TRAINING GUIDE FOR
OPC SYSTEMS.NET

Simple steps to successful development and deployment.

Step by Step Guide

SOFTWARE DEVELOPMENT TRAINING

OPC Systems.NET Training Guide

Open Automation Software
Evergreen, Colorado, US
www.opcsystems.com

Table of Contents

Chapter

1

Chapter 1 - Overview and Installation

OPC Systems Service

The OPC Systems.NET software uses a central real-time database as its data source for all client applications. This real-time database runs as a Windows Service and can be deployed on multiple computers to create a company or global networked database. The information stored each in service can include a collection of Tags that represent live data, data logging groups, alarm logging groups, alarm notification groups, recipes, reports, security groups, security users, and system options. These configurations in the service can be configured locally or remote using 3 basic methods.

1. OPC Systems Configure Application.

2. CSV Export and Import.

3. Programmatic interface using OPC Systems .NET Component

Once the configurations are created they can then be stored to binary files local to the service. Using Configure-Options the default configuration files to load when the service starts can be defined.

The Tags can have a data sources from OPC Servers, OPC Clients, .NET applications, or databases like SQL Server, Oracle, Access, and mySQL. You can also use the DirectOPC interface from the client components to connect directly to OPC Servers without having to create Tags. All interfaces support .NET communications so remote connections are easily performed over the Internet.

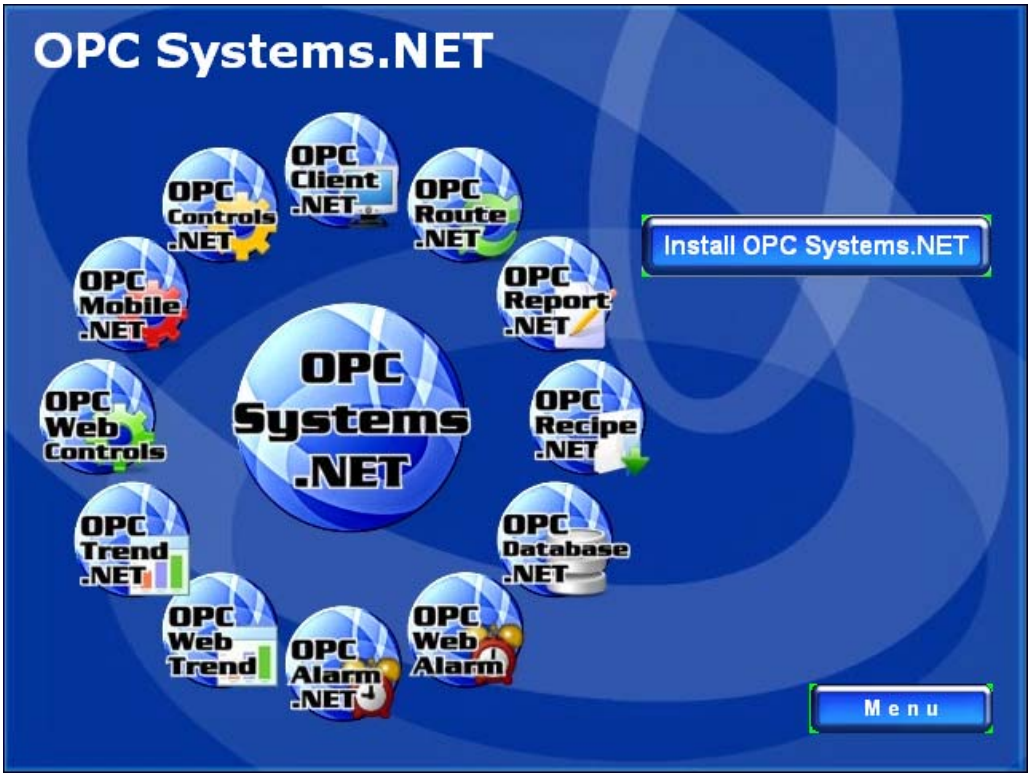
Client Components

Each client component for real-time visualization, data access, trending, alarming, and historical access is configured directly with the component. All components use .NET

communications to talk to the OPC Systems Services. This implementation makes it possible to display and control live data over the Internet without the need for DCOM configuration. All .NET components are 100% managed so they can be used in Smart Client and ASP.NET web applications without introducing any COM or ActiveX legacy technologies. There are trend and alarm ActiveX controls provided for integration into legacy ActiveX containers.

Installation

From the OPC Systems.NET product CD or from the download installation from www.opcsystems.com run the setup for OPC Systems.NET.



