Table of Contents

Chapter 1 What is TCPDAQ ActiveX Control?

TCPDAQ.OCX is a collection of ActiveX controls for performing I/O operations within any compatible ActiveX control container, such as Visual Basic, Delphi, etc. You can easily perform the I/O operations through properties, events and methods. Specific information about the properties, methods, and events of the TCPDAQ ActiveX controls can be found later in this manual.

With TCPDAQ ActiveX Control, you can perform versatile I/O operations to control your Inlog EDAM-9000 module series.

The TCPDAQ ActiveX Control setup program installs TCPDAQ.OCX through a process that may take several minutes. Installing the necessary software to use the TCPDAQ.OCX in your application involves two main steps:

Installing the TCPDAQ ActiveX Control

Using the Inlog EDAM-9000 utility to configure the modules that is attached to your computer.

You can use these ActiveX controls in any development tool that supports them, including Microsoft Visual C++, Microsoft Visual Basic, Borland C++ Builder, Borland Delphi

1.1 Installing the TCPDAQ ActiveX Controls

Before using the TCPDAQ ActiveX Control, you must install the TCPDAQ.OCX first

- Insert the TCPDAQ installation CD-ROM disc into your computer.
- The installation program should start automatically. If autorun is not enabled on your computer, use your Windows Explorer or the Windows Run command to execute Setup.exe on the TCPDAQ installation CD-ROM disc (assume "d" is the letter of your CD-ROM disc drive):

  D: \Setup.exe

Chapter 2 Building TCPDAQ ActiveX Control with Various Tools

This chapter describes how you can use the TCPDAQ ActiveX Control with the following development tools:

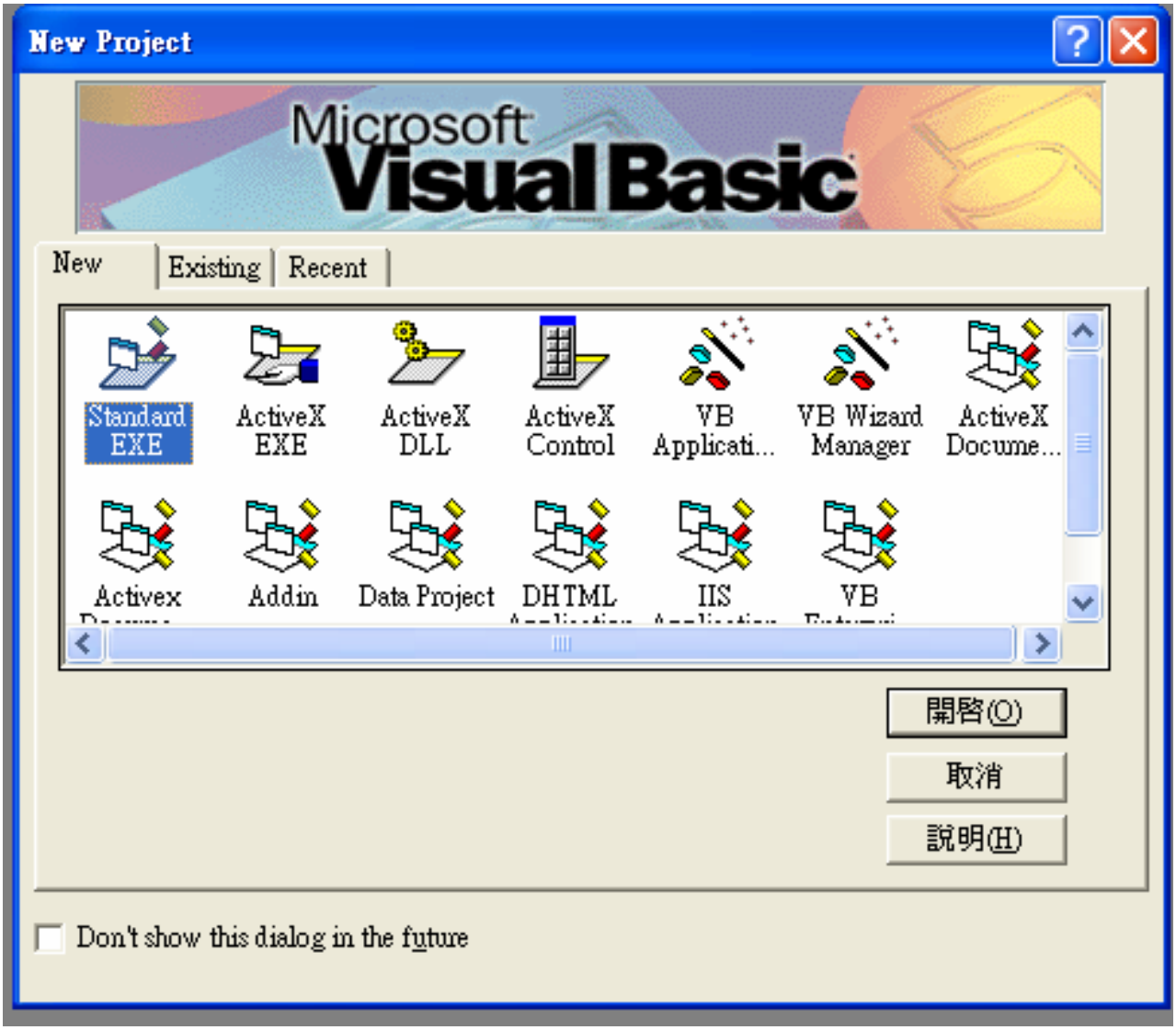
- Microsoft Visual C++ version 6.0 (SP5)
- Microsoft Visual Basic version 6.0 (SP5)
- Borland Delphi version 4.0 (with the Delphi 6 Update Pack fixes for ActiveX installed)
- Borland C++ Builder version 5.0

This chapter assumes that you are familiar with the basic concepts of using Visual Basic, Delphi, Borland C++ Builder, and Visual C++, including selecting the type of application, designing the form, placing the control on the form, configuring the properties of the control, creating the code (event handler routines) for this control.

