

Interfacing DataRayOcx to Excel™ VBA

Excel is a registered trademark of Microsoft Corp.

1. Applies to:

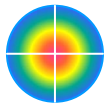
- BeamMap Series
- BeamScope-P7
- WincamD Series
- Versions 5.00B and higher of **iDataRayOcx.exe**. Does not apply to lower software revisions.

2. You need to know that:

- a) If you print this document, use a color printer or much information will be lost.
- b) Learn to use the DataRay product with the DataRay software before trying to use a VBA interface. If you understand the product, the VBA will make sense. If you do not, the VBA may not make sense. A few hours invested in understanding how to use the product, will pay off many times when setting up a VBA interface.
- c) You may save time by reading this document sequentially, rather than by jumping to the section that you believe applies.
- d) This interfacing document applies only to Excel 2000 (Version 9.0) and higher versions. We do not support earlier versions of Excel. If you try to make the OCX work with earlier Excel versions, we wish you the best of luck, but if you encounter Excel down-revision related problems, then you are on your own, as in completely on your own.
- e) **iDataRayOcx.exe** must be installed before you attempt to link to Excel.
- f) OCX architecture should eventually allow the DataRay software to simultaneously run on-screen & to simultaneously interface to other software. For now, run the OCX interface without running the screen version of the DataRay software. With WinCamD, the DataRay software may be open but not running.
- g) VBA stands for Visual Basic Application. It is a cut down version of Visual Basic designed to work with Excel. It includes some additional spreadsheet related functions, but lacks many Visual Basic functions. That said, we recommend that if you familiar with Microsoft Visual Basic, that you have it installed also. It may help.
- h) We are experts in our software. We are not Excel VBA interfacing experts. To use **iDataRayOcx** with Excel VBA, you must at least be competent in Excel, and preferably with Excel VBA, *before* you attempt to use this Application Note.

See Appendix A for a list of books on VBA. If you try to learn VBA from Excel's **Help** file, you will probably fail - the fault of the Excel VBA Help file rather than any personal deficiency on your part.

DataRayOCxExample.xls, included with the software, contains an example of the interfacing. We suggest that you follow this example through before trying to create your own. As in much of programming, it is sometimes easier to adapt an existing program than to start from scratch. If what you want is not there, let us know what it is and we can probably easily add it.



- i) Send suggestions on additions/deletions/changes to this Application Note to: support@dataray.com

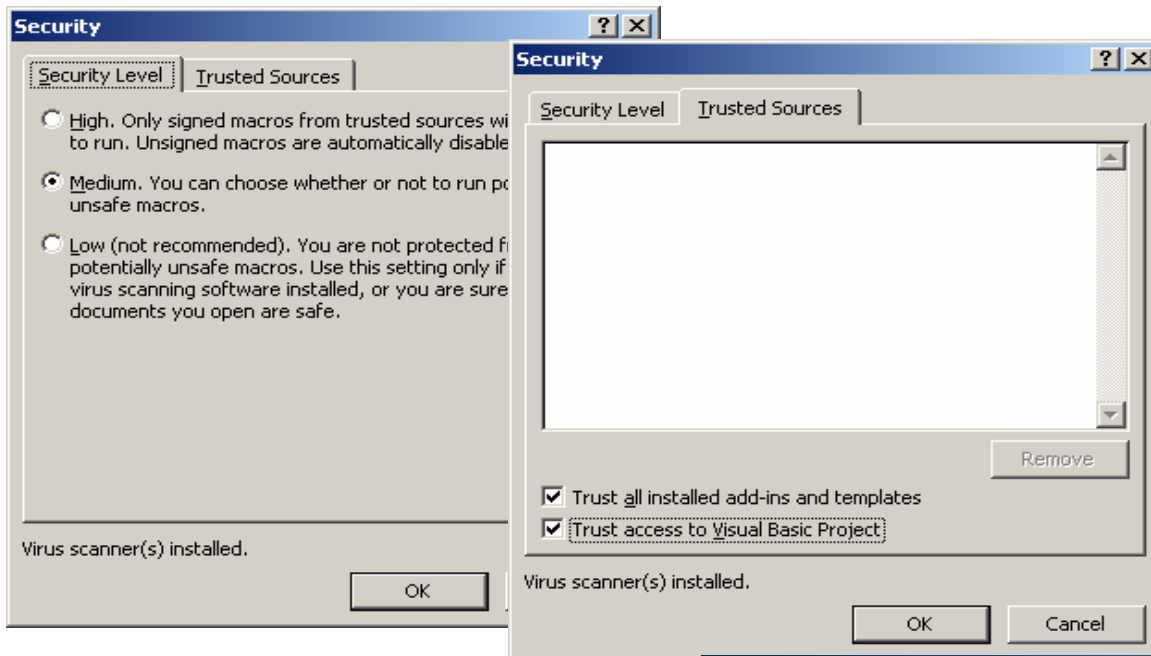
3. Setting up Excel for VBA

If your Microsoft Office / Excel was installed using **Typical** settings, then a number of functions are probably not enabled for VBA. You will need to check the following, and you will need to have your Microsoft Office CD to hand.

a) Change/check your security settings.

In the **Tools, Macro, Security...** pull-down menu, **Security Level** tab set it to **Medium**. You may set it to **Low** at your own risk.

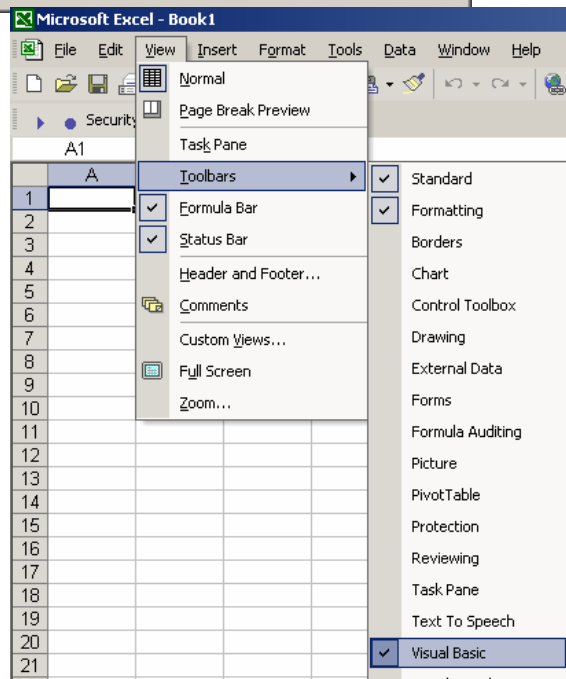
In the **Trusted Sources** tab settings to **Trust access to Visual Basic Project**.

