

Create a CustomDSO Plug-In with Visual Studio

Peter Wilhelm LeCroy Corporation

Customize Your Scope With Your Own Program Developed In Visual Basic From Microsoft's Visual Studio

The new WaveMaster platform allows to plug-in a custom ActiveX Control designed in an environment like Visual Studio in Visual Basic and to merge this GUI with the scope user interface. This feature opens up the door to a huge variety of applications previously not possible or only via remote control from an external PC.

Here we discuss now how a simple CustomDSO PlugIn can be created in Visual Basic and how it is installed on the WaveMaster. Note: What we refer to WaveMaster is also true for the WavePro7000 – Series and the SDA, the Serial Data Analyzer. Also, it is assumed that Visual Studio is installed and run on WaveMaster.

System Requirements:

- Visual Studio 6.0
- WaveMaster/WavePro7000/SDA
- XDEV Software Option



Figure 1 The CustomDSO PlugIn is completely integrated into the scopes interface



Start now Visual Basic and create a new ActiveX Control as shown in Figure 2.

Figure 2

🐂 Proj	ect1 - Microsoft Visual Basic [design]		(
<u>File</u> Edi	t <u>V</u> iew <u>P</u> roject F <u>o</u> rmat <u>D</u> ebug <u>R</u> un Query Diagram <u>T</u> ools <u>A</u> dd-Ins <u>1</u>	<u>N</u> indow <u>H</u> elp		
- 🛃	🏷 • 🏗 🚅 🛃 🙏 🗈 🏙 🏘 🗠 斗 🔳 🔳 😻 🖆	' 名 📽 🛠 🔁		>>
	1 BA ?			
	🗙 🛒 Project1 - UserControl1 (UserControl)	Properties - UserCo	ontrol1	×
Genera		UserControl1 User	Control	-
N 👪		Alphabetic Categor	ized	
A lah	T	(Name)	UserControl1	~
2 1 100		AccessKeys		
]	Alignable	False	
		Appearance	1 - 3D	=
. ⊡		AutoRedraw	False	
	. D	BackColor	8H800000F8	ι
		BackStyle	1 - Opaque	
ana 🔺		BorderStyle	0 - None	
-		CanGetFocus	True	
å∈]	ClipBehavior	1 - Use Region	
		ClipControls	True	
]	ControlContainer	False	
	· · · · · · · · · · · · · · · · · · ·	DataBindingBehavior	0 - vbNone	
R ~		DataSourceBehavior	0 - vbNone	
		DefaultCancel	False	
		DrawMode	13 - Copy Pen	×
2		(Name)		
		Returns the name use	d in code to identif	fy an object.

Figure 3

What you will see is similar to the project window to the left. The project is called by default Project1 and the UserControl UserControl1. The Form which is the user interface is an empty form. Click into Properties — UserControl1 and change the

- (Name) to PlugIn1
- Width to 11940
- Height to **2475**

Save File As				? 🛛
Save in: 🗀	CustomDSO PlugIn	•	* 🖻	· · · ·
File <u>n</u> ame:	Plugin1.ctl			<u>S</u> ave
Save as type:	User Defined Control Files (*.ctl)		-	Cancel
				<u>H</u> elp

Then save the control with File/Save Project As... PlugIn1.ctl.

Figure 4	Save the control	as PlugIn1

Save Projec	t As		? 🔀
Save in: 🗀	CustomDSO PlugIn	▼ ← € (* 📰 •
File <u>n</u> ame:	Project1.vbp		<u>S</u> ave
Save as type:	Project Files (*.vbp)	•	Cancel
			Help

Figure 5 Save the project as Project1

Visual Basic will prompt for the project name as well. Name the project Project1.vbp

ĥ	Project	1 - Microsoft Visual Basic [design]			
Ð	e Edit y	ew Project Figmat Debug Bun Query Djagram Tools Add-Ins Window Help			
П	s - 15	・ 〒 😂 🔲 月 時 読 蒔 ク ク 🖡 🚛 😹 輝 保 智 🛠 🔂 中 0.0 - 2月1940 x 2475			
		8 2			
	×	Refert - Ruslet (Recontrol)	Properties - PlugIn	1	×
(ieneral		PlugIn1 UserContro	1	*
			Alphabetic Catago	Ineri	_
			(Asses)	The tot	-
P	abi		(Name)	Plugini	
121	1		Aliceable	False	
L		a	Appearance	1 - 3D	_
F			AutoRedraw	False	
_			BackColor	BH800000F&	
E			BackStyle	1 - Opaque	
	비원		BorderStyle	0 - None	_
Γ.		-	CanGetFocus	True	- 11
Ę) 🗆		ClipBehavior	1 - Use Region	- 11
e	I B		CaptrolControls	True	- 11
-			DataBiodicoBehavior	0 - ubNone	
2	\sim				<u> </u>
-	ः सम		Width		
			Returns/sets the widt	h of an object.	

