

VCI - Virtual CAN Interface

How to use in LabWindows/CVI

IXXAT

Headquarter

IXXAT Automation GmbH
Leibnizstr. 15
D-88250 Weingarten

Tel.: +49 (0)7 51 / 5 61 46-0
Fax: +49 (0)7 51 / 5 61 46-29
Internet: www.ixxat.de
e-Mail: info@ixxat.de

US Sales Office

IXXAT Inc.
120 Bedford Center Road
USA-Bedford, NH 03110

Phone: +1-603-471-0800
Fax: +1-603-471-0880
Internet: www.ixxat.com
e-Mail: sales@ixxat.com

Support

In case of unsolvable problems with this product or other IXXAT products please contact IXXAT in written form by:

Fax: +49 (0)7 51 / 5 61 46-29
e-Mail: support@ixxat.de

Copyright

Duplication (copying, printing, microfilm or other forms) and the electronic distribution of this document is only allowed with explicit permission of IXXAT Automation GmbH. IXXAT Automation GmbH reserves the right to change technical data without prior announcement. The general business conditions and the regulations of the license agreement do apply. All rights are reserved.

1	Overview	5
2	System requirements	6
3	The VCI under LabWindows/CVI	7
	3.1 Creating an ActiveX Controller for the VCI.....	7
	3.1.1 Starting ActiveX Controller Wizard	7
	3.1.2 Selection of the VCI-Wrapper as ActiveX Server.....	8
	3.1.3 Saving the ActiveX Controller	8
	3.1.4 Advanced options	9
	3.1.5 Generation of ActiveX Controller completed	9
	3.2 Using the VCI-Wrapper ActiveX Controller	11
	3.2.1 Definition of a user interface	11
	3.2.1.1 Create new user interface	11
	3.2.1.2 Creating a control element	12
	3.2.1.3 Configuring a control element	13
	3.2.1.4 Code generation	14
	3.2.2 Inserting VCI-Wrapper function calls	15
	3.2.2.1 Selecting the type library	15
	3.2.2.2 Selecting the interface	15
	3.2.2.3 Selection of interface method	16
	3.2.2.4 Parameterization of the function call	17
	3.2.2.5 Subsequent modification of the function call	18
	3.2.3 Registering the Rx-Event Callback function.....	19
	3.2.3.1 Selecting the type library	19
	3.2.3.2 Selecting the Event Callback registration	19
	3.2.3.3 Selecting the Event interface.....	20
	3.2.3.4 Selecting the registration function for the Event.....	20
	3.2.3.5 Parameterization of the registration call-up.....	21
	3.3 Example application VCI Demo	22

1 Overview

The Virtual CAN Interface (VCI) is an application programming interface (API) for all IXXAT CAN-interfaces.

The VCI-API is a function interface, i.e. the user is provided with functions exported from the VCI-DLL.

It is theoretically possible under LabWindows/CVI to work directly on this VCI-API. However, this is somewhat awkward and complicated. We therefore recommend installing the VCI-Wrapper, which provides the VCI-API via a COM-interface as an ActiveX component. As an ActiveX, the VCI can be integrated very quickly and easily into a LabWindows/CVI project.

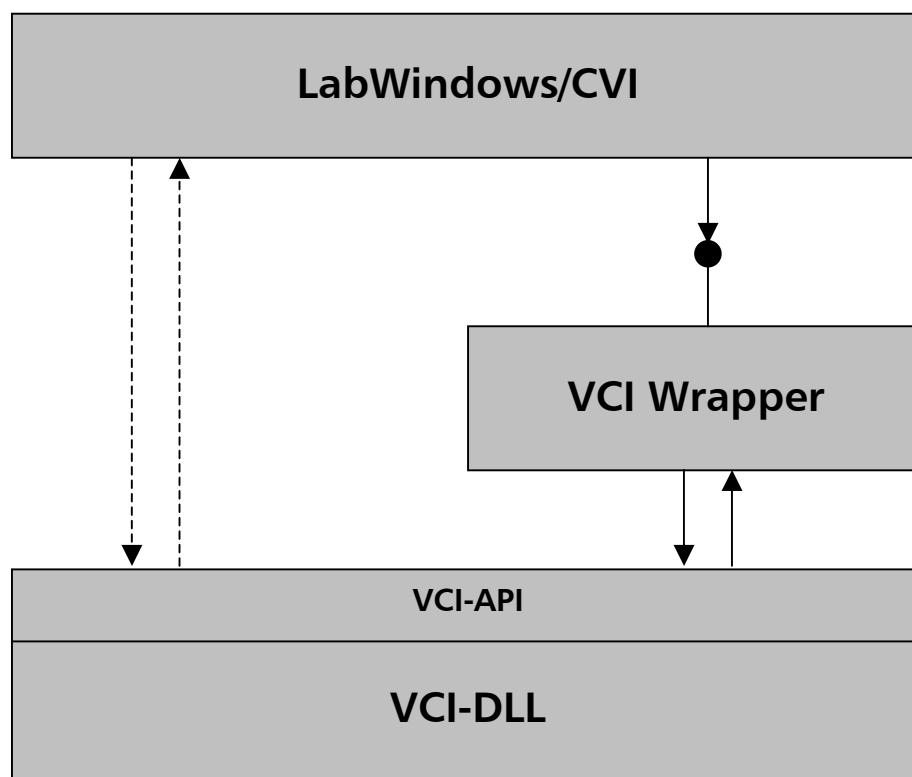


Figure 1-1: Overview

2 System requirements

- VCI V2.10 (or higher)
- VCI Wrapper V2.0

or

- VCI V2.14 (or higher – VCI Wrapper integrated)

3 The VCI under LabWindows/CVI

3.1 Creating an ActiveX Controller for the VCI

Before the VCI-Wrapper can be used in a LabWindows project, a so-called ActiveX Controller must be created as a free definable parameter.

3.1.1 Starting ActiveX Controller Wizard

Start the ActiveX Controller Wizard via Tools | Create ActiveX Controller...

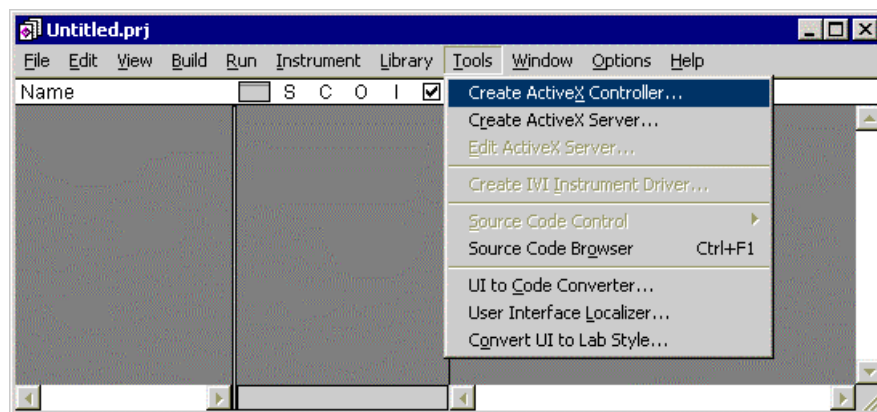


Figure 3.1-1: Starting ActiveX Controller Wizard

The following Figure shows the started ActiveX Controller Wizard. Continue by clicking on the "Next"-button.