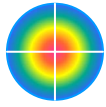


b) Check/install the VBA toolbar.



If the VBA toolbar shown above is not already visible, then in the pull-down menu go **View, Toolbars**, and select the **Visual Basic** toolbar.



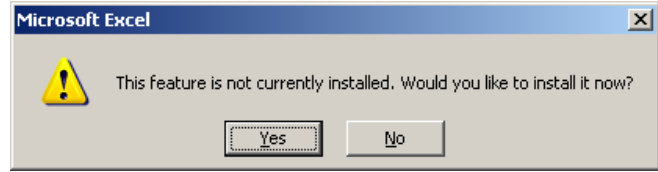


c) Install/verify the Microsoft Script Editor.



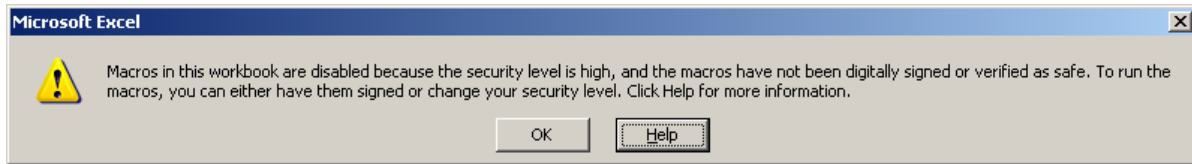
Microsoft Script Editor

If when you click on the Script Editor tool, you get the message: **This feature is not currently installed. Would you like to install it now?**, then insert your Microsoft Office CD and click **Yes**. [You may need Office Professional rather than Office Small Business Edition in order to enable this function.]

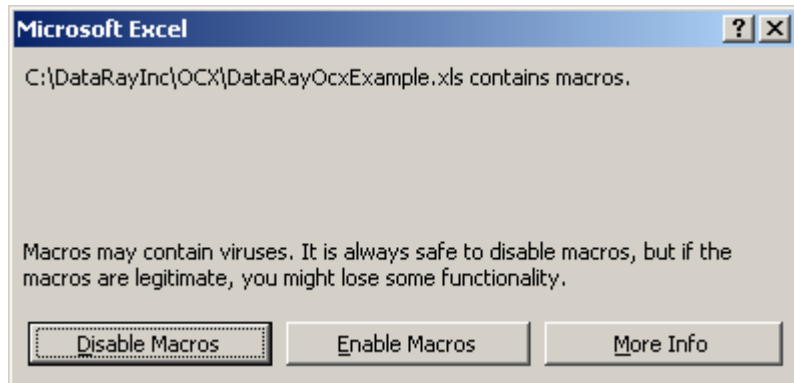


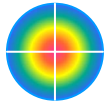
4. Running DataRayOcxExample.xls

- 1) Open Excel. Open **DataRayOCxExample.xls** from the c:\Program Files\DataRay, or in the directory in which you placed it. If you see the warning below, it means that you skipped Section 3 above.



If you see this screen warning, choose **Enable Macros**.





Once the **DataRayOCxExample.xls** workbook is open in Excel you will see the following screen. (Actual details may evolve as the software evolves.)

Multi-camera Alignment System

DataRayOCxExample.xls

WinCamD readout values are shown in green cells.
User entered values shown in blue cells are read by WinCamD.

Warning: Stop data collection before changing an input value.

General Settings

Centroid Clip level = 13.53 %
Crosshair orientation: A Value = A (for Auto) or 0 (for 0°)
Set Gamma: 1 Value = 0.2 to 5
Set Linear Filter %: 1 Value = 0, 0.1, 0.2, 0.5, 1, 2, 5 or 10 %
Set Area Filter NxN: 1 N = 1, 2, 3 or 4

Camera # and Settings

	1	2	3	4	
Y _C Centroid =	65	65	389	1320	μm
Y _G Centroid =	0	0	171	1325	μm
Mix Factor M =	1	1	1	1	M Value = 0 to 1
Mix: (M·Y _C) + (1-M)Y _G =	65	65	389	1320	μm
Δ (Camera # - Camera 4)	-1255	-1255	-931	0	μm
Auto Exposure time =	2.672	2.672	16.800	0.416	ms
Manual Exposure time =					ms
Exposure Setting:	A	A	A	A	A = Auto; M = Manual
Image Select:	1	2	3	4	Scroll All

Delta Centroid position: Camera N - Camera 4

Y μm	1	2	3	4
0				
-200				
-400				
-600				
-800				
-1000				
-1200				
-1400				

Yc Centroid position - Coarse

Yc μm	1	2	3	4
2500				
2000				
1500				
1000				
500				
0				
-500				
-1000				
-1500				
-2000				
-2500				

Start

Not open

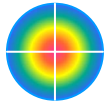
Camera1 Camera2 Camera3 Camera4

Yc[abs] 0.0 um

This example takes the data from up to four WinCamD cameras connected to up to two PCI cards.