What you will see is the window Project1 with the empty resized Form Project1 — PlugIn1 in Figure 6.

Figure 6 The UserControl PlugIn1 is set to the correct size

ta P	roject	1 - Microsoft Visual Basic [design]			. 🗆 🛛
Ele	Edit V	ew Broject Figmat Debug Bun Query Djagram Tools Add-Ins Window Help			
	- 🍖	- TE 🚅 🖬 基 📭 🗈 前 🗁 🕞 🚛 😸 📾 🖓 🛠 😭 🕆 5400,960 - 井 1215 x 495			
	1 66 8				
	66 G		Properties - Con	Ibacao	
		😰 Project1 - Plugin1 (UserControl)	rioperties - con		
Gen	eral		Command1 Com	mandButton	-
k			Alphabetic Cate	gorized	
A	abi		(Name)	Command1	^
227		· · · · ·	Appearance	1 - 3D	- -
\square	_	Command1	Cancel	False	
N	6		Caption	Command1	
EB	EB		CausesValidation	True	
			Default	False	
최 최	1		Descripture	(None)	
rða -			DragIcon	(None)	
\sim	-		DragMode	0 - Manual	
	6		Enabled	True	
10	~		Font	MS Sans Serif	<u>×</u>
	ÐÐ		BackColor		
	1	¢	Returns/sets the b	ackground color used to	o display

Figure 7 Create a Command Button



Figure 8 Edit the Properties of the Command Button

Project - Microsoft Visual Basic (design)

E Edit View Dopat Figmat Debug Bun Query Dagram Tools & ddd brs Vindow (bdo

Soft A A Ref
Project - Plugin1 (UserControl)

Project - Plugin1 (Code)

Single

Project - Plugin1 (Code)

Single

Control

Single

Control

Single

Single

Control

Single

Code. The code window shows the code prepared for the button Single. If there is another name than Single displayed in the selection list drop-down menu on the left-hand side — click the arrow on the right end of the selection list and select Single before you release the mouse button.

Right-click on the button Single and select View

Figure 9 View the code that is linked with the Command Button Single

In the next step, we are going to place a couple of buttons on the Form and link them with functions that are executed when the related buttons are pressed.

Double-click on the CommandButton in the Control window on the left hand side. A button called Command1 is placed in the Form.

Rename the (Name) and the Caption to **Single** in the CommandButton Properties.

🖍 Project	1 - Microsoft Visual Basic [design]					🛛
Ele Edit V	ew Broject Format Debug Bun Que	ry Djagram Ijools Add-Ins Window	Help				
General	Project1 - PlugIn1 (UserCon	trol)			Properties - Sin Single Command	igle Button	×
► S	Project1 - Plugin1 (C Single Frivate Sub Sir Dim app As	pde) gle_Click()	Click		Alphabetic Cat (Name) Appearance BackColor Canol	single 1 - 3D False Single	
	End Sub E ODE	cANCELBOOL COLOR			CausesValidation Default DisabledPicture DownPicture DragIcon	Single True False (None) (None)	
		HANDLE OPTEXCLUSIVE		• • •	DragMode Enabled Font	0 - Manual True MS Sans Serif	
	<			>	Returns the name	used in code to ident	ify an object.

Figure 10 Define app as an Object

Insert the line: Dim app As Object

This defines app as an Object.

🖍 Project	Microsoft Visual Basic [design]	
Ele Edit y	Project Format Debug Run Query Diagram Tools Add-Ins Window Help	
3.6	◎ 📽 🖬 🚴 № 🏙 🏘 🕶 🖙 🔰 📓 🕼 🕾 🕾 🛠 🔁 🕼 13, 60 48	
	2	
General	Project1 - PlugIn1 (UserControl)	×
N 🔛	Project 1 - Plugin1 (Code)	_
A [ab]	Single Click	
	Dim app As Object Set app = CreateObject("LeCroy.XStreamDSO") End Sub End Sub	=-
ৰু মন নু	DaddedFicture (Yone) DownPicture (Yone)	=
	□ Dragtom (Vione) □ Dragtom (Vione) □ Dragtom (Vione) □ Tragtom (Vione) □ Tragtom (Vione) □ Tragtom (Vione)	
🔊 🔨	The second secon	<u>~</u>
S 22	Returns the name used in code to identify an	h object.