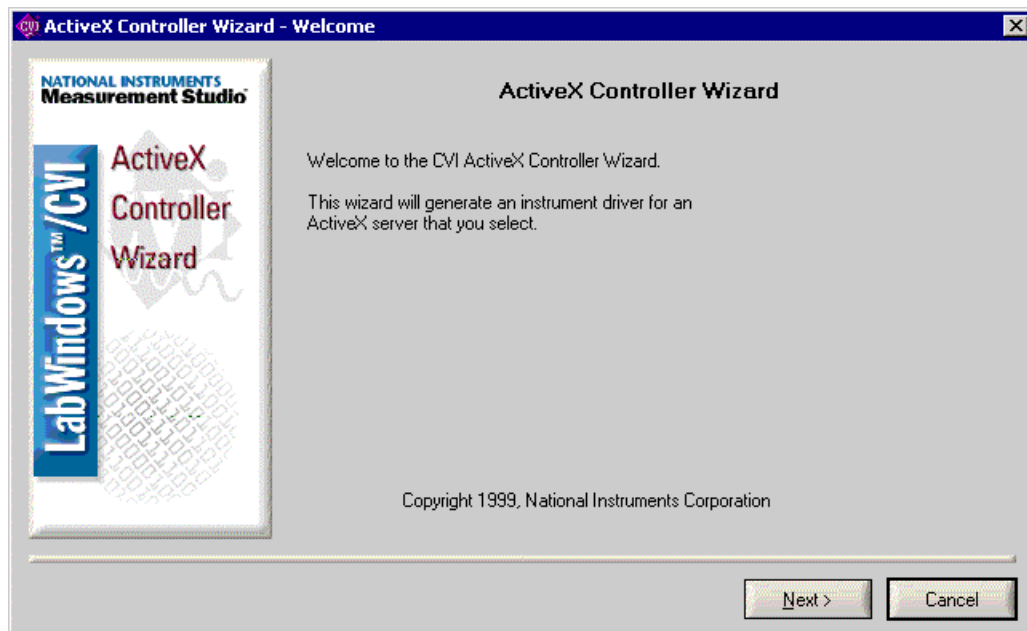


Figure 3.1-2: Start window ActiveX Controller Wizard

3.1.2 Selection of the VCI-Wrapper as ActiveX Server

Select the VCI-Wrapper as ActiveX Server for the ActiveX Controller and click on the "Next"-button.

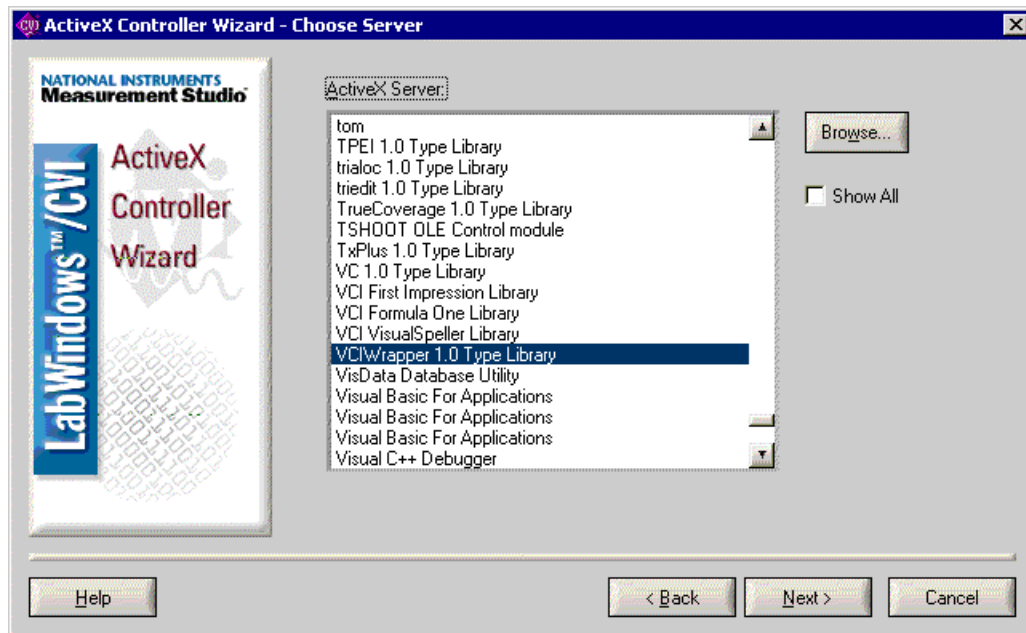


Figure 3.1-3: Selection of the VCI-Wrapper as ActiveX Server

3.1.3 Saving the ActiveX Controller

Under "Target .fp File:" enter the absolute path of the file under which you want to save the ActiveX Controller. Click on the "Next"-button.

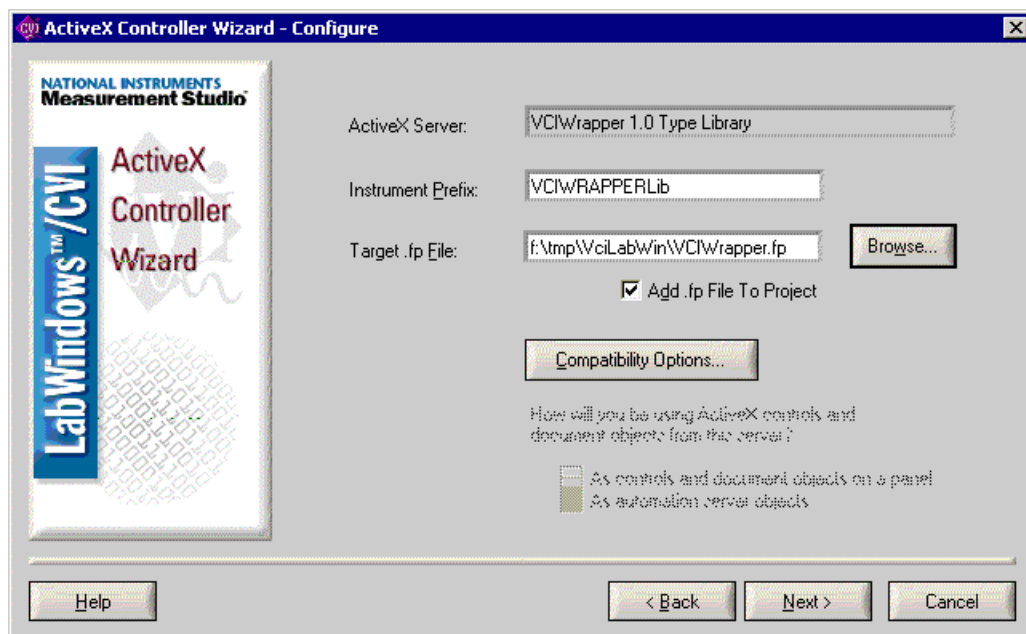


Figure 3.1-4: Saving the ActiveX Controller

3.1.4 Advanced options

You can skip the following dialog with the "Next"-button, as the VCI-Wrapper type library only contains one object.

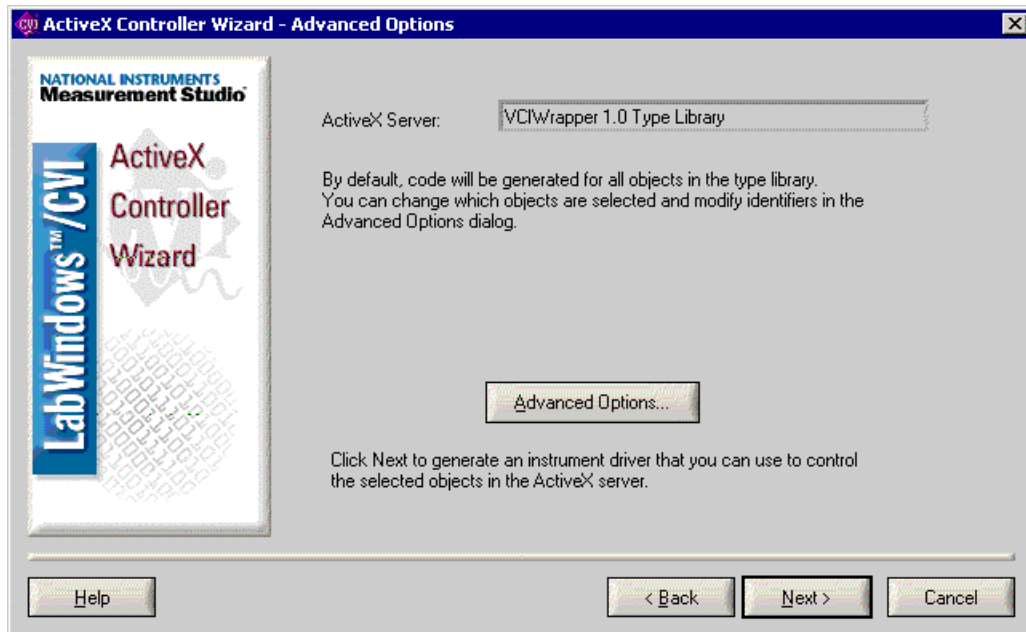


Figure 3.1-5: Advanced options

3.1.5 Generation of ActiveX Controller completed

The generation of the ActiveX Controller is now completed. Click on the "Close"-button.