Note

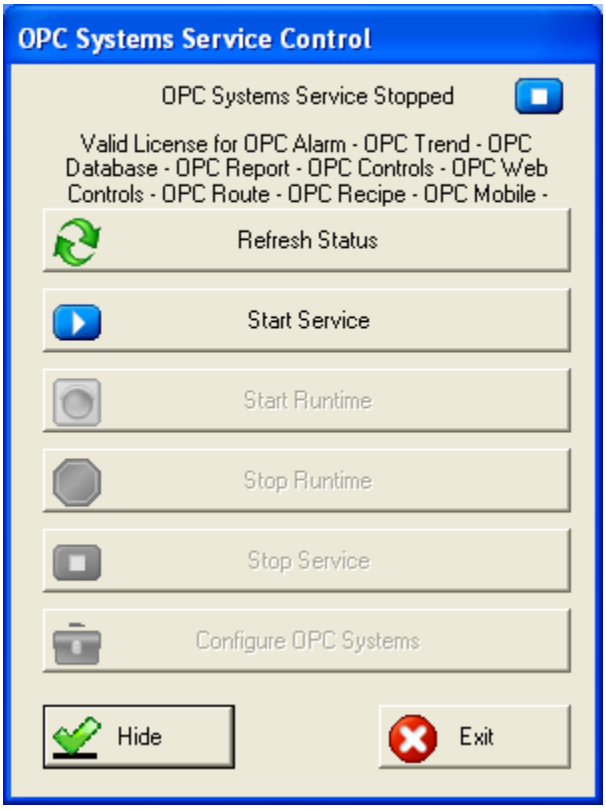
During the installation if prompted that Internet Information Server is not installed you can select Ignore if you do not plan to use OPC Mobile.NET, OPC Web Controls.NET, OPC Web Trend.NET, or OPC Web Alarm.NET.

I

Select Typical installation to install all product features or if you are not sure which components to install. Select Custom if you know the specific product features you wish to install selectively.

Service Start

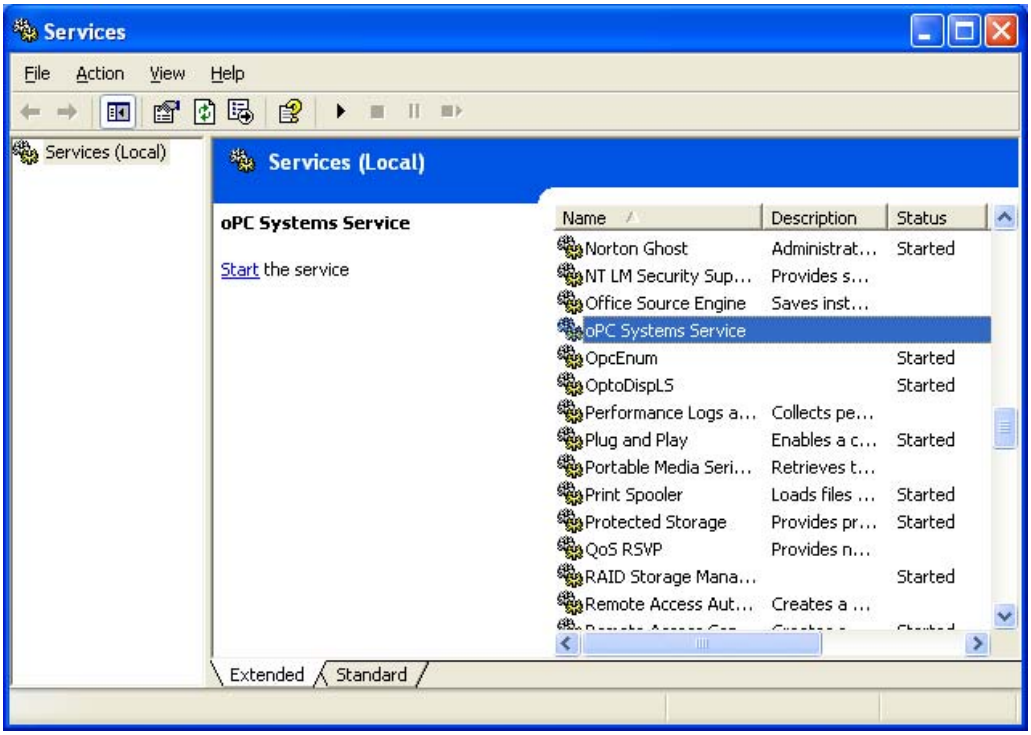
The OPC Systems Service will need to be started after the installation is completed. The OPC System Service Manager will appear after the installation is completed.



Select the Start Service button to start the service. When finished you can Hide the service manager or exit the application. This application can be restored at any time from the OPC Systems.NET program group as the Service icon or from the lower right tray applications as the blue circle.