Insert the line: Set app = CreateObject ("LeCroy.XStreamDSO")

This makes the connection to the WaveMaster scope.

Figure 11 Create the object LeCroy.XStreamDSO that connects to WaveMaster

🖍 Project	1 - Micr	rosoft	: Visual Basic (design)				
Ele Edit V	ew Broje	ect Fi	grmat Debug gun Query Diagram Tools Add-Ins Window Help				
B. 3	• 🖪 🖡	~ L	┃ 赤 喩 臨 M ** ?* ▶ = 刻 証 * 凸 音 ズ G Ln 6, Col 1				
	2 ?						
×	R Pro	niect 1	1 - Pluela 1 (UserControl)	a	Properties - Singl	le	×
General				1	Single CommandBu	utton	•
N 🔛	100		Sector M. Blanded (Code)		Alphabetic Categ	porized	
A abi	11.		roject1 - Plugin1 (Lode)		(Name) S	Single	^
TUR TOWN		Sir	ngle Click 🔹		Appearance 1	1 - 3D	
			Private Sub Single Click()		BackColor	&H8000000F&	
F G	1111		Dim app As Object		Canton	-ase Sinale	
_			Set app = CreateObject("LeCroy.XStreamD50")		CausesValidation T	Frue	
	1111		app.Acquisition.TriggerMode = "Single"		Default F	False	
মম শ্	1111		Liid Sub	1	DisabledPicture (None)	
				1	DownPicture (None)	
0 -					DragIcon (None)	
					Enabled T	True	
-		26			Font N	MS Sans Serif	~
N >	I '				Leen in		
	<			,	(Name)	and in code to identify	an abiant
	194 - C			<u> </u>	Returns the name us	sed in code to identify	ar object.

Figure 12 Add the Automation command Single Trigger

Insert the line: app.Acquisition.TriggerMod e = "Single"

This command tells the instrument to execute a single acquisition as the user would press the Single button on the user interface.

Hint:

To learn about the syntax of the Automation commands, consult either the Automation Manual or run the program X-Stream Browser on the instrument with its interactive connection and online syntax.

10	Project	- Microsoft Visual Basic [design]			
Ble	Edit ye	w Broject Figmat Debug Bun Query Djagram Tools Add-Ins Window Help			
In					
12	· · u				
	표왕	9			
	×		Properties - Nor	rmal	×
Ger	real	Project1 - Plugin1 (UserControl)	Normal Comman	dB ittoo	1
k			Alphabetic Cat	egorized	_
Δ	िका	Single Normal -	(Name)	Normal	~
-	Tant	· · · · · · · · · · · · · · · · · · ·	Appearance	1 - 3D	
111			BackColor	aH800000F8.	
_	_		Cancel	False	- E .
P	æ		Caption	Normal	
E.B.			CausesValidation	True	
Ξü	8		Default	False	
ALM.	1		DisabledPicture	(None)	
	2		DownPicture	(None)	
Ö			DragIcon	(None)	
×	_		DragMode	0 - Manual	
			Enabled	True	
	~		Font	MS Sans Serif	~
~					
	빈민		Caption	and dealers die an ab	
-	9		Returns/sets the	text displayed in an obj	jects title

Place another button on the Form, rename the (Name) and Caption to **Normal**. Arrange the two buttons on the Form in line and then View Code of the Normal button.

Figure 13 Add another Command Button Normal and edit its properties

1 , Project	1 - Micr	isual Basic [design]	
Ele Edit)	jew Broje	nat Qebug Bun Query Djagram Tools Add-Ins Window Help	
1. 10	- 🗇 🕻	※ 部 職 構 🕺 😳 🕨 🗉 😸 超 名 🗑 🛠 🕃 (n 4, col 42	
	2 ?		
×	🖬 Pro	PlugIn1 (UserControl)	ormal 🗙
General	10000	Normal Com	andButton .
N 🔛		njecti - Piugini (Code) 📃 🗖 🔀 - Aphabetic 🖉	ategorized
A abi		sal 💌 Click 💌 🚺 (lame)	Normal
1791	11111	rivate Sub Normal Click()	1-3D
		Dim app As Object	False
9 9	1111	Set app = CreateObject("LeCroy.XStreamDSO") Capton	Normal
	1111	app.Acquisition.TriggerMode = "Normal"	on True -
		Default	False
되지 뭐		na Sub	2 (None)
A		Downwrthe	(None)
00		Dim app As Object DragMode	0 - Manual
🗀 🗈		Set app = CreateObject("LeCroy.XStreamDSO") Enabled	True
• •		app.Acquisition.TriggerMode = "Single"	MS Sans Serif
• · ·		na sub	
[] []	<	Caption	a text declared in an object's title
		Keturnsjøets t	e text displayed in an object's title