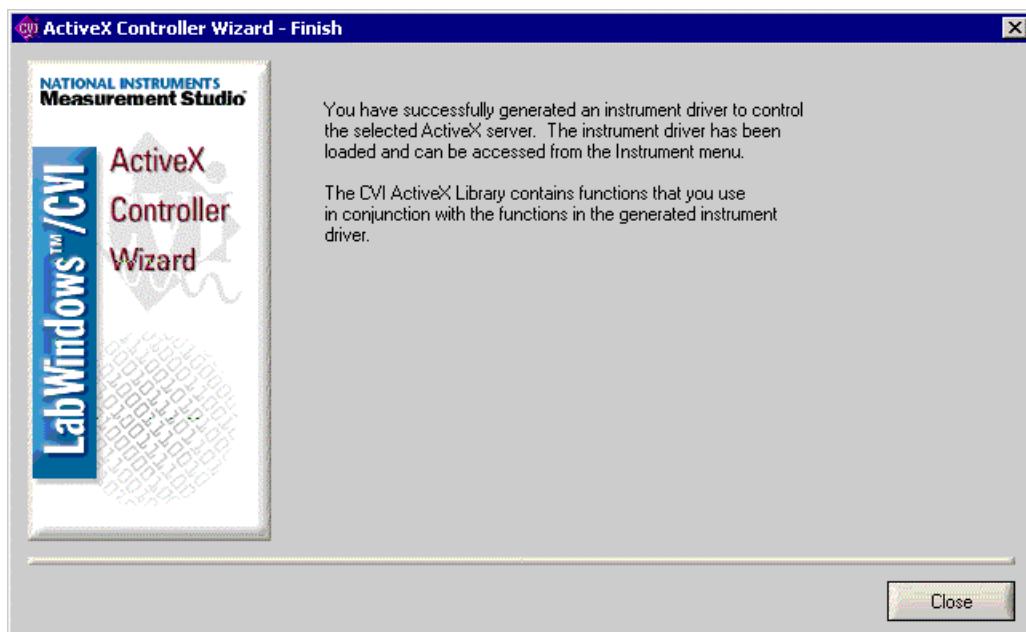


Figure 3.1-6: Generation of ActiveX Controller completed

The VCI under LabWindows/CVI

The ActiveX Controller now appears in the main window.

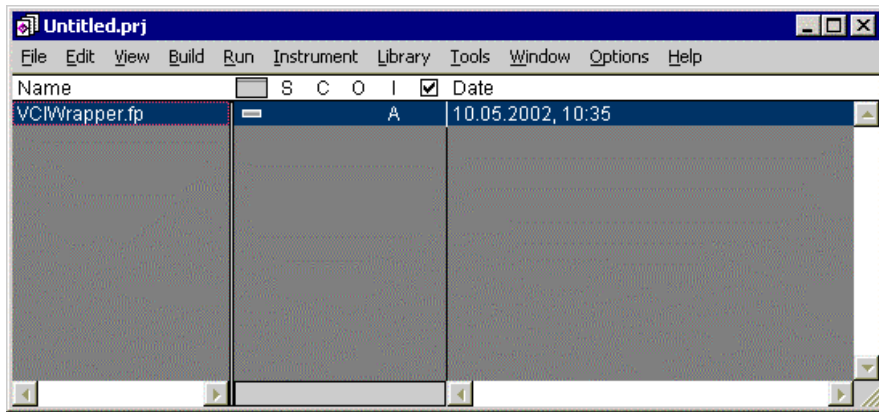


Figure 3.1-7: ActiveX Controller in the main window

3.2 Using the VCI-Wrapper ActiveX Controller

3.2.1 Definition of a user interface

3.2.1.1 Create new user interface

Create a new user interface via File | New | User Interface.

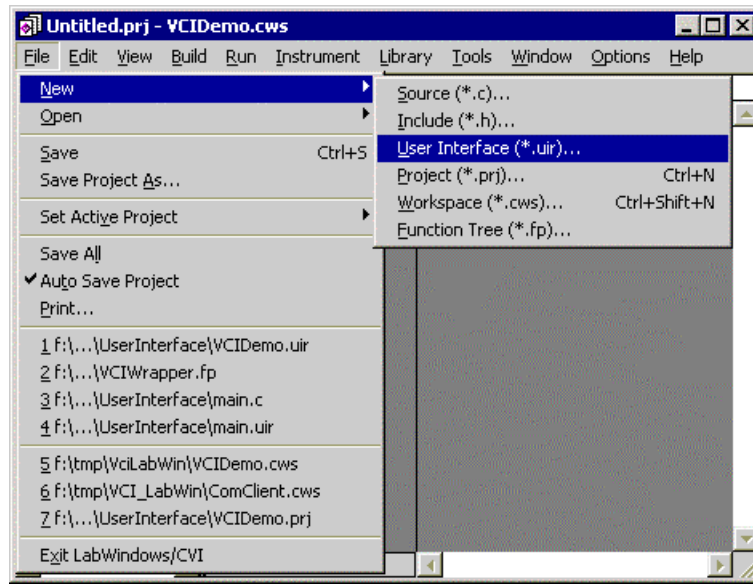


Figure 3.2-1: Creating a new user interface

A new window now appears with an as yet empty user interface.

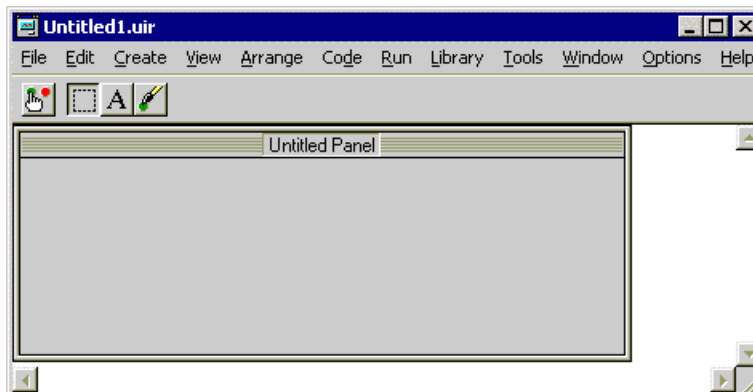


Figure 3.2-2: The new user interface

3.2.1.2 Creating a control element

Create a button, for example, via Create | Command Button.