You can optionally Start Runtime here also or later in the example. All modifications are allowed during Runtime except loading a Tag configuration file. You can add, delete, and modify Tags, Data Logging Groups, Alarm Logging Groups, Alarm Notification Groups, Recipes, Reports, Security Groups, and Security Users during Runtime.

You can also use the Service Control Manager that is built into the operating system to start and stop the OPC Systems Service. This can be brought up by selecting Control Panel-Administrative Tools-Services. From the list of Services select the OPC Systems service.



Service LogOn

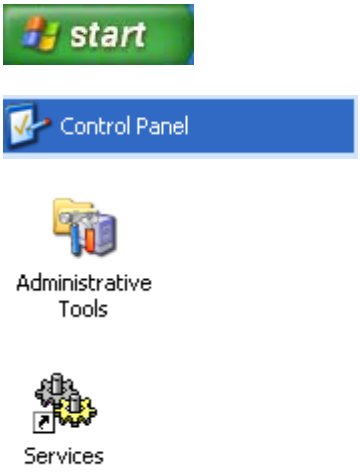
The OPC Systems Service can be configured to Logon as a LocalSystem account or a User account. To correctly launch OPC Servers in the same window space it is recommended to set the OPC Systems Service to Logon with the User Account of the logged on user. Use the Service Control Manager to modify the service's Logon Account. If your OPC Servers run as a Windows Service you can leave the Service LogOn at the default of System.

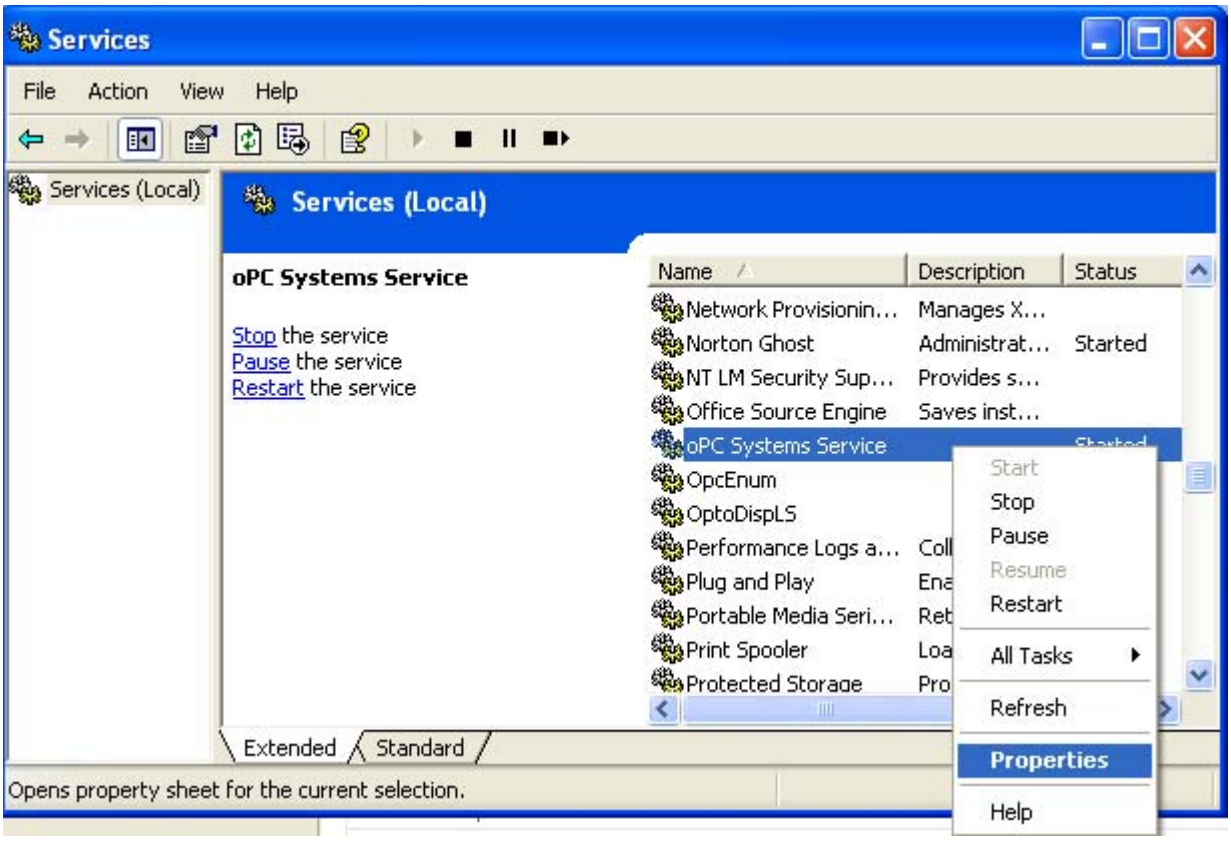
If you are receiving bad data quality from the OPC Server it is most likely due to a security logon restriction in the operating system. To correct this it is recommend to set the OPC Systems Service LogOn to a valid user account.