Copy the code we entered in Private Sub Single and paste it Private Sub Normal. Just change the trigger mode from "Single" to "Normal".

Figure 14 Both Command Buttons are now linked with their call back functions

Project1 - Project Properties
Debugging
When this project starts
○ Wait for components to be created
Start component: PlugIn1
C Start program:
C Start browser with URL:
OK Cancel Help

Figure 15 When the project starts start component PlugIn1

We are now ready for a first run. Go either to the menu Run and select Start or type F5 or click on the Start-array in the toolbar.

After running the project, this window comes up. Press OK to continue, ensuring the start component checkbox is selected.

C:\Program Files\Microsoft Visual Studio\VB98\PlugIn1.html - Microsoft Internet Explorer	
Ele Edit View Favorites Tools Help	
🕞 Back - 🌍 - 🖹 🙆 🏠 🔎 Search 📌 Favorites 🜒 Media 🤣 😥 - چ 😿 - 🗔 🖓	
🕴 Address 🖉 C: \Program Files \Viicrosoft Visual Studio \VB98 \PlugIn 1.html 🔍 🄁 🛛	io Einks »
	~
Single Normal	
	\sim
🛎 Done 😔 😌 My Compu	iter 🚲

Figure 16 Run mode is executed in Internet Explorer

🖌 Project1 - Microsoft Visual Basic [run]		
Bie Edit View Broject Figmat Debug Bun Query 장·친·ሽ 의 문 사람 환 분 세 이	Dagram Tools Add=tins Window Help ▷ ▶ II ■ 형 윤 동 영 옷 In 4, Col 42	
Project1 - PlugIn1 (Code)	▼ Click	Project - Project X
Private Sub Normal_Cii Dim app As Object Sat app - CreateQ app.Acquisition.Ti End Sub Private Sub Single_Cii Dim app As Object Sat app = CreateQ app.Acquisition.Ti End Sub	ck() ject("LeCroy.XStreamDSO") iggerNode = "Hormal [*] ck() ject("LeCroy.XStreamDSO") iggerNode = "Single"	 , A HOUTCHOILES

Figure 17 Stop the execution of the ActiveX Control in run-mode

If all goes well, no syntax error or typos are detected, Internet Explorer starts up and shows the Form we created and is ready to use.

Click now on Single and watch the instrument. Does it trigger once? And also hit Normal. In this case the trigger mode is changed from Single mode to Normal Mode.

To stop the run-mode of your ActiveX Control click either the Stop icon in the toolbar or select End from the Run-menu.

🖍 Project	1 - Microsoft Visual Basic [design]			
Ele Edit y	iew Project Figmat Qebug Bun Query Djagram Iools Add-Ins Window Help			
1. 10	・ 面 📽 🖬 糸 釉 ඬ 柿 ロ 🕫 🖡 🗉 😸 🕼 名 答 🛠 🕄 怕 0,0 🛛 🟥 11940 x 2475			
	ê ?			
×	Project Plugla 1 (Iron Control)	Properties - Plug	n1	×
General		PlugIn1 UserContr	ol	•
N 🔛		Alphabetic Categ	orized	
A abi	Single Normal	(Name)	PlugIn 1	^
‴] 🗖		Alignable	False	-
N N	•	Appearance AutoRedraw	1 - 3D False	
		BackColor	BHODEDEDED&	-
AD 1		BorderStyle p	alette System	- 1
	•	CanGetFocus		
00		ClipControls		
		ControlContaine		
ø <		LatasindingBeh		
	x	BackColor		
		Returns/sets the		

Figure 18 Match the colors of PlugIn and scope user interface

Before we are going to "Make" the project, we'll set the color of the Form so it matches the scope's interface color.

Select the Form, search for BackColor in the properties and left-click on the arrow of that control to open the Palette.