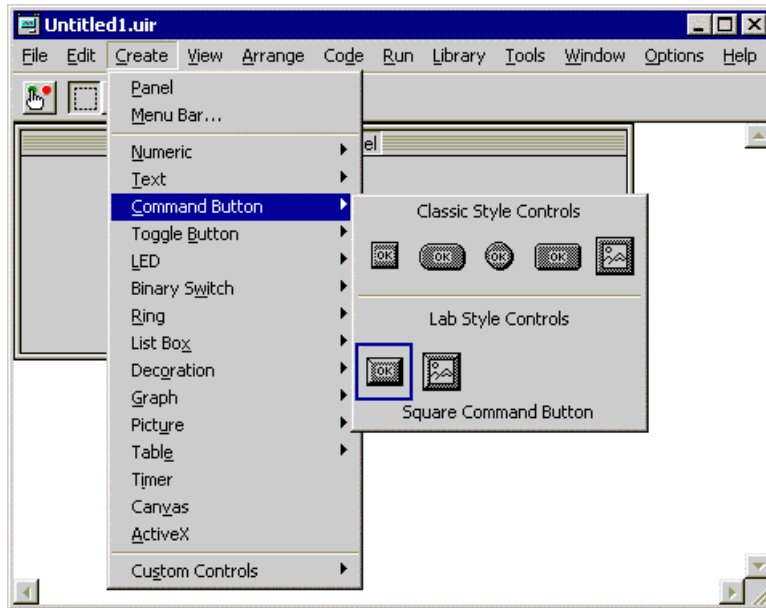


Figure 3.2-3: Creating a control element

LabWindows/CVI creates a new button for you on the user interface.

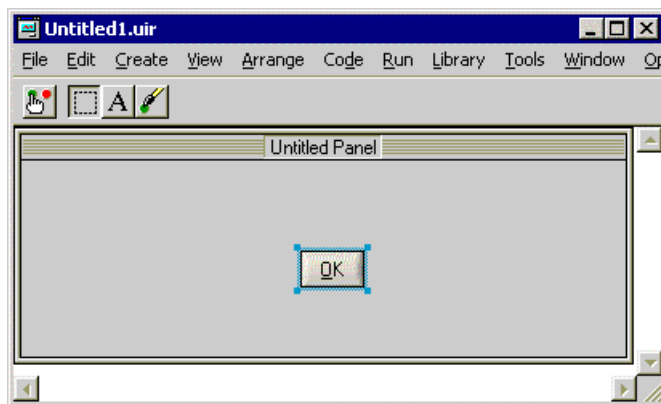


Figure 3.2-4: The new control element

3.2.1.3 Configuring a control element

The button is configured by double-clicking on it. Enter "Init Board, for example, as a label. Under "Callback Function", the name of a source code function (here "OnInitBoard") can be entered which is called up during runtime when the user clicks on the button.

The Callback function is later created automatically during code generation by LabWindows/CVI.

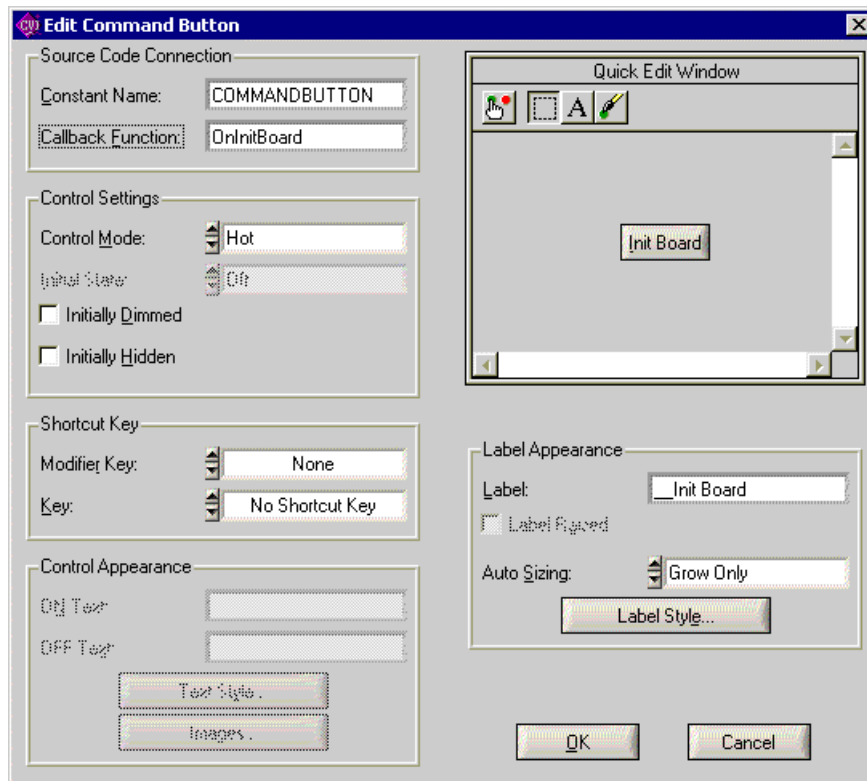


Figure 3.2-5: Configuring a control element

After acknowledging with the "OK"-button, the configured button appears.

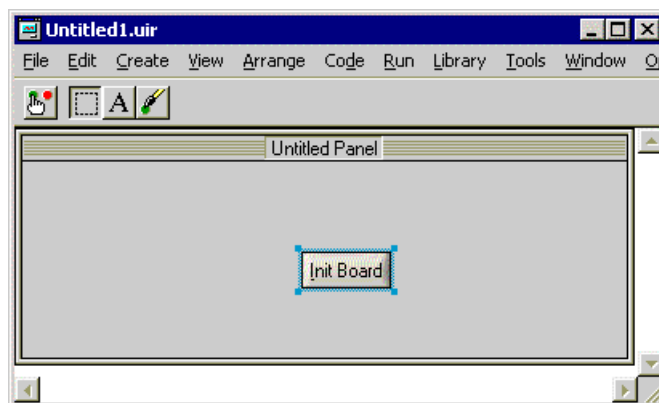


Figure 3.2-6: The configured control element

3.2.1.4 Code generation

Now start generation of the source code via Code|Generate|All Code.

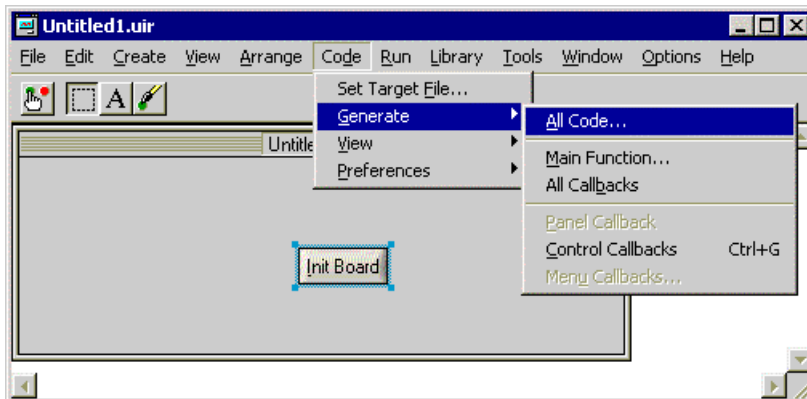


Figure 3.2-7: Code generation

In the following dialog the name of the source code file has to be entered. It is important to switch to "Add To Current Project" under "Target Files". Otherwise LabWindows/CVI creates a new project and closes the current one.

Acknowledge the dialog with the "OK"-button.