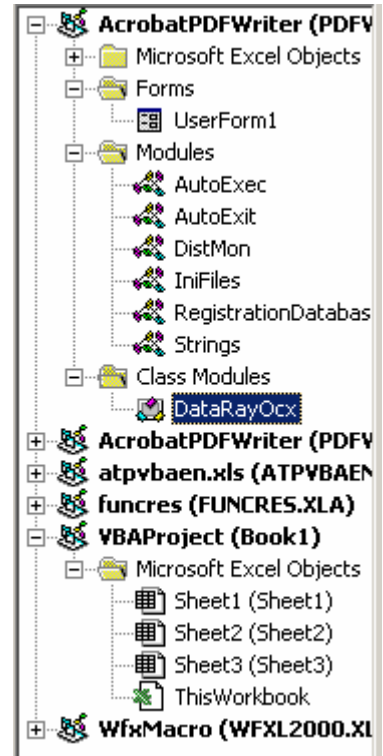
Press the **Visual Basic Editor** button to open it in Excel VBA.

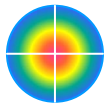


Click on the Maximize button (top right) to maximize the window.

It may well be that another file occupies the screen, in the author's case, **PDFWriter(1).xla** which is an Adobe Acrobat macro, used to create PDF documents out of Excel.

Find **DataRay.Ocx** in the **Class modules** directory, which may be a subdirectory of other VBAs that are already on your PC. (In the writer's case, **AcrobatPDFWriter**).





C) Steps to create Excel Form with DataRayOcx

- 1) From the menu select **Insert** and then select **UserForm**.

In the **Toolbox** dialog, right click an empty space, then select **Additional Controls...**

- 2) Under **Additional Controls** check:

Button Control

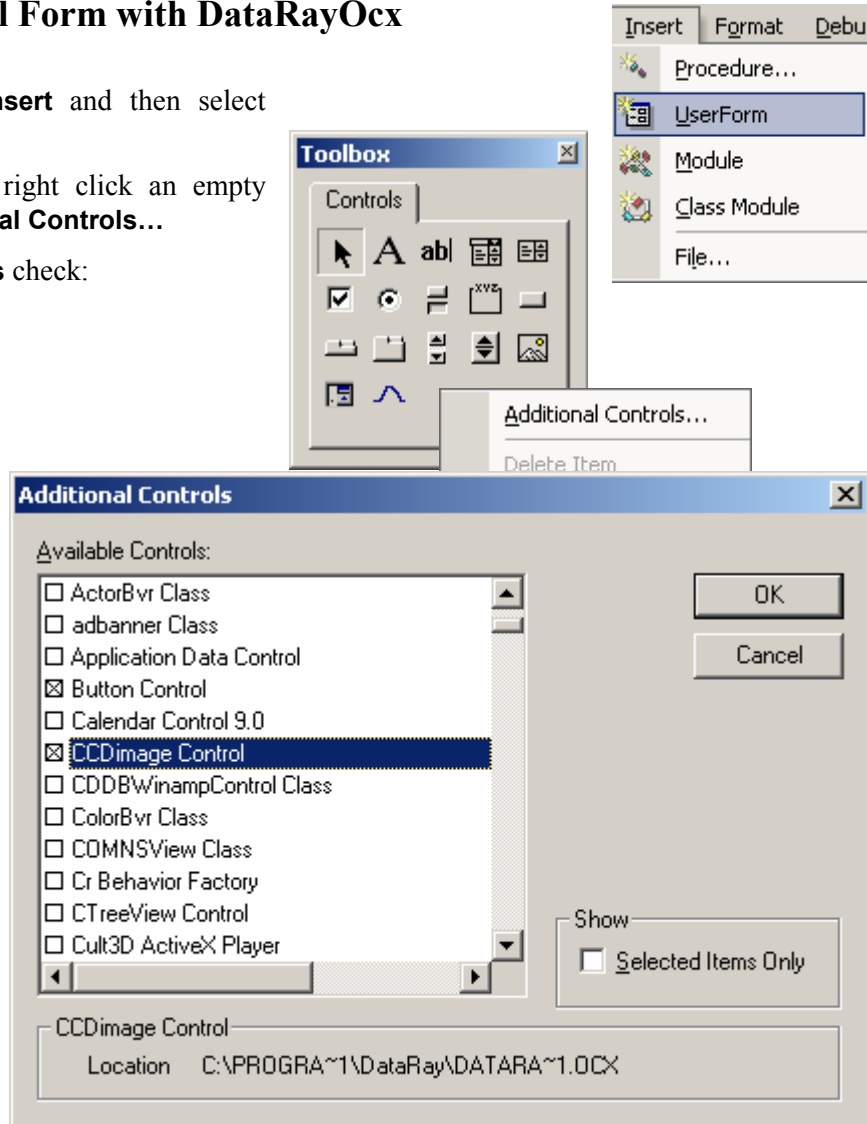
CCDimage Control

GetData Control

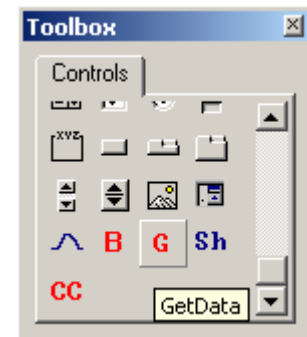
ShutterControl Control

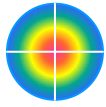
Profiles Control.

[**Note:** In Visual Basic it is much easier, you just need to select the **DataRayOcx Control.**]



- 3) The new controls should show up in the **Toolbox**, select **G** (for **GetData**) and drag and drop it onto the form. The **GetData** control is not visible when the form is run, so it can go anywhere.





