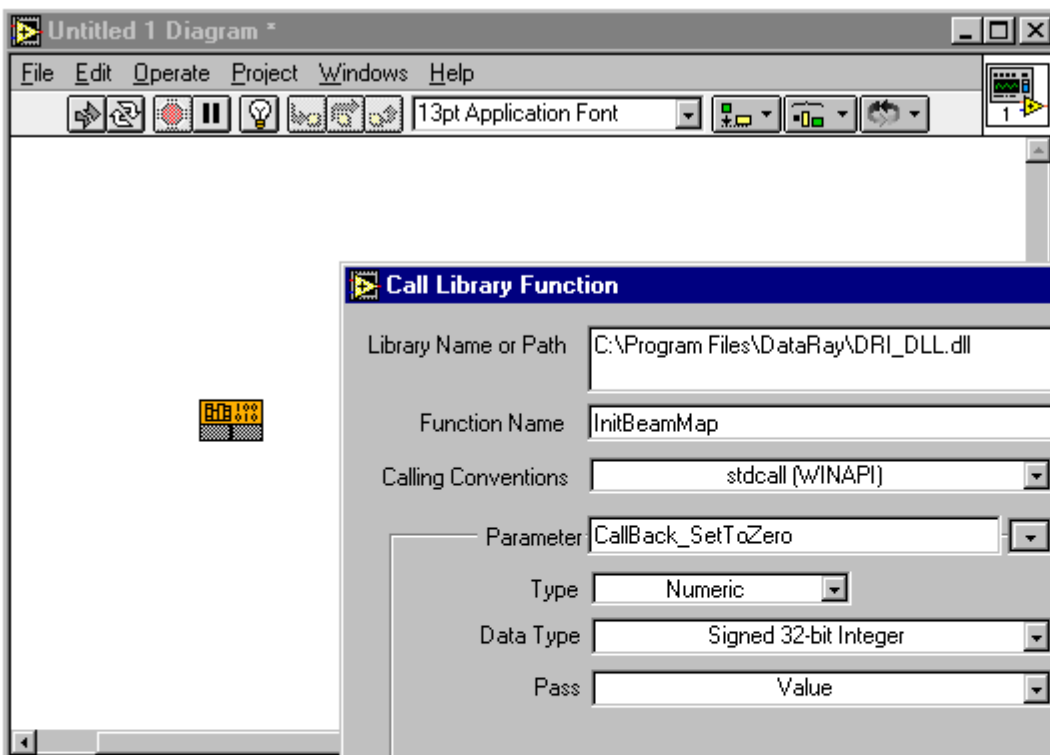


To initialize the instrument (BeamMap example is shown)

Select: **Functions...Advanced**

Then enter definitions as shown below.

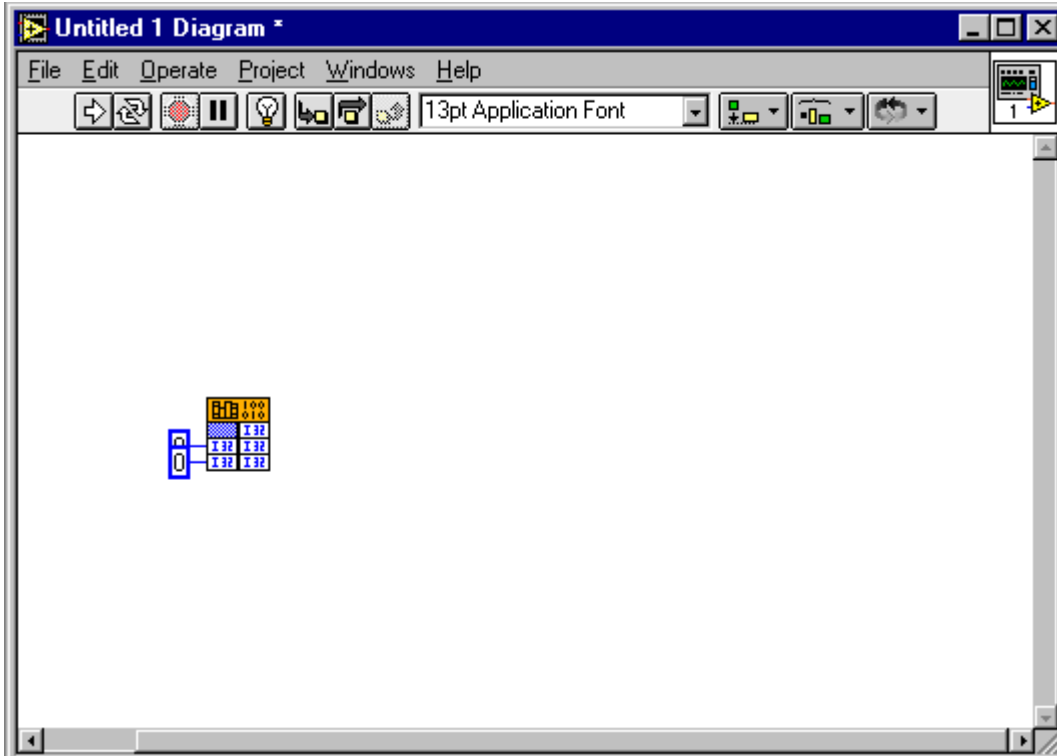
Note that **stdcall (WINAPI)** is required.



Set the two inputs to zero, now this call will initialize the BeamMap

All other functions are similarly defined.

Organizing all the calls into a specific LabView application is beyond our ability to support. However, National Instruments or one of their consultants will be able to help.



8. Problems?

Contact: support@dataray.com or your local distributor.