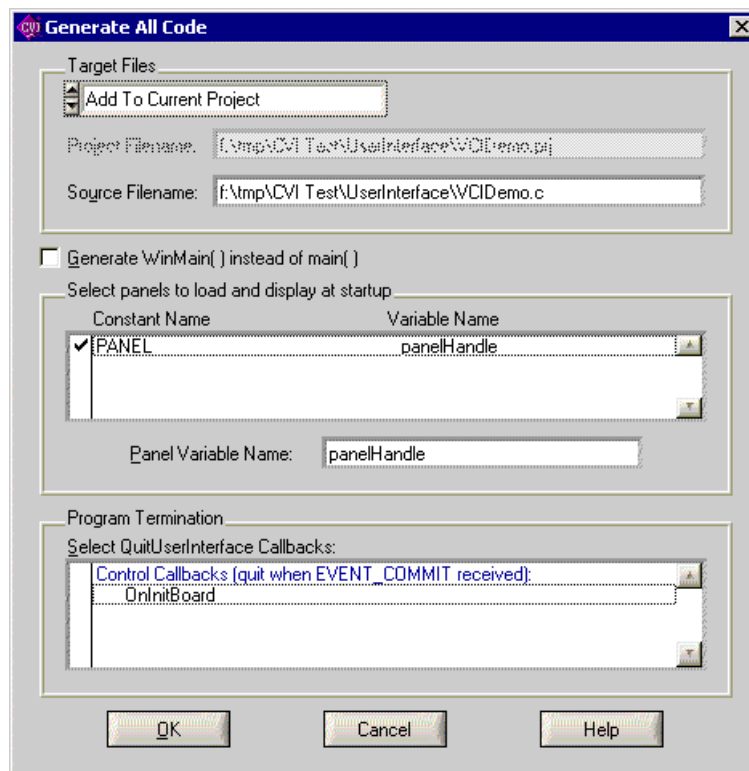


Figure 3.2-8: Name of the source file

3.2.2 Inserting VCI-Wrapper function calls

3.2.2.1 Selecting the type library

After successful generation of the source code, the following source code window appears. Place the cursor in the line in which you want to insert the call of a VCI-Wrapper method. Select the VCI-Wrapper library in the Instrument menu.

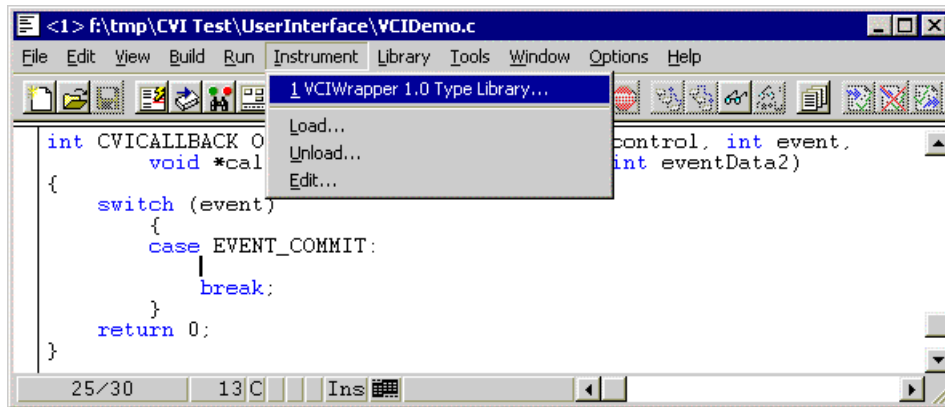


Figure 3.2-9: Selecting the type library

3.2.2.2 Selecting the interface

In the following dialog the interface that contains the required method must be selected. Select either by double-clicking on the interface IVCIWrapper or mark the interface and continue via the "Select"-button.

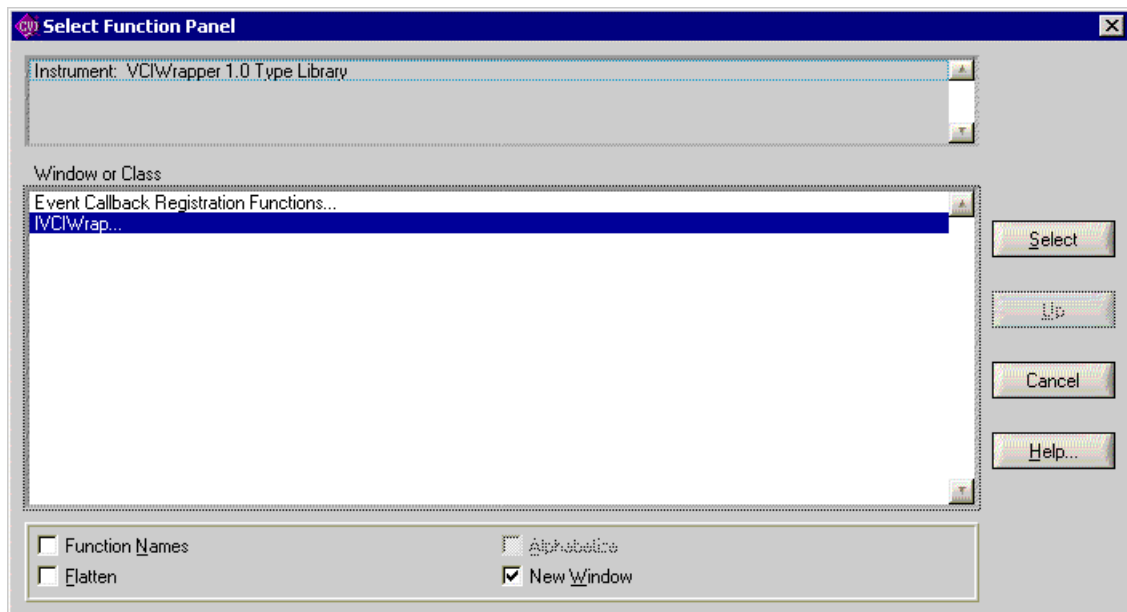


Figure 3.2-10: Selecting the interface

3.2.2.3 Selection of interface method

Now select the interface method that you would like to call up. Select either by double-clicking on the method or mark the method and continue via the “Select”-button.

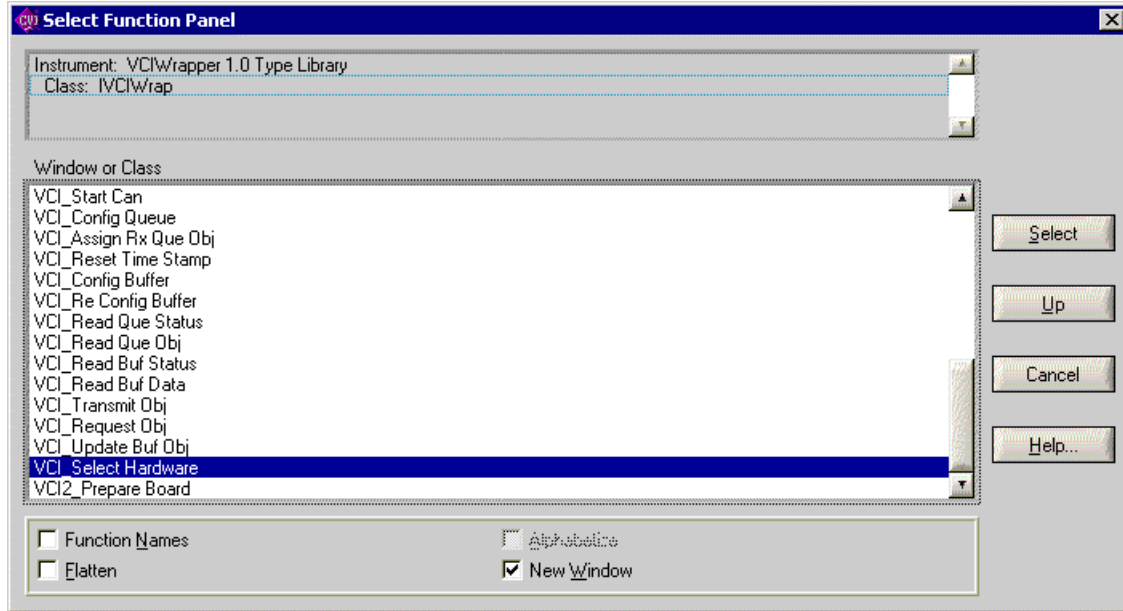


Figure 3.2-11: Selection of interface method

3.2.2.4 Parameterization of the function call

In the following dialog it is possible to parameterize the function call to be inserted. You can enter a variable name, complete arguments or also constants for each parameter. If you do not yet know these, you can leave these fields unchanged or empty.