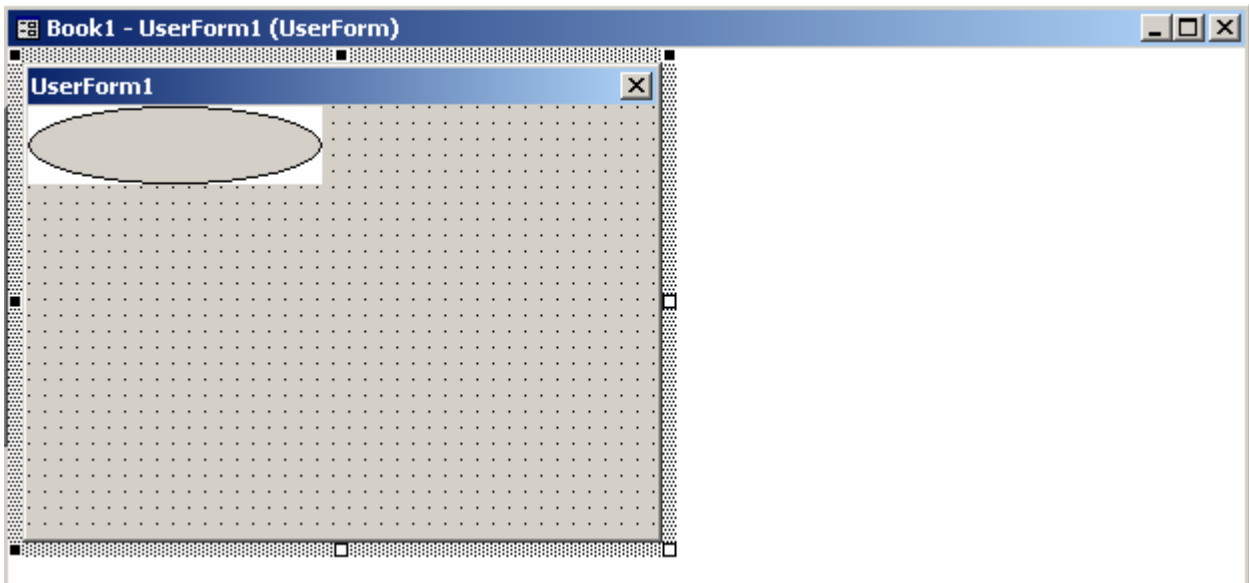
- 4) Double click the **GetData Control**, a new sub routine is created, unfortunately it's not complete yet, so for now just note the **GetData1** name. It signifies that it is the first instance of the **GetData Control** and there should only be one. This is a callback function that tells the camera that something

```

PDFWriter(1).xla - UserForm1 (Code)
GetData1  SendMessage
Private Sub GetData1_SendMessage(ByVal Message As Long, ByVal LongValue
End Sub
  
```

has changed. This is just an example to show where it comes from, but you will not need it further. Highlight the function that you just created and press **Delete**.

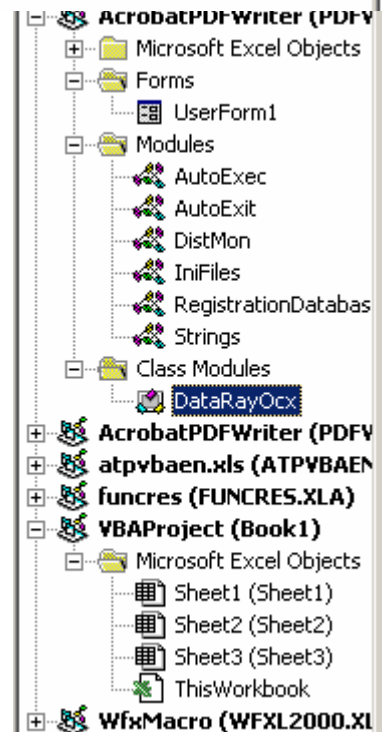
- 5) Select the **B (Button)** tool, and drag and drop it onto the form.

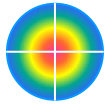


Note that the properties window is shown. Here's where we will define it. But first we need a class file that contains DataRay symbols. From the menu, select **File...**, **Import File** and import **DataRayOcx.cls**.

It will appear in the **Class modules** directory, which may be a subdirectory of other VBAs that are already on your PC. (In the writer's case, **AcrobatPDFWriter**, used to create PDF documents out of Excel.)

Double-click on **DataRayOcx.cls** to open it.





About one-third on the way down the **DataRayOcx(Code)** you will find a list starting:

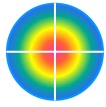
Enum DataRay_Parameters_Buttons

BlankParameters = 0

The screenshot shows a VBA code editor window titled "PDFWriter(1).xla - DataRayOcx (Code)". The code lists several parameters with their values, including **Yc_WinCamD = 164**. A Properties window for "Button1" is open on the right, showing various properties such as Name, ButtonID (164), Height, Width, and Visible.

Properties - Button1	
Button1 Button	
Alphabetic Categorized	
(About)	
(Custom)	
(Name)	Button1
ButtonID	164
ControlTipText	
Height	29.5
HelpContextID	0
Left	0
TabIndex	0
TabStop	True
Tag	
Top	30
Visible	True
Width	110.55

Scroll down the list to find the Y centroid position, **Yc_WinCamD**, and note that it is defined as = **164**.



Go back to the form and place **164** into the **ButtonID** location. The new button is now defined as **Yc_WinCamD**.



Now place another DataRay button into the form, search for 'StatusButton = 233', and define this ButtonID to 233. This is the "DataRay Status Button" that can be used to start and stop the cameras. GetData.StartDevice() / GetData1.StopDevice() will do the same thing without the status text.

6) Inserting a WinCamD image.

Menu item 'View->ToolBars->Visual Basic' must be checked.

Open the Microsoft Script Editor (colored infinity symbol). Note if you did a 'Typical' Excel install, then when you click on the Microsoft Script Editor tool, you will/may get the message: 'This feature is not currently installed. Would you like to install it now?' Insert your Microsoft Office disk and click OK. It may be that you will need Office Professional rather than Office Small Business Edition in order to enable this function.

Press [CTRL + ALT + X] "View toolbox"

Select menu Item 'Tools->Customized toolbox'

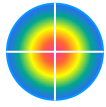
Verify CCDImage Control checked, then OK.

Select CCDImageCtrl, Right-click then Copy.

Go back to Excel, right click a free cell, noting the cell address, and click Paste. A black rectangle will appear.

In the Excel spreadsheet, click on a cell in the top left of the area where you wish to place the image, and then go Insert, Object, GetData control, and click OK. An oval will appear in the cell.

In the Toolbox, Click the Design mode (Ruler + Triangle + Pen) tool. The black rectangle will disappear and the oval will be outlined with a rectangle and eight handles, (small circles). Drag the cell locating the black rectangle (noted in 8) above, onto the outlined oval. Drag the lower right corner handle to size the image to taste. Click the Design mode button again. The large black rectangle will appear.