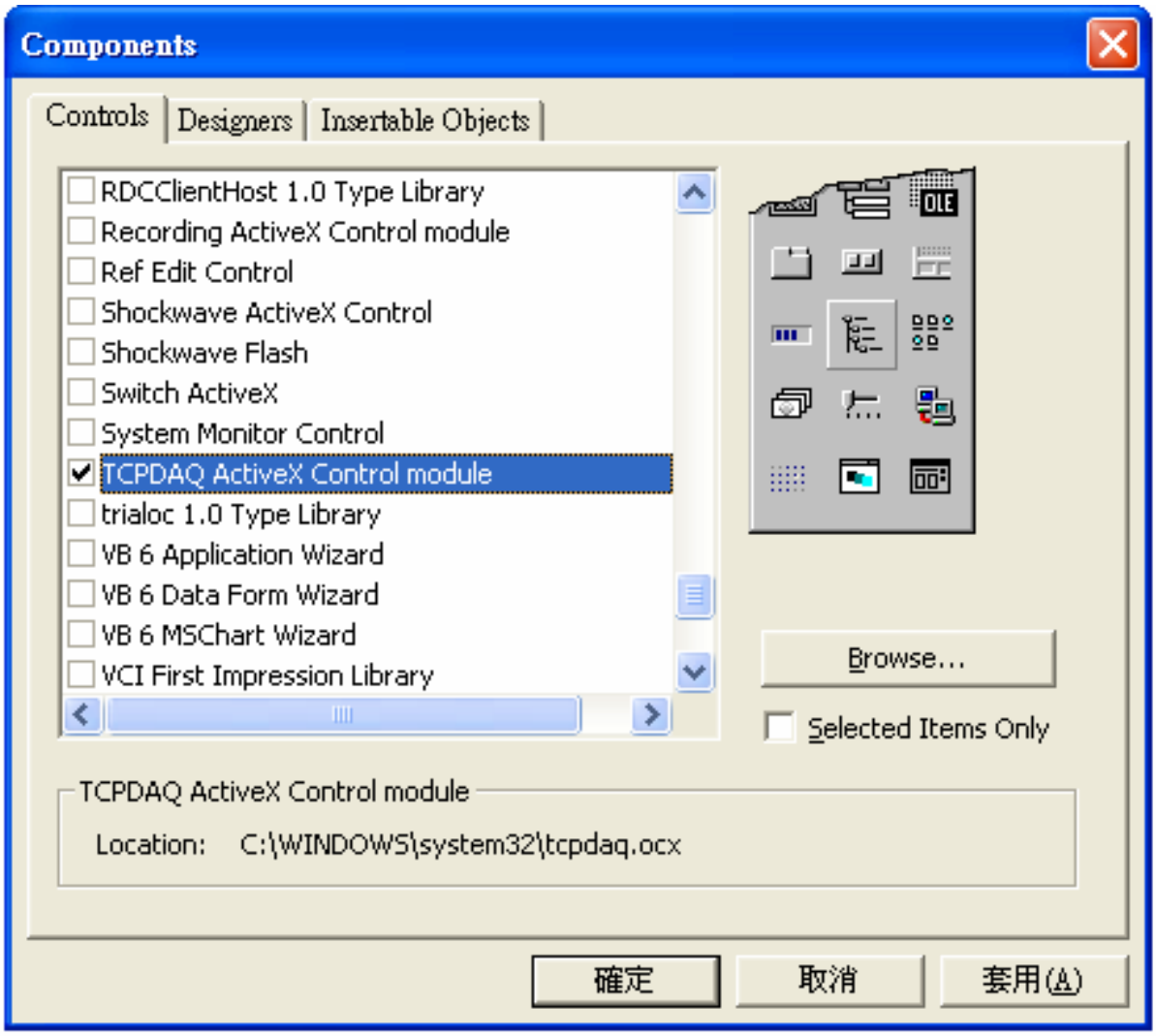
Note: For Borland Delphi 6, the Delphi 6 Update Pack fixes for ActiveX must be installed.

2.1 Building TCPDAQ Applications with Visual Basic

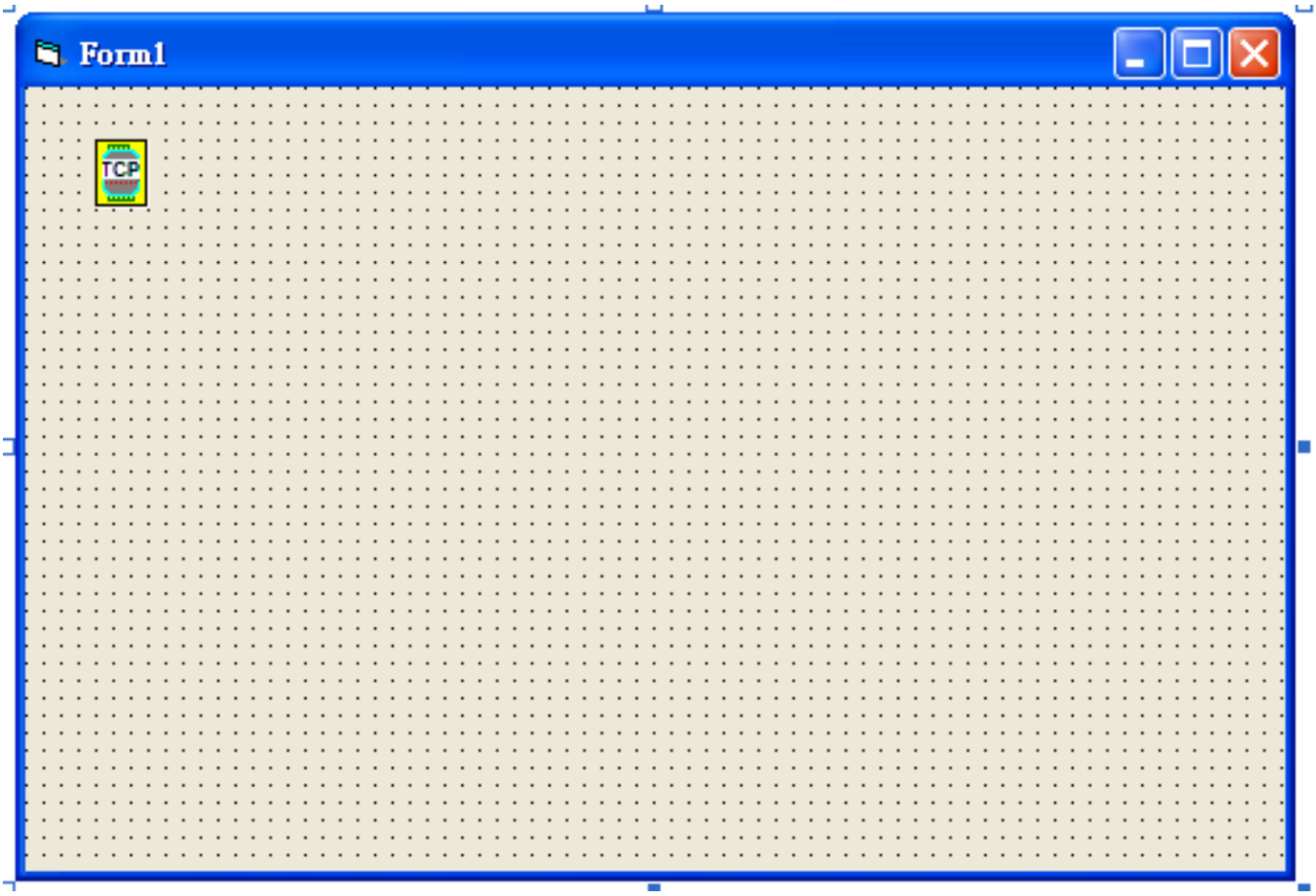
- Start Visual Basic.