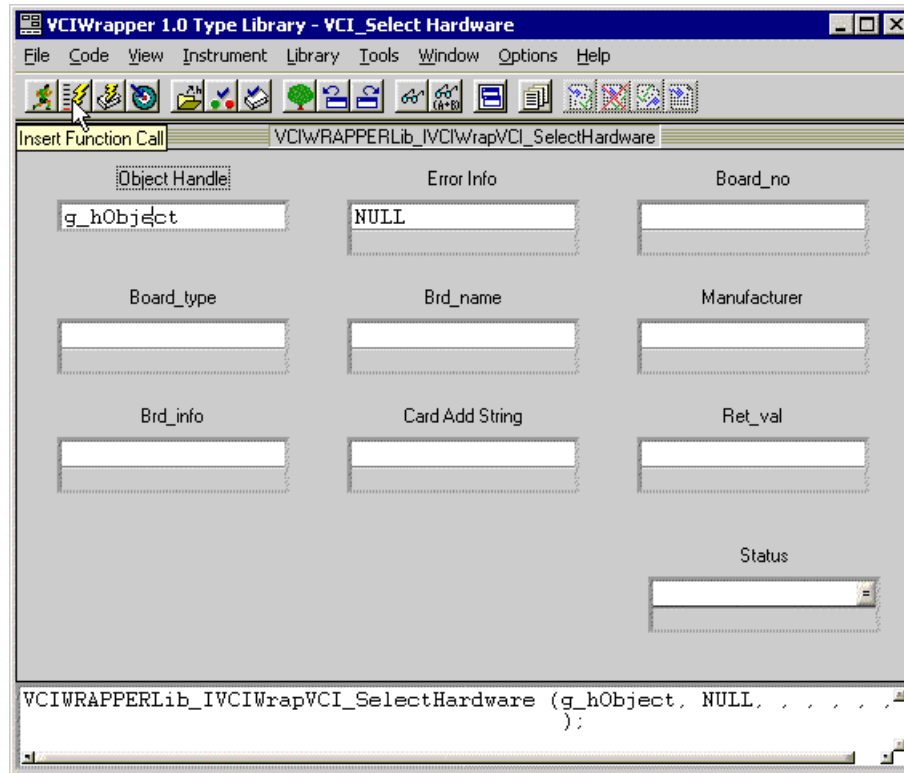


Figure 3.2-12: Parameterization of the function call

Now click on the button "Insert Function Call" in the toolbar. The function call is then inserted in the previously marked line.

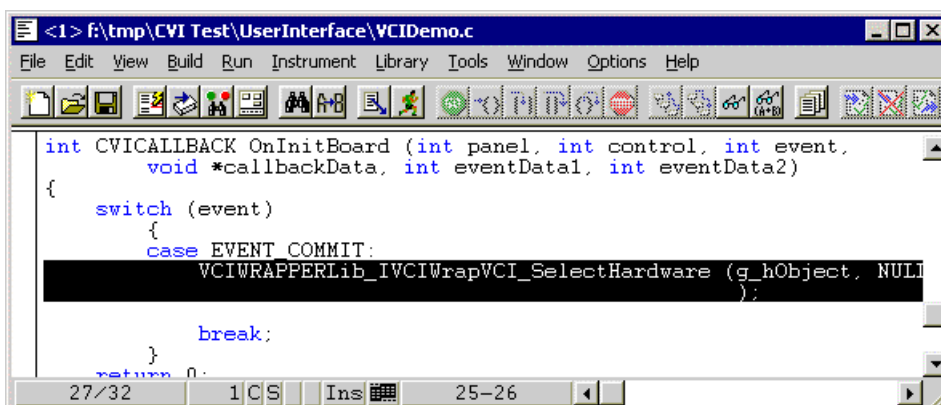


Figure 3.2-13: The new function call

3.2.2.5 Subsequent modification of the function call

The function call can be subsequently modified. It is possible to alter the call manually in the source code or to open the “Function Panel” by clicking with the right-hand mouse button on the function call in the source code. There the parameters can be modified individually in the overview.

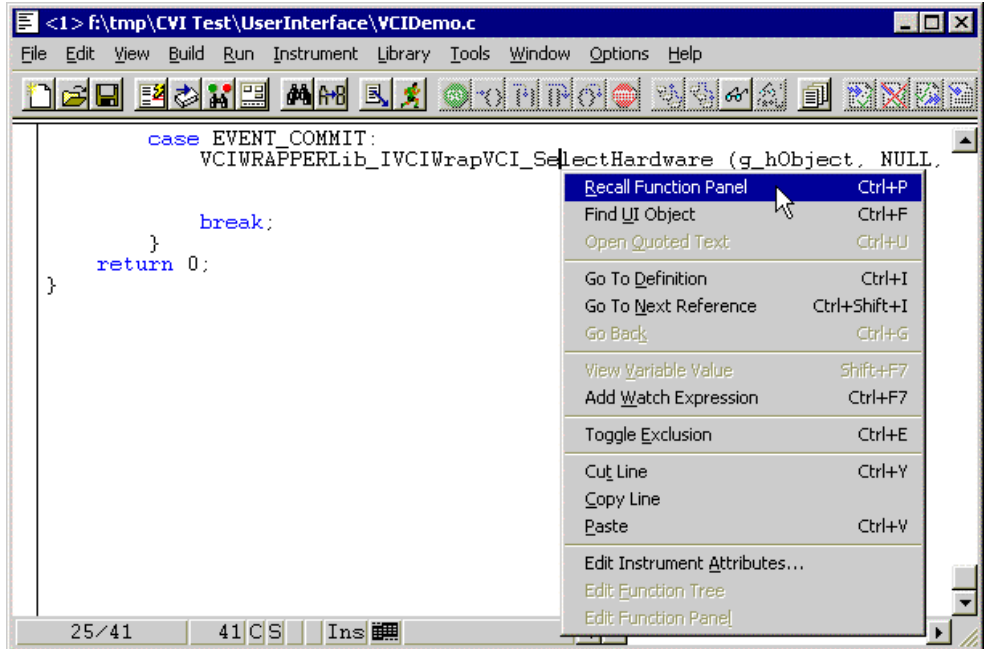


Figure 3.2-14: Modification of the function call

3.2.3 Registering the Rx-Event Callback function

The Event Callback function is registered via the “Function Panel”. The Event Callback function itself must be implemented manually.

3.2.3.1 Selecting the type library

Place the cursor in the source code window in the line in which you want to insert an Event Callback registration. Select the VCI-Wrapper library in the Instrument menu.

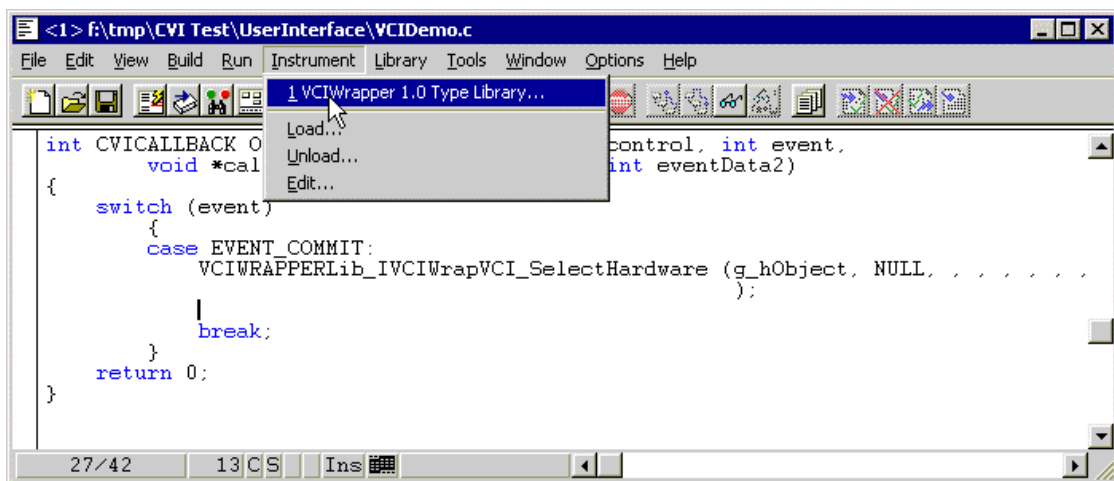


Figure 3.2-14: Selecting the type library

3.2.3.2 Selecting the Event Callback registration

Select “Event Callback Registration Functions...” in the following dialog.