Right-click in one of the white undefined color fields and the



olor/Solid Lum: 1240 Blue: 1255 Add Color Gose Help Figure 19 Define the color in the

Define Color window



Figure 20 Set RGB to match WaveMaster color Set RGB to the following values:

In the Define Color window you define RGB for WaveMaster user interface.

R: 66 G: 55 B: 114

Click Add Color to accept the settings and to return to the Form.

🆍 Project	1 - Microsoft Visual Basic [design]			
Ele Edit y	ew Broject Format Debug Bun Query Djagram Tools Add-Ins Window Help			
3.10	- 同 📽 🖬 あ 印 配 時 12 P 🖡 🗉 😹 🕼 名 替 🛠 🕃 📋 0,0 🛛 อี11940 x 2475			
	0 9			
×	Project1 - Plugin1 ///configured)	Properties - PlugIn	1	×
General		PlugIn1 UserContro	1	•
k 🔛		Alphabetic Catego	rized	
A abi	Single Normal	(Name)	PlugIn 1	^
1771		AccessKeys	B -1	
		Aignable	Faise	
F 6		Appearance	I - JU Ealer	
10 IV		BackColor	84007237428	- -
		BackStude	1.0007237420	
		BorderStyle	0 - None	
ৰম ন		CanGetFoous	True	
An	· · · · · · · · · · · · · · · · · · ·	CloBehavior	1 - Use Region	
00		ClipControls	True	
🗀 🖻		ControlContainer	False	
		DataBindingBehavior	0 - vbNone	~
		<u></u>	1. T.	
		BackColor		

The background of the Form is now set.

Figure 21 The background color is set



Figure 22 Edit the color of the command buttons

Make Project	? 🗙
Save in: CustomDSO PlugIn 💌 🗲 🖪	• 🖬 🖬 •
File name: Project 1.ocx	ОК
	Cancel
	<u>H</u> elp
	Options

Figure 23 Make and register the ActiveX Control Project1



Figure 24 Open CustomDSO from the Analysis menu

Still, we may change the color of the buttons, too.

This time, we choose a pre-defined color and select Style=1-Graphical, to view the new color.

After both buttons have the same color, we save the project.

The ActiveX Control is ready to be created. From File-menu select Make Project1.ocx

The Make Project window prompts for the name of the ActiveX Control. Keep the default name and click OK.

The code and functions of this project are now compiled into the ocx-file and at the same time it registers the ActiveX Control in the system.

If no error came up, there is now a new ActiveX Control in the registry of the operating system that can be called by WaveMaster.

On WaveMaster open the menu Analysis and highlight CustomDSO.



Figure 25 CustomDSO Setup tab



Figure 26 Click PlugIn on the CustomDSO tab and click into COM ProgID



Figure 27 Type in the name of our ActiveX Control

By default, the CustomDSO menu is off. Click now on PlugIn to bring up the PlugIn menu.

In the PlugIn menu you click in the control COM ProgID. This is where we type in the name of the ActiveX Control we just made and registered in the system.

Type in the exact name of the project and the control separated by a dot.



Figure 28 The name of the ActiveX Control is now in the ProgID control



Figure 29 The ActiveX Control is now plugged in



Figure 30 Present the CustomDSO PlugIn when menu closed

The ActiveX Control name is now entered in the control COM ProgID, ensuring same case is used.

Click on Install now to plug in the ActiveX Control.

If we typed in the correct name and the ActiveX Control is already registered in the system, we see the interface we created in Visual Basic merged or plugged into the scope interface.

In order to keep the PlugIn interface visible when the menus are closed, tick Present CustomDSO at powerup and when menu closed.

Close now the CustomDSO Setup tab by clicking on Close.



Even the menus are closed the PlugIn we created in this document is visible and accessible. Whenever we open context menus the PlugIn will be hidden and after the contect menus are closed the PlugIn is again on screen.

Figure 31 The PlugIn stays visible if menus are closed

Hint:

Occasionally, the development environment will not be installed on the WaveMaster but on another PC. In such cases the ActiveX Control needs to be ported from this PC to the instrument and registered. Follow this installation procedure to register the ocx-file on the instrument:

- copy the ocx-file to c;\ on the instrument in this case the file is called Project1.ocx
- open the DOS window
- execute regsvr32 c:\Project1.ocx. Alternatively just type regsvr32 c:\Project1.ocx into the Start/Run... window.

Obviously, it is not the intention of this document to provide a guide to develop a real world application CustomDSO PlugIn but to show how simple and fast a CustomDSO PlugIn for WaveMaster can be developed in Visual Basic.