- Select **Standard EXE** icon and press the **Open** button. A new project is created. Click on **Components...** from the **Project** menu. The Components dialog box is loaded as shown below:

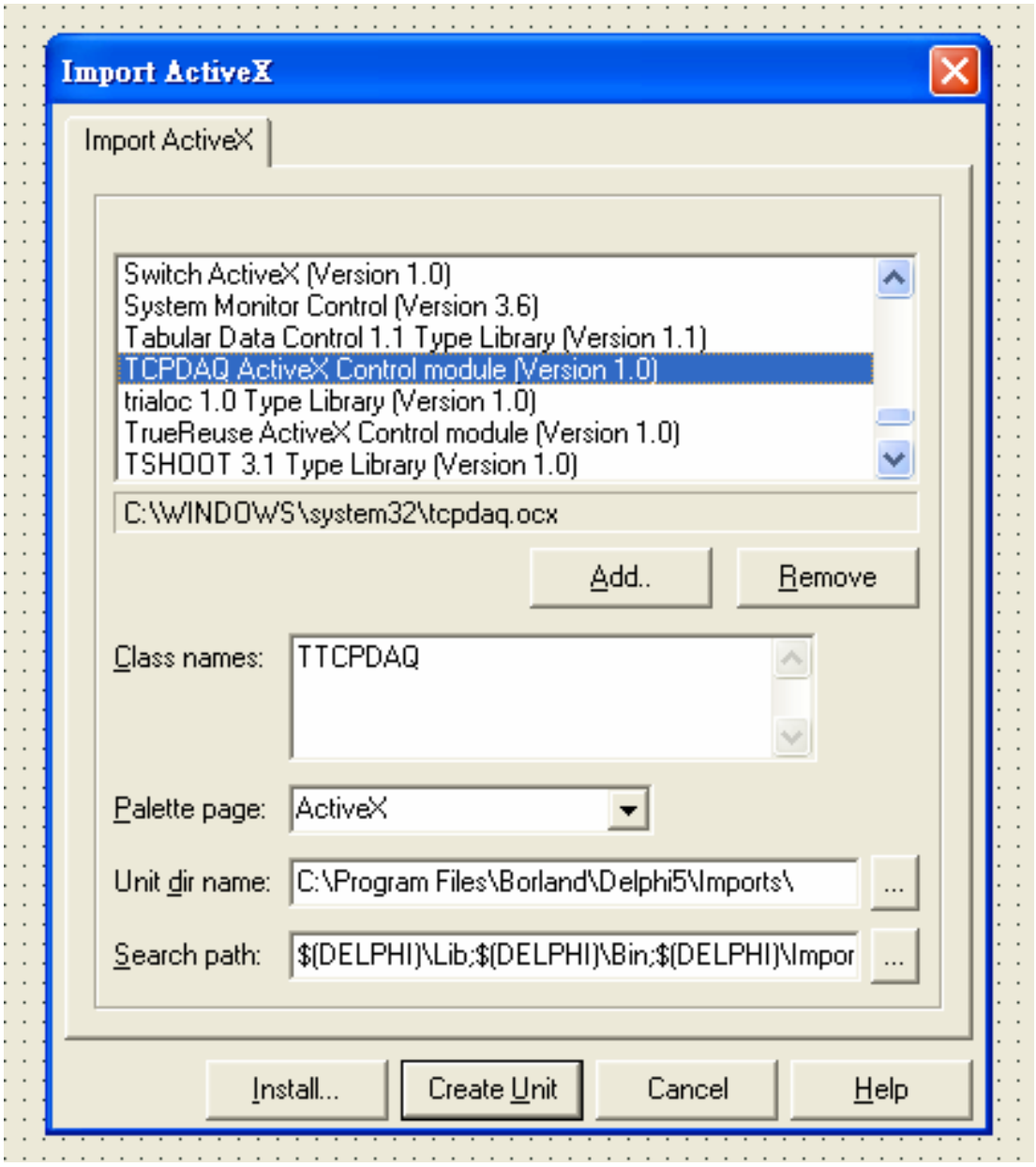


- Place a **TCPDAQ** control from the Toolbox on the form. Use the default name.
- Your form should look similar to the one shown below:

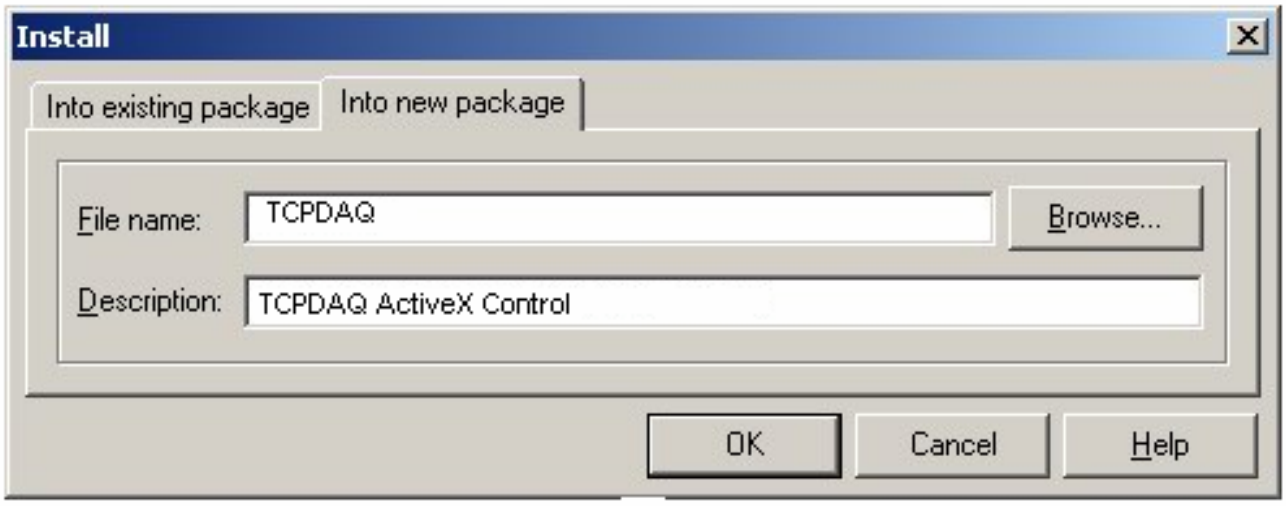


2.2 Building TCPDAQ Applications with Delphi

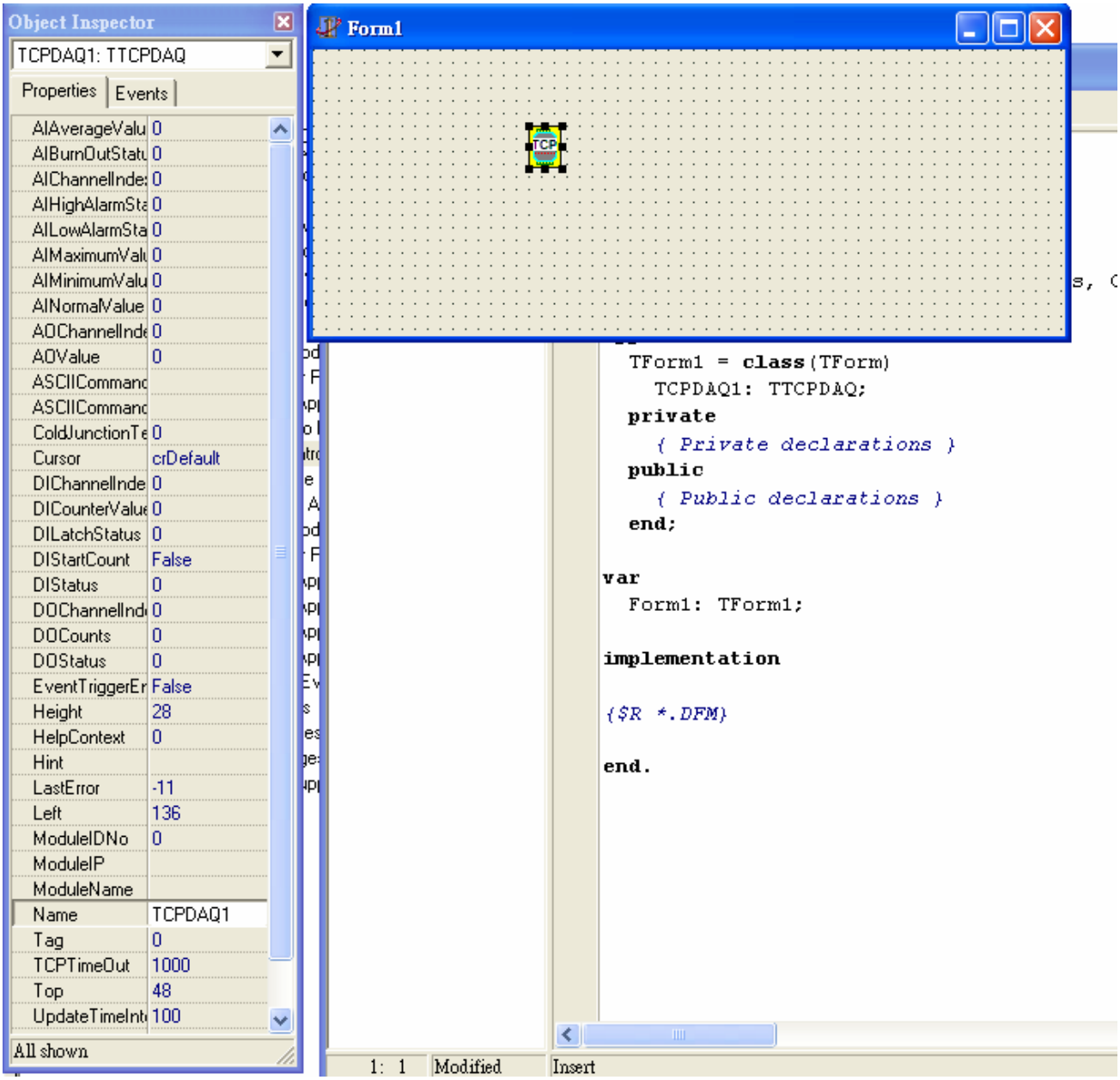
- Start Delphi, Delphi will launch as shown below:
- Select **Import ActiveX Control...** from the **Component** menu. The Import ActiveX dialog box loads:
- Select the **TCPDAQ ActiveX Control Module** and press the **Install...** button. A dialog box is displayed as follows:



- The **TCPDAQ control** is loaded into the **Component Palette**. You can check it by clicking on **Install Package...** from the **Component** menu. A dialog box is shown as below.