- 7) Select the 'Sh' (Shutter) tool and insert a Exposure control into the form. Go ahead and double click it, "Private Sub ShuterControl1_Enter() " function is created.

In side the function type in 'ShuterControl1' then add a '.' a selection dialog should appear, find 'SetID', now add a bracket '(' , you should see...

"ShuterControl1. SetID(ScrollID as Integer) As Boolean "

The finished expression should be "ShuterControl1. SetID(Exposure_Slider)

Where *Exposure_Slider* is from the DataRay OCX class we imported.

- 8) Next I added a 'Start' button (Not a DataRay Control Button) this because I couldn't find the default "On Load" function. Here I start the DataRayOcx driver with

'GetData1.OpenWinCamD(0)' Again after 'GetData1.' Is typed a dialog will appear that displays all the functions; this is the self documenting feature of Active X.

Note: The '0' parameter is not used and is there for backward compatibility only.

GetData1.OpenWinCamD(0) starts the DataRayOcx machine.

Next we need to tell the program what device we want to use is, this is done as follows.

GetData1.SetCurrentDevice(IS_WINCAM)

Then we can set which camera to use.

GetData1.CurrentCamera = Camera1

Note here that **CurrentCamera** is not a function but a property so it can be assigned or read.

TheCamera = GetData1.CurrentCamera

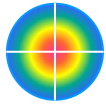
- 9) But you want to read data!

Dim Yc As Double

Yc = GetData1.GetParameter(Yc_WinCamD)

- 10) But you want to open the setup dialog!

a. GetData1.OpenDialog (CAPTURE_DLG)



- b.
- F5 Run
- F7 View code
- View object

D) DataRayOcx.cls Description

To simplify your interpretation of this list, it is necessary that you are first familiar with the normal operation of the DataRay software for the product that you are using. Then items which seem initially obscure will become more self-explanatory.

Text which is not color-coded applies to all products. Color-coded items are applicable as follows:

BeamMap, BeamMap-C, Beam'R, BeamScope-P7 family

BeamMap

BeamMap-C, ColliMate

Beam'R

Beamscope-P7

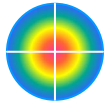
WinCamD, TaperCamD series

```
VERSION 1.0 CLASS
BEGIN
    MultiUse = -1 'True
    Persistable = 0 'NotPersistable
    DataBindingBehavior = 0 'vbNone
    DataSourceBehavior = 0 'vbNone
    MTSTransactionMode = 0 'NotAnMTSObject
END
Attribute VB_Name = "DataRayOcx"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False

Rem Use examples
Rem GetData1.FilterValue = 0.5
Rem GetData1.FilterValue = 0
Rem Filter = GetData1.FilterValue

Rem GetData1.InkSaverState = 0 // Off. The Inksaver palette function is off.
Rem GetData1.InkSaverState = 1 // On The Inksaver palette function is On.

Rem GetData1.JitterSuppression = 0 // Off
Rem GetData1.JitterSuppression = 1 // On. Normally set to On.
```



```
Rem GetData1.Wavelength = .667 // Sets Current device

Rem GetData1.WinCamFilter = 1 OR 2 OR 3 OR 4 // Sets the NxN Area averaging
function.

Rem CurrentCamera = GetData1.CurrentCamera // Initiates getting data from the
current camera

Rem GetData1.BaselineLocked = 0 // Off. Locks baseline - disables auto-zero.
Rem GetData1.BaselineLocked = 1 // On. Unlocks baseline - enables auto-zero.

Rem GetData1.WinCamNormalized = 0 // Off. Sets 2D image normalization off.
Rem GetData1.WinCamNormalized = 1 // On. Normalizes the 2D image peak to 100%.

Rem GetData1.BackGroundSubtraction = 0 // Off. No background subtraction.
Rem GetData1.BackGroundSubtraction = 1 // On. Initiates the Background
Subtraction routine.n

Rem GetData1.AutoNaming = 0 // Off
Rem GetData1.AutoNaming = 1 // On. Autonaming of saved files with date and
time is on.

Rem GetData1.OpenBeamScopeP7(0) // open all BeamScopes
Rem GetData1.OpenBeamMap(0) // open all Beam'R -- BeamMaps

Rem IsThere = GetData1.IsBeamScopeP7HeadThere

Rem GetData1.DoSearch() // starts search Beam'R, BeamMap and
BeamScopes

Rem GetData1.HomeP7Head() // Homes BeamScope finger

Rem GetData1.SaveFile() // Same as in exe program
Rem GetData1.PreviousProfile() // points to previous buffer location
Rem GetData1.NextProfile() // points to next buffer location
Rem GetData1.SelectProfile() // Opens beam select dialog

Rem GetData1.OpenClipLevelDlg(0) // opens Cliplevel a dialog
Rem GetData1.OpenClipLevelDlg(1) // opens Cliplevel b dialog
Rem Clip = GetData1.GetClipLevel(0)

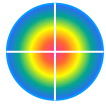
Rem GetData1.LoadDefaults() // Reloads all defaults
Rem Index = GetData1.GetBeamScopeIndex() // The Buffer index
Rem Count = GetData1.GetSampleCount() // Gets # of samples in the data buffer

Rem GetData1.SetAverageNumber(0_to_200) // Sets the # of samples to average.

Rem GetData1.OpenFile() // Dialog will prompt for filename.
Rem GetData1.SaveJobFile() // Dialog will prompt for filename.
Rem GetData1.LoadJobFile() // Dialog will prompt for Job filename.
Rem Live = GetData1.GetCurrentState() // Live data = 0, Recalled file = 1
Rem GetData1.SetDisplayMode(0to6)

Rem GetData1.StartDevice() // Starts current device
Rem GetData1.StopDevice() // Stops current device

Rem GetData1.Update() // Forces all data to be recalculated
```



```
Rem GetData1.SetToZero()           // Current Xc and Yc become the new zero
Rem GetData1.SetToAbsolute()       // Current Xc and Yc become absolute values
Rem BoolCameraThere = GetData1.IsCameraThere(0_to_7) // Checks for the
presence of a camera.

Rem GetData1.Palette = PAL_256 // Selects the image palette from list below.

Enum DataRay_Palettes
    PAL_256 = 1 // 256 colors
    PAL_GRAY = 2 // 256 gray scale
    PAL_16 = 3 // 16 colors
    PAL_10 = 4 // 10 colors (10% solid contours)
    MAX_PAL = 5 // Sets color range to maximum in Display Settings.
End Enum

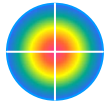
Enum DataRay_Cameras // Defines cameras 1 to 8 as 0 to 7.
    Camera1 = 0
    Camera2 = 1
    Camera3 = 2
    Camera4 = 3
    Camera5 = 4
    Camera6 = 5
    Camera7 = 6
    Camera8 = 7
End Enum

Rem GetData1.SetCurrentDevice(IS_WINCAM) // Sets the current device.

Enum DataRay_Devices // Specifies the number associated with the head types.
    IS_BEAMSCOPE = 1
    IS_BEAMR = 2
    IS_BEAMMAP = 3
    IS_BEAMC = 4
    IS_WINCAM = 5
    IS_TWOD_SCAN = 6
    IS_WINCAM_LOG = 7
    IS_SPARE3 = 8
    IS_SPARE4 = 9
    LAST_DEVICE = 10
End Enum

Rem Use examples
Rem GetData1.ToggleDialog(WANDER_DLG) // Toggles Wander dialog open/close.
Rem GetData1.ToggleDialog(BEAM_FIT_DLG) //
Rem GetData1.OpenDialog(EPPROM_DATA) // Opens the EEPROM data screen
Rem GetData1.CloseDialog(EPPROM_DATA) // Closes the EEPROM data screen

Enum DataRay_Dialogs // Specifies the number associated with certain dialogs.
    M2_DEBUG_DLG = 1 // M-squared debug dialog.
    M2_BEAMSCOPE_DLG = 2 // M-squared dialog
    M2_BEAMMAP_DLG = 3
    M2_BEAMC_DLG = 4
    DIV_BEAMMAP_DLG = 5
    DIV_BEAMC_DLG = 6
    WC_FLUENCE_DLG = 7 // WinCamD fluence dialog.
    NUMERIC_DISPLAY_MODES = 8 // The numeric display modes dialog
```



```
EPPROM_DATA = 9
LOGGING_DLG = 10
BS_PULSED_DLG = 11
WAVE_LENGTH_DLG = 12
CAPTURE_DLG = 13
PCI_EEPROM_DLG = 14
WANDER_DLG = 15
CENTROID_CLIP_DLG = 16
WC_IMAGE_LOG_SETUP_DLG = 17 // The image log set up dialog.
WC_IMAGE_LOG_DLG = 18 // The Image log dialog
WC_LOGGING_DLG = 19 // The logging dialog
BEAM_FIT_DLG = 20 // ??
```