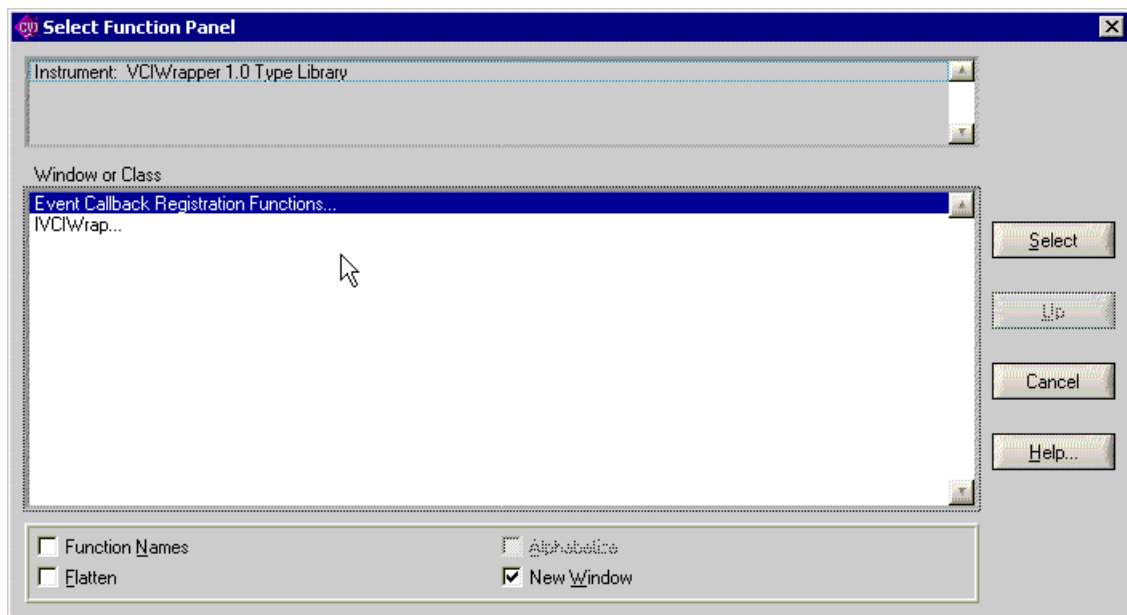


Figure 3.2-15: Selecting the Event Callback registration

3.2.3.3 Selecting the Event interface

Now select the Event interface that contains the Event for which you want to register a Callback Function. Only the Event interface IVCIWrapEvents exists for the VCI-Wrapper.

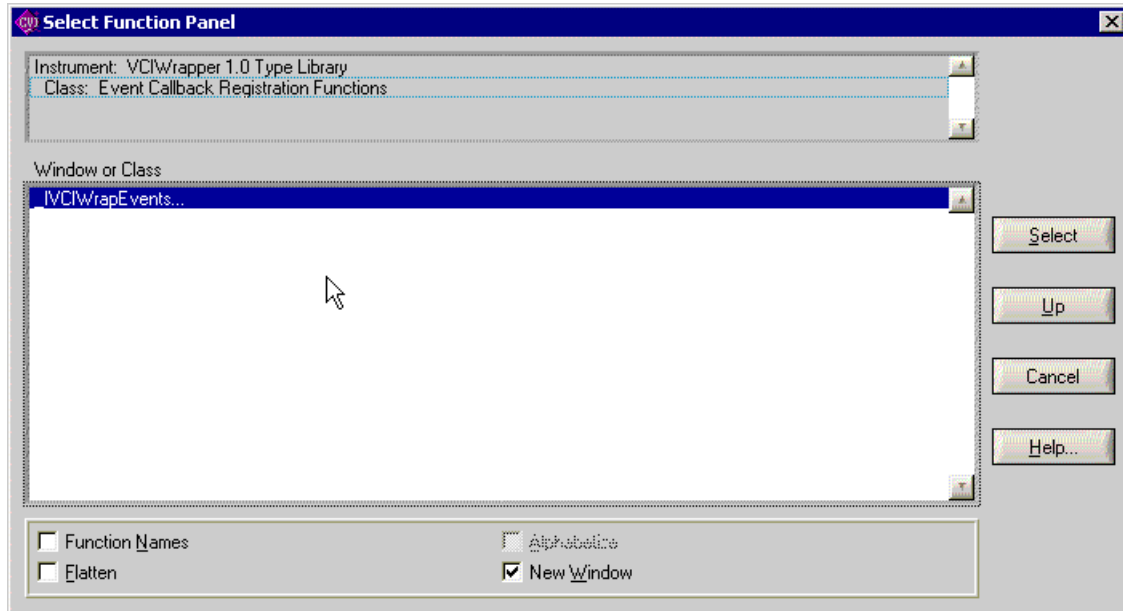


Figure 3.2-16: Selecting the Event interface

3.2.3.4 Selecting the registration function for the Event

Now select the registration function for the corresponding Event. The VCI-Wrapper provides only one VCIRxEvent.

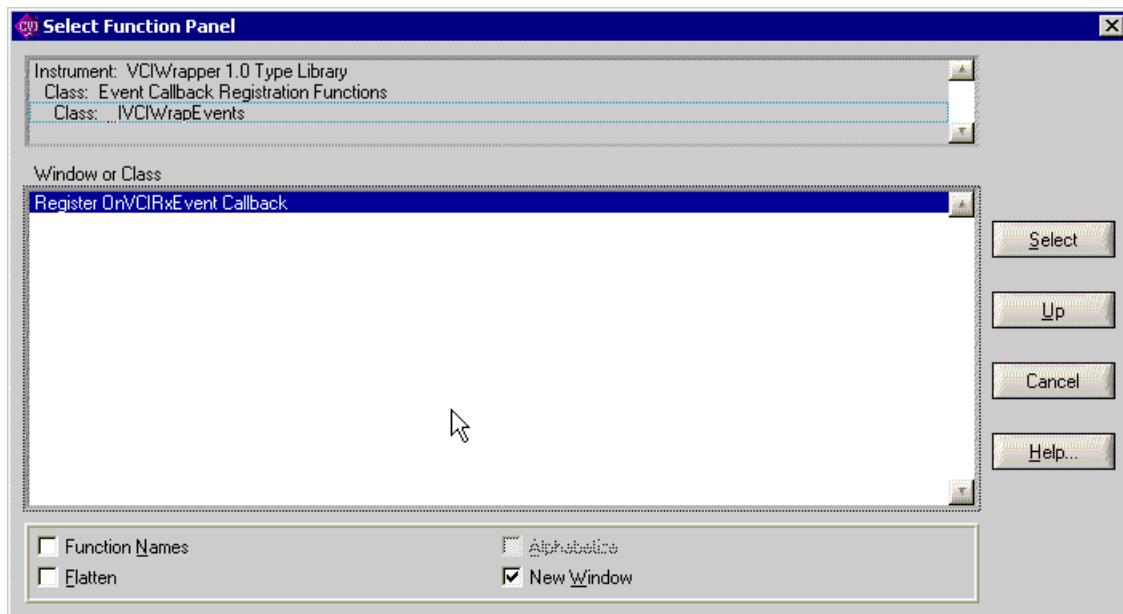


Figure 3.2-17: Selecting the registration function

3.2.3.5 Parameterization of the registration call-up

For the parameterization of the Registration call-up, now enter the name of the Event Callback Function that you later implement manually (here "OnRxEvent").