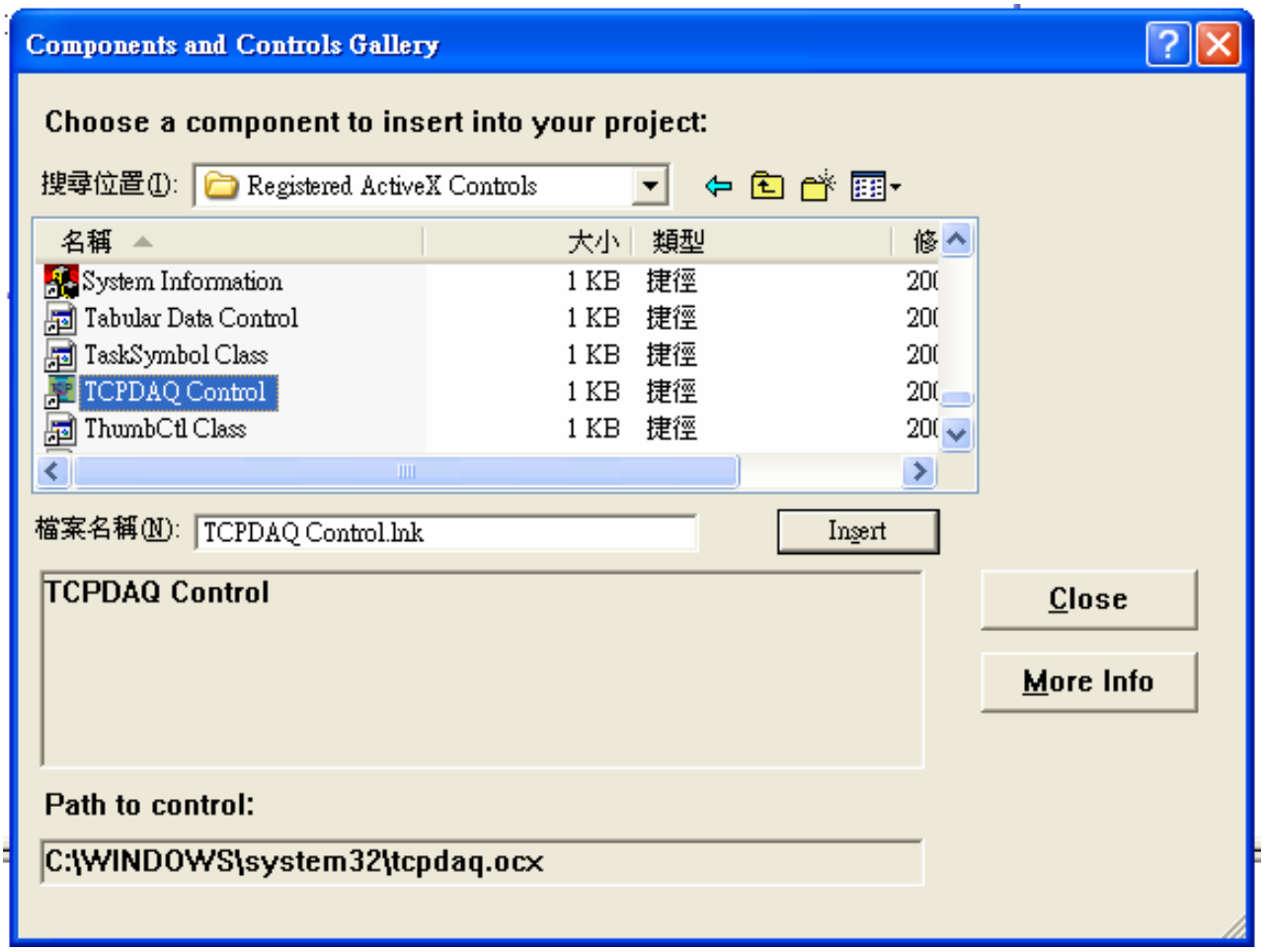


- Switch to the form and select the ActiveX tab from the **Component Palette**.
- Place a **TCPDAQ** control from the **Component Palette** on the form. Use the default names TCPDAQ1.
- Your form should look similar to the one shown below:

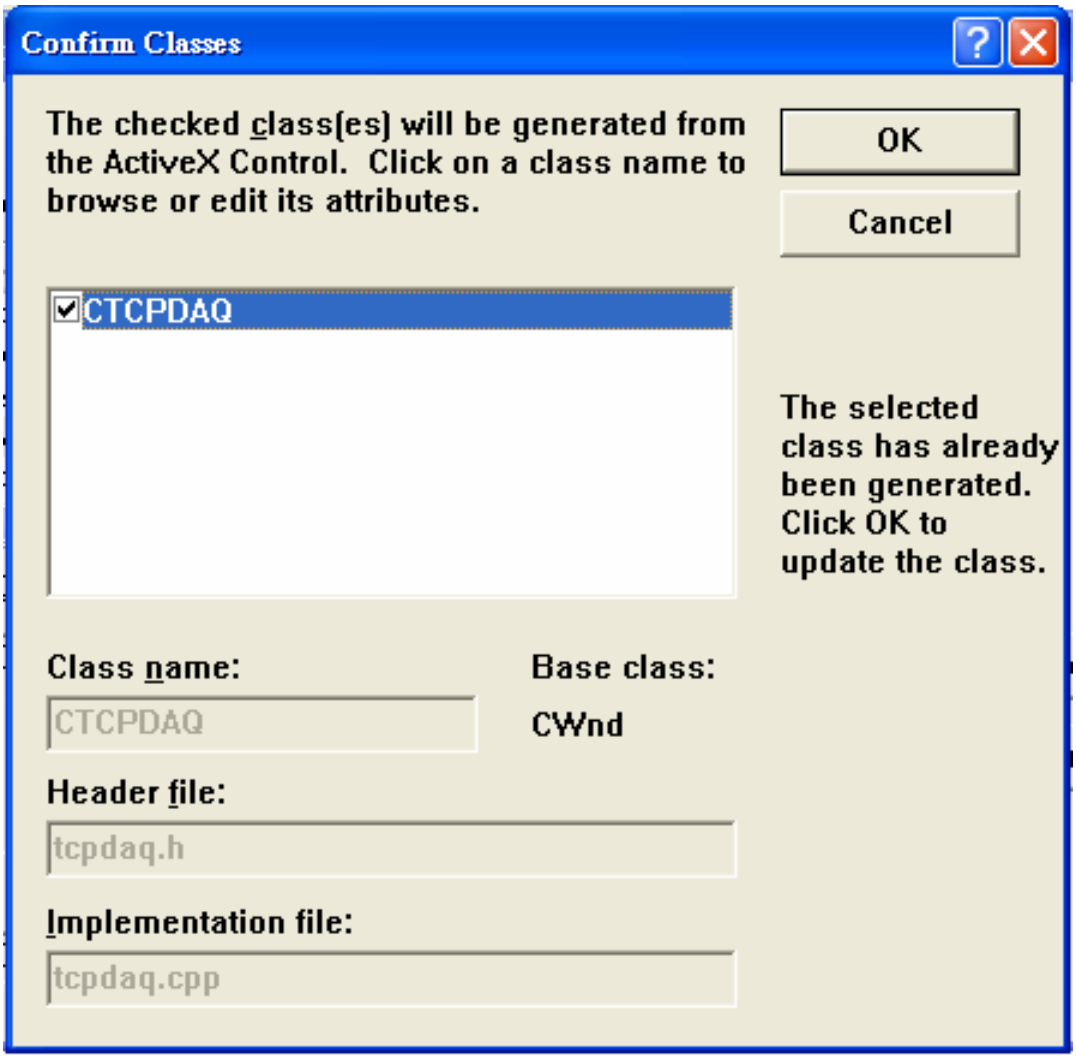


2.3 Building TCPDAQ Applications with Visual C++

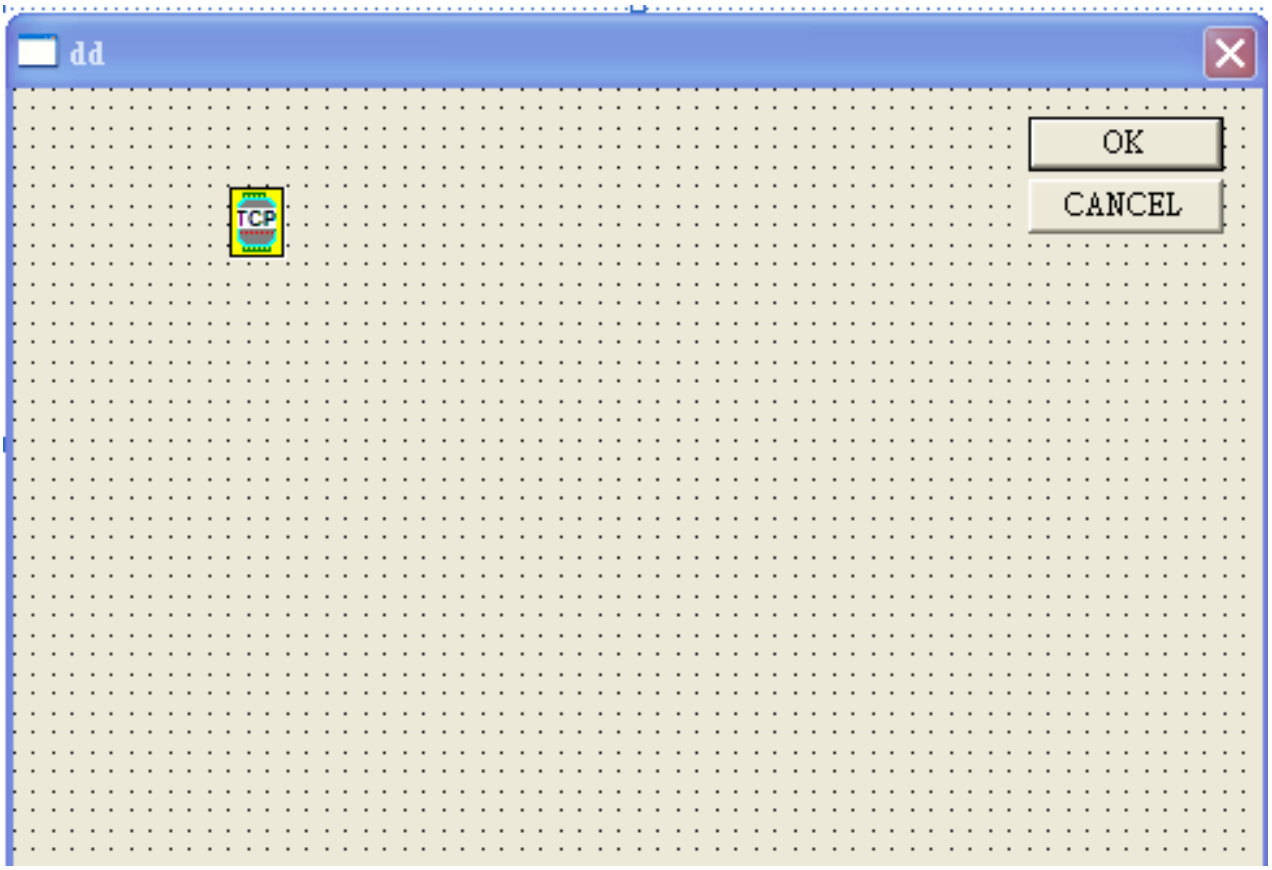
- Start Visual C++ program.
- Select **Add to Project...** -> **Components and Controls** from the **Project** menu, and double-click on **Registered ActiveX Controls**. The result should be as below:



- Scroll down to the **TCPDAQ Control** and press the **Insert** button. A Class Confirm dialog box is displayed, Press **OK** button.