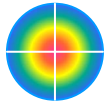
End Enum

Rem Use examples

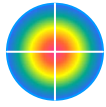
```
Rem Xc = GetData1.GetParameter(Xc_WinCamD) // Gets the X centroid value
Rem Yc = GetData1.GetParameter(Yc_WinCamD) // Gets the Y centroid value
Rem Text = GetData1.GetOcxResultName(Xc_WinCamD) // Text = "Xc_WinCamD"
Rem Text = GetData1.GetOcxResultName(Yc_WinCamD) // Text = "Yc_WinCamD"
```

Enum DataRay_Parameters_Buttons

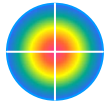
```
BlankParameters = 0
DefaultParameters = 1
u_BeamR_Width_at_Clip_1 = 2
u_BeamR_Width_at_Clip_2 = 3
u_BeamR_GFit = 4
u_BeamR_TopHat = 5
v_BeamR_Width_at_Clip_1 = 6
v_BeamR_Width_at_Clip_2 = 7
v_BeamR_GFit = 8
v_BeamR_TopHat = 9
v1_BeamC_Width_at_Clip_1 = 10
v1_BeamC_Width_at_Clip_2 = 11
v1_BeamC_GFit = 12
v1_BeamC_TopHat = 13
u1_BeamC_Width_at_Clip_1 = 14
u1_BeamC_Width_at_Clip_2 = 15
u1_BeamC_GFit = 16
u1_BeamC_TopHat = 17
v2_BeamC_Width_at_Clip_1 = 18
v2_BeamC_Width_at_Clip_2 = 19
v2_BeamC_GFit = 20
v2_BeamC_TopHat = 21
u2_BeamC_Width_at_Clip_1 = 22
u2_BeamC_Width_at_Clip_2 = 23
u2_BeamC_GFit = 24
u2_BeamC_TopHat = 25
v3_BeamC_Width_at_Clip_1 = 26
v3_BeamC_Width_at_Clip_2 = 27
v3_BeamC_GFit = 28
v3_BeamC_TopHat = 29
u3_BeamC_Width_at_Clip_1 = 30
u3_BeamC_Width_at_Clip_2 = 31
u3_BeamC_GFit = 32
u3_BeamC_TopHat = 33
v4_BeamC_Width_at_Clip_1 = 34
v4_BeamC_Width_at_Clip_2 = 35
```