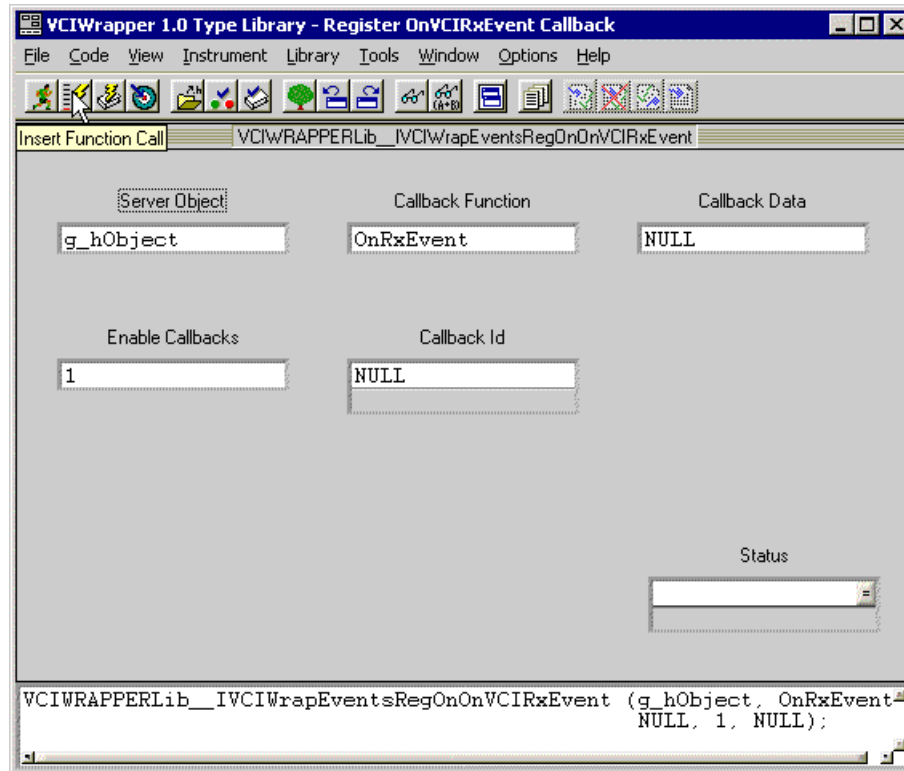


Figure 3.2-18: Parameterization of the registration call-up

Now click on the button "Insert Function Call" in the toolbar. The registration call-up is then inserted in the previously marked line.

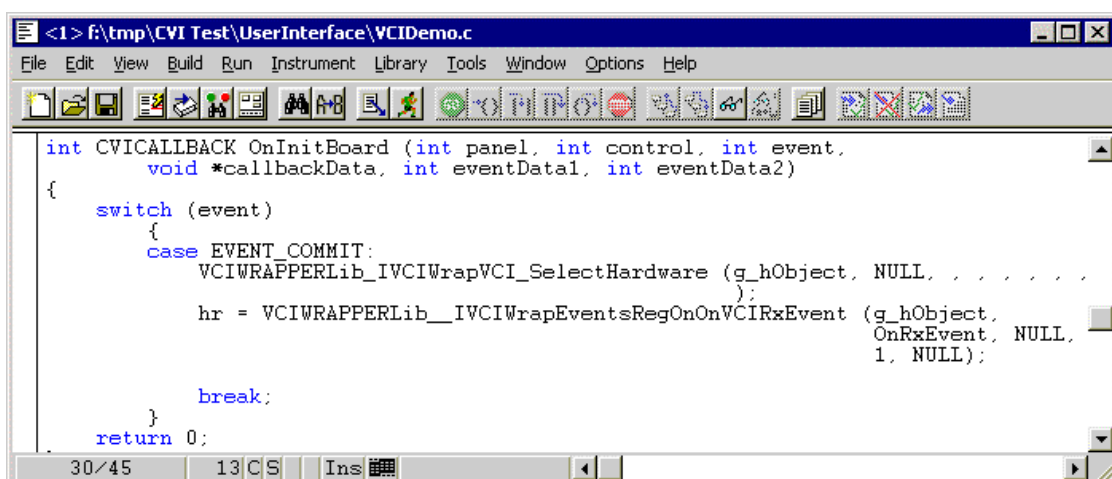


Figure 3.2-19: The registration call-up

3.3 Example application VCI Demo

The example application VCI Demo is a complete example of a VCI integration into LabWindows/CVI.

The project VCI Demo is included with complete source code.

ATTENTION:

LabWindows/CVI works with absolute paths. In order to translate the project without problems, it may be necessary to individually adapt the absolute path information in the files VCIDemo.cws and VCIDemo.prj.

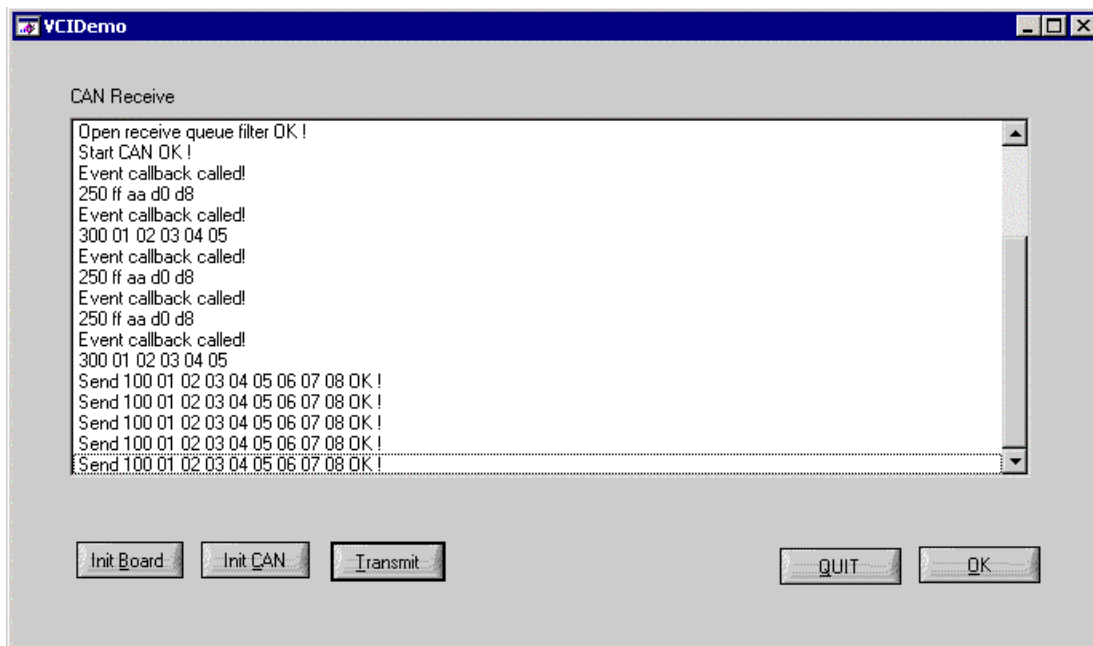


Figure 3.3-19: Example application VCI Demo

VCIDemo displays the received CAN telegrams in the list box. Every time the "Transmit"-button is activated, a telegram is transmitted with ID 100 and 8 data bytes.