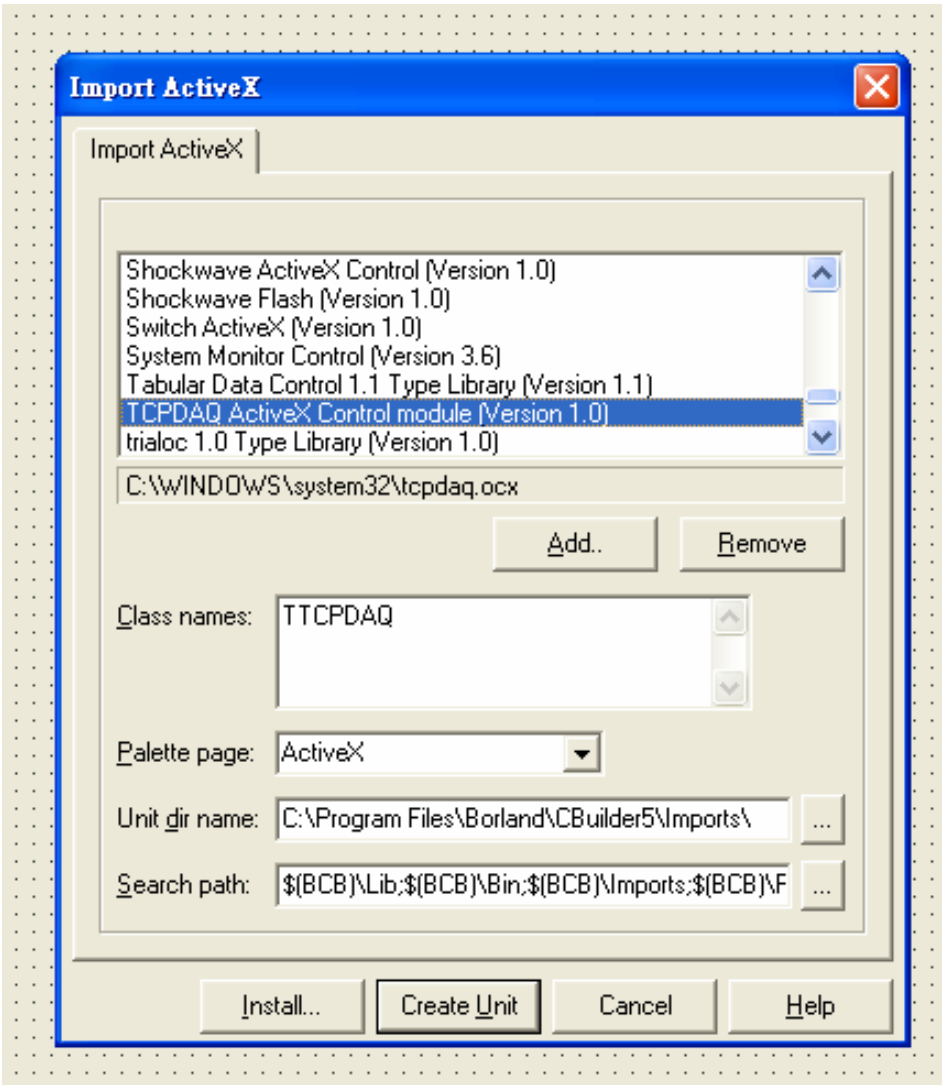


- The **TCPDAQ** control will be showed in Visual C++ Toolbar.
- Place a **TCPDAQ** control from the Controls Toolbar on the dialog-based form.