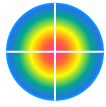
```
v4_BeamC_GFit = 36
v4_BeamC_TopHat = 37
u4_BeamC_Width_at_Clip_1 = 38
u4_BeamC_Width_at_Clip_2 = 39
u4_BeamC_GFit = 40
u4_BeamC_TopHat = 41
minus2_BeamMap_Width_at_Clip_1 = 42
minus2_BeamMap_Width_at_Clip_2 = 43
minus2_BeamMap_GFit = 44
minus2_BeamMap_TopHat = 45
minus1_BeamMap_Width_at_Clip_1 = 46
minus1_BeamMap_Width_at_Clip_2 = 47
minus1_BeamMap_GFit = 48
minus1_BeamMap_TopHat = 49
zero_BeamMap_Width_at_Clip_1 = 50
zero_BeamMap_Width_at_Clip_2 = 51
zero_BeamMap_GFit = 52
zero_BeamMap_TopHat = 53
plus1_BeamMap_Width_at_Clip_1 = 54
plus1_BeamMap_Width_at_Clip_2 = 55
plus1_BeamMap_GFit = 56
plus1_BeamMap_TopHat = 57
plus2_BeamMap_Width_at_Clip_1 = 58
plus2_BeamMap_Width_at_Clip_2 = 59
plus2_BeamMap_GFit = 60
plus2_BeamMap_TopHat = 61
plus45_BeamMap_Width_at_Clip_1 = 62
plus45_BeamMap_Width_at_Clip_2 = 63
plus45_BeamMap_GFit = 64
plus45_BeamMap_TopHat = 65
minus45_BeamMap_Width_at_Clip_1 = 66
minus45_BeamMap_Width_at_Clip_2 = 67
minus45_BeamMap_GFit = 68
minus45_BeamMap_TopHat = 69
u_BeamScope_Width_at_Clip_1 = 70
u_BeamScope_Width_at_Clip_2 = 71
u_BeamScope_GFit = 72
u_BeamScope_TopHat = 73
v_BeamScope_Width_at_Clip_1 = 74
v_BeamScope_Width_at_Clip_2 = 75
v_BeamScope_GFit = 76
v_BeamScope_TopHat = 77
u_WinCamD_Width_at_Clip_1 = 78 // The Clip[a] 2Wua beam diameter
u_WinCamD_Width_at_Clip_2 = 79 // The Clip[b] 2Wub beam diameter
u_WinCamD_GFit = 80 // The Clip[a] 2Wua profile Gaussian fit %
u_WinCamD_TopHat = 81 // The Clip[a] 2Wua profile Top Hat fit %
v_WinCamD_Width_at_Clip_1 = 82 // The Clip[a] 2Wva beam diameter
v_WinCamD_Width_at_Clip_2 = 83 // The Clip[b] 2Wvb beam diameter
v_WinCamD_GFit = 84 // The Clip[a] 2Wva profile Gaussian fit %
v_WinCamD_TopHat = 85 // The Clip[a] 2Wva profile Top Hat fit %
Xc_BeamR = 86
Yc_BeamR = 87
Xg_BeamR_not_used = 88
Yg_BeamR_not_used = 89
Xp_BeamR_not_used = 90
Yp_BeamR_not_used = 91
Ellipticity_BeamR = 92
Power_BeamR = 93
Xc1_BeamC = 94
```



```
Yc1_BeamC = 95
Xg1_BeamC_not_used = 96
Yg1_BeamC_not_used = 97
Xp1_BeamC_not_used = 98
Yp1_BeamC_not_used = 99
Xc2_BeamC = 100
Yc2_BeamC = 101
uMsquared_BeamC_Alt = 102
uM2_Zo_BeamC_Alt = 103
Xp2_BeamC_not_used = 104
Yp2_BeamC_not_used = 105
Xc3_BeamC = 106
Yc3_BeamC = 107
vMsquared_BeamC_Alt = 108
vM2_Zo_BeamC_Alt = 109
Xp3_BeamC_not_used = 110
Yp3_BeamC_not_used = 111
Xc4_BeamC = 112
Yc4_BeamC = 113
Xg4_BeamC_not_used = 114
Yg4_BeamC_not_used = 115
Xp4_BeamC_not_used = 116
Yp4_BeamC_not_used = 117
Ellipticity_BeamC = 118
Power_BeamC = 119
Xc_BeamMap = 120
Yc_BeamMap = 121
Xg_BeamMap_not_used = 122
Yg_BeamMap_not_used = 123
Xp_BeamMap_not_used = 124
Yp_BeamMap_not_used = 125
Ellipticity_BeamMap = 126
Power_BeamMap = 127
DivergenceNA_BeamMap = 128
DivergenceDegrees_BeamMap = 129
DivergenceRadians_BeamMap = 130
Msquared_BeamMap = 131
M2_2Wo_BeamMap = 132
M2_Zo_BeamMap = 133
M2_Zr_BeamMap = 134
M2_Theta_BeamMap = 135
M2_NA_BeamMap = 136
Xc_BeamScope = 137
Yc_BeamScope = 138
Xg_BeamScope = 139
Yg_BeamScope = 140
Xp_BeamScope = 141
Yp_BeamScope = 142
Ellipticity_BeamScope = 143
Power_BeamScope = 144
uDivergenceNA_BeamC = 145
uDivergenceDegrees_BeamC = 146
uDivergenceRadians_BeamC = 147
uMsquared_BeamC = 148
uM2_2Wo_BeamC = 149
uM2_Zo_BeamC = 150
uM2_Zr_BeamC = 151
uM2_Theta_BeamC = 152
uM2_NA_BeamC = 153
```



```
vDivergenceNA_BeamC = 154
vDivergenceDegrees_BeamC = 155
vDivergenceRadians_BeamC = 156
vMsquared_BeamC = 157
vM2_2Wo_BeamC = 158
vM2_Zo_BeamC = 159
vM2_Zr_BeamC = 160
vM2_Theta_BeamC = 161
vM2_NA_BeamC = 162
Xc_WinCamD = 163 // The Xc mathematical centroid
Yc_WinCamD = 164 // The Yc mathematical centroid
Xg_WinCamD = 165 // The Xg geometric centroid
Yg_WinCamD = 166 // The Yg geometric centroid
Xp_WinCamD = 167 // The Xp peak position
Yp_WinCamD = 168 // The Yp peak position
Ellipticity_WinCamD = 169 // The ellipticity
Power_WinCamD = 170 // The relative power value in the selected units
Oreintation_WinCamD = 171 // The orientation of the ellipse in degrees
MajorWidth_WinCamD = 172 // The ellipse major diameter 2W_Major
MinorWidth_WinCamD = 173 // The ellipse minor diameter 2W_Minor
MeanWidth_WinCamD = 174 // The ellipse mean diameter 2W_Mean
Peak_WinCamD = 175 // The Peak value as a % of the maximum possible ADC
level
AverageFluence_WinCamD = 176 // The Average fluence value from the dialog
uM2_M2_BeamScope = 177
uM2_2Wo_BeamScope = 178
uM2_Zo_BeamScope = 179
uM2_Zr_BeamScope = 180
uM2_Theta_BeamScope = 181
uM2_NA_BeamScope = 182
vM2_M2_BeamScope = 183
vM2_2Wo_BeamScope = 184
vM2_Zo_BeamScope = 185
vM2_Zr_BeamScope = 186
vM2_Theta_BeamScope = 187
vM2_NA_BeamScope = 188
ID_WANDER = 189
PointingX_BeamMapC = 190
PointingY_BeamMapC = 191
PointingX_BeamMap = 192
Msquared_BeamMap_Alt = 193
M2_Zo_BeamMap_Alt = 194
M2_Theta_BeamMap_Alt = 195
M2_Zo_BeamMap_Alt2 = 196
Spare9 = 197
Spare10 = 198
Spare11 = 199
Spare12 = 200
Spare13 = 201
Spare14 = 202
Spare15 = 203
Spare16 = 204
Spare17 = 205
Spare18 = 206
Spare19 = 207
Spare20 = 208
Spare21 = 209
Spare22 = 210
Spare23 = 211
```



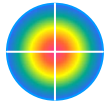
```
Spare24 = 212
Spare25 = 213
Spare26 = 214
Spare27 = 215
Spare28 = 216
Spare29 = 217
Spare30 = 218
Spare41 = 219
Spare42 = 220
Spare43 = 221
Spare44 = 222
Spare45 = 223
Spare46 = 224
Spare47 = 225
Spare48 = 226
Spare49 = 227
Spare50 = 228
Last_PARAMETER_ID = 229

SetClipLevel1 = 230 // Sets the Clip[a] level
SetClipLevel2 = 231 // Sets the Clip[b] level
PeakButton = 232 // Displays the Peak value as a % of the maximum possible
ADC level
StatusButton = 233 // Displays the Device status
ZeroButton = 234 // Toggles between zero and absolute centroid display
SpacerButton1 = 235 // A filler bar
SpacerButton2 = 236 // A filler bar
ZoomButton = 237 //
CrossHairButton = 238 //
AuxStageScale_2D = 239
TwoDsetup_2D = 240
SetReference_2D = 241
Reference_2D = 242
HomeStage_2D = 243
Scan_2D = 244
DoSearch = 245

Last_ID = 246
End Enum

Rem GetData1_SendMessage(ByVal Message As Long, ByVal LongValue As Long, ByVal
DoubleValue As Double)
Rem You can safely ignore all messages, LongValue and DoubleValue not used

Enum DataRay_Messages
M2_DONE = 1
M2_START = 2
GLOBAL_CLIP_CHANGE = 3 // Global changes in clip level is selected
GLOBAL_STOP_CAPTURE = 4 // ??
AUTO_3D_UPDATE = 5 // Auto-update of the 3D display is selected
AUTO_2D_UPDATE = 6 // Auto-update of the 2D display is selected
JITTER_CONTROL = 7 // Jitter suppression is selected for BeamMap/ Beam'R
style heads.
PALETTE_CHANGE = 8 // ??
INK_SAVER_CHANGE = 9 // ??
AUTO_NAMES = 10 // ??
LIVE_RECALL_STATE = 11 // ??
XC_YC_CHANGE = 12 // ??
STATE_CHANGE = 13 // ??
```



```
DATA_CHANGE = 14 // ??  
BS_DATA_CHANGE = 15 // ??  
HELP_MSG = 16 // ??  
End Enum
```

```
Rem ShuterControll.SetID (Exposure_Slider) // ??  
Rem ShuterControll is the First slider inserted // ??
```

```
Enum DataRay_Sliders // Defines the Slider numbers  
    Trigger_Slider = 0 // The trigger slider for pulsed lasers  
    Gain_Slider = 1 // The Gain slider for CCD Gain  
    Exposure_Slider = 2 // The CCD Exposure slider  
End Enum
```

Appendix A. Excel and VBA bibliography culled from the Amazon.com website.

Excel 2002 Power Programming with VBA by John Walkenbach (Author) (**Paperback** - July 2001) Avg. Customer Rating: ★★★★★

Excel 2002 VBA Programmers Reference by Rob Bovey, et al (**Paperback**) Avg. Customer Rating: ★★★★★

Microsoft Excel 2002 Visual Basic for Applications Step by Step by Reed Jacobson (**Paperback**) Avg. Customer Rating: ★★★★★

Definitive Guide to Excel VBA by Michael Kofler (**Paperback** - November 2000) Avg. Customer Rating: ★★★★★

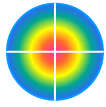
Excel 2000 VBA Programmers Reference by John Green, et al (**Perfect Paperback** - April 1999) Avg. Customer Rating: ★★★★★

Power Programming with VBA/Excel by Steven C. Chapra (**Paperback**)

Microsoft Excel VBA Programming for the Absolute Beginner by Duane Birnbaum (**Paperback**) Avg. Customer Rating: ★★★★★

Course ILT: Microsoft Excel 2002: VBA Programming by Course Technology (Editor), et al (**Paperback**)

Learn Microsoft Excel 2000 VBA Programming by Julitta Korol (**Paperback** - February 2000) Avg. Customer Rating:



DataRay Inc.
Application Note

Beam Profiling ... Engineered as a System
Delivered as a Solution!
www.dataray.com

Draft release: 19 Jan. 2002
This Rev: 0310A 10 Oct.
2003

Learn MS Excel 2002 VBA/XML Programming by Julitta Korol (**Paperback** - November 2002)