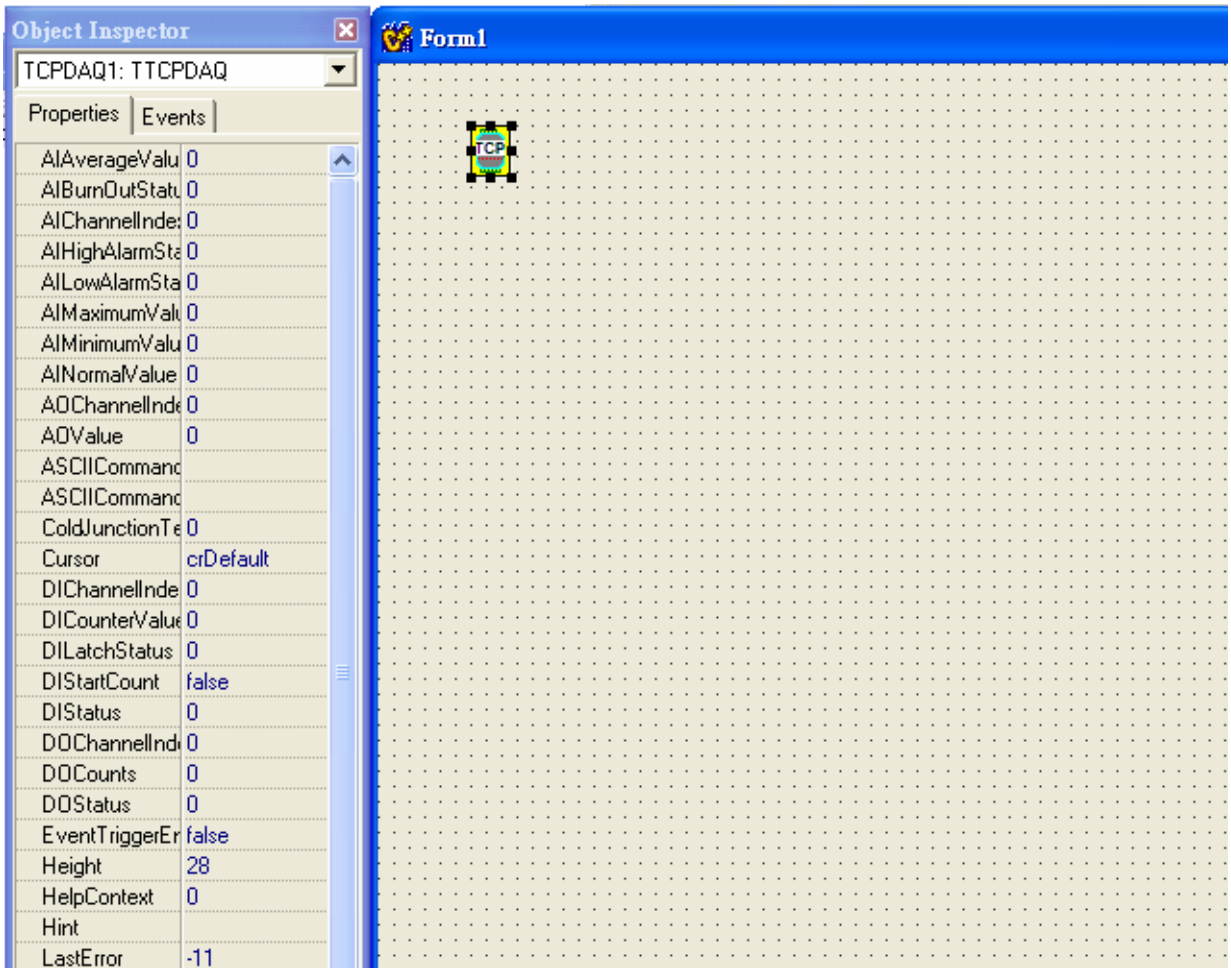


2.4 Building TCPDAQ Applications with Borland C++ Builder

- Start Borland C++ Builder (BCB), BCB will launch as shown below:
- Select **Import ActiveX Control...** from the **Component** menu. The Import ActiveX dialog box loads:
- Select the **TCPDAQ Control** and press the **Install...** button. A dialog box is displayed as follows:



- Enter "TCPDAQ" into the File name field under the **Into new package** tab, and press **OK** button. A Confirm dialog box is displayed. press **Yes** button.
- The **TCPDAQ control** is loaded into the **Component Palette**. You can check it by clicking on **Install Package...** from the **Component** menu. A dialog box is shown as below.



Chapter 3 Properties of TCPDAQ ActiveX Control

Name	Type	Description	Avaliable Model(s)
AIChannelIndex	short	Specifies the analog input channel to perform other AI properties read/write operation.	9015,9017,9019
AINrmalValue	double	Normal voltage of specifies the analog channel	9015,9017,9019
AIAveragevalue	double	Average voltage value of the channels that are in average	9015,9017,9019
AIMaximumValue	double	Maximal voltage of specifies the analog channel	9015,9017,9019
AIMinimumValue	double	Minimal voltage of specifies the analog channel	9015,9017,9019
AILowAlarmStatus	short	Return the low alarm status of specifies the analog channel (1=Alarm occurred, 0=No alarm)	9015,9017,9019
AIHighAlarmStatus	short	Return the high alarm status of specifies the analog channel (1=Alarm occurred, 0=No alarm)	9015,9017,9019
AIBurnOutStatus	short	Return the Burnout status of specifies the analog channel (1=open, 0=normal)	9015 and 9019
AOChannelIndex	short	Specifies the analog output channel to perform other properties read/write operation.	Reserved for Ver 1.0
AOValue	double	Set the analog output voltage	All models
ASCIICommandReceive	string	Return the ASCII response message from module	All models
ASCIICommandSend	string	Send the ASII command message to module	All models
ColdJunctionTemperature	double	Return the cold junction temperature	9019
DIChannelIndex	short	Specifies the digital input channel to perform other DI properties read/write operation.	9050,9051,9052
DIounterValue	long	Return the counting value for the specified DI channel which functions in "Count/Frequency mode"	9050,9051,9052
DILatchStatus	short	Return the latch status for the specified DI channel which functions in "Lo-Hi/Hi-Lo latch mode" (1=Latched, 0=No latched)	9050,9051,9052
DIStartCount	boolean	Start/stop counting for the specified DI channel which functions in "Count/Frequency mode" (True=Start, 0=Stop)	9050,9051,9052
DIStatus	short	Return the status for the specified DI channel which functions in "DI mode" (1=Active, 0=Inactive)	9050,9051,9052
DOChannelIndex	short	Specifies the digital output channel to perform other DO properties read/write operation.	9017,9019,9050,9051,9052
DOCount	long	Set the output count value for the specified	9050,9051,9052

		DO channel which functions in "Pulse output mode"	
DOStatus	short	Return/set the status for the specified DO channel which functions in "D/O mode" (1=Active, 0=Inactive)	9017,9019,9050,9051,9052
EventTriggerEnable	boolean	Enable/disable event trigger mode (True=Enable, False=Disable)	All models
LastError	short	Return the Error code of operation	All models
MoudleIDNo	short	Return the module ID number	All models
ModuleIP	string	Set the remote module IP address	All models
ModuelName	string	Return the module name	All models
TCPTimeOut	long	Return/set the TCP/IP Timeout (ms)	All models
UpdateTimeInterval	long	Return/set data update time interval(ms)	All models

Chapter 4 Methods of TCPDAQ ActiveX Control

Name	Arguments	Returned type	Description
Open	None	None	Open TCPDAQ.OCX to start operation (Must be called before accessing properties at run time)
Close	None	None	Close TCPDAQ.OCX(Must be called before terminating the APP)
ModBusReadCoil	short Startaddress short Counts short coildata[]	None	Read coil data from remote module, and stored into coildata[] buffer
ModBusWriteCoil	shot StartAddress short Counts short coildata[]		Write coil data stored in coildata[] buffer to remote module
ModBusReadReg	short Startaddress short Counts short regdata[]	None	Read holding register data from remote module, and stored into regdata[] buffer
ModBusWriteReg	shot StartAddress short Counts short regdata[]		Write register data stored in regdata[] buffer to remote module

Chapter 5 Events of TCPDAQ ActiveX Control

Name	Arguments	Returned type	Description
OnError	short ErrCode(out) string Errmsg(out)	None	be called when error occurred
EventDataArrival	string Datetime(out) short EventChannel(out) short EventType(out) short EventStatus(out) short EventValue(out)	None	be called when received an event data from the remote module **(*)**

(*): Please see *TCPDAQ_Data_Structure.pdf* file to understand the means of parameters

5.1 Building TCPDAQ ActiveX Applications with Various Development Tools

The demo programs of TCPDAQ AvtiveX control module are included in the provided DISC. The Installed folders include the demo programs for various development tools.

Microsoft Visual C++ version 6.0 (SP5)

Microsoft Visual Basic version 6.0 (SP5)

Borland Delphi version 6.0 (with the Delphi 6 Update Pack fixes for ActiveX installed)

Borland C++ Builder version 6.0