If the OPC Server is running as a Service you can often leave the Service LogOn to the Local System account, but check the box for "allow service to interact with desktop".

To start the Service Control Manager you must first have Administrative rights to the system.

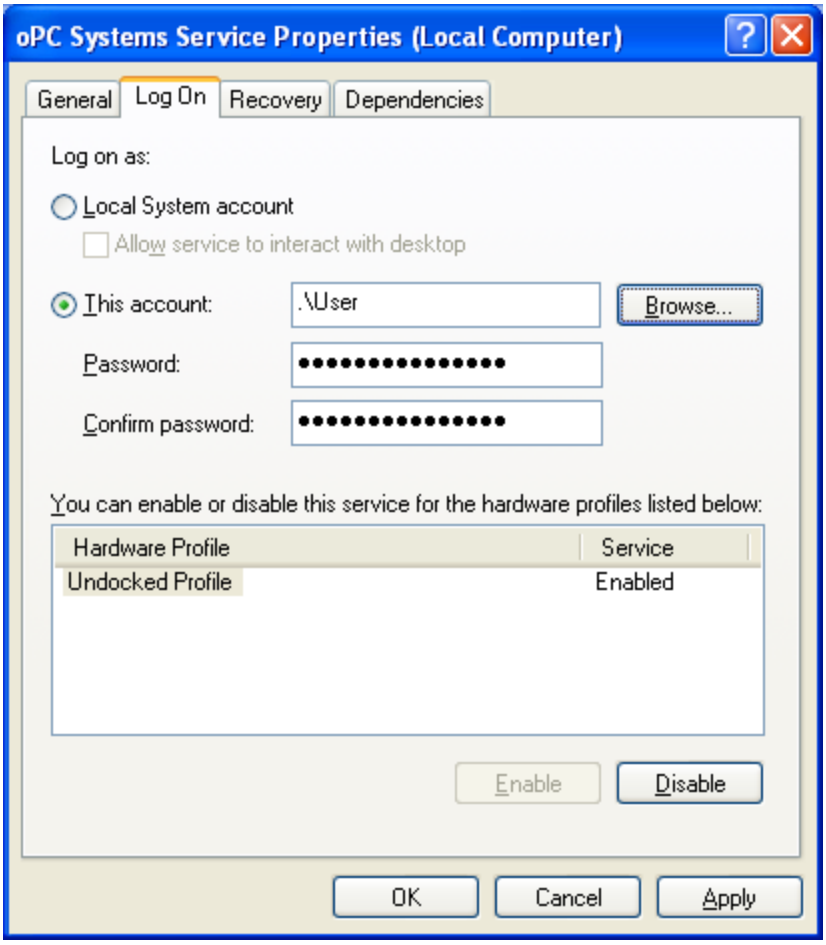
Select Start-Control Panel-Administrative Tools-Services to view all Services installed on the system.



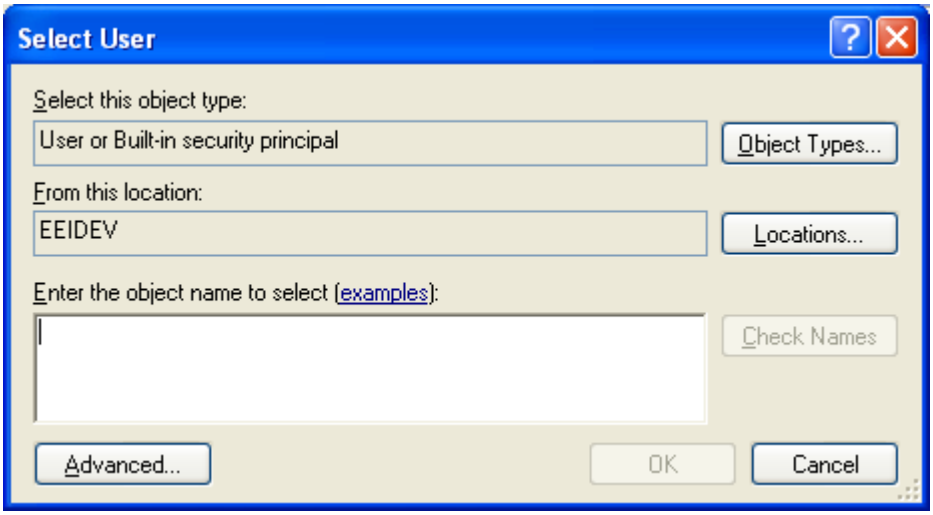


Select OPC Systems and right click to select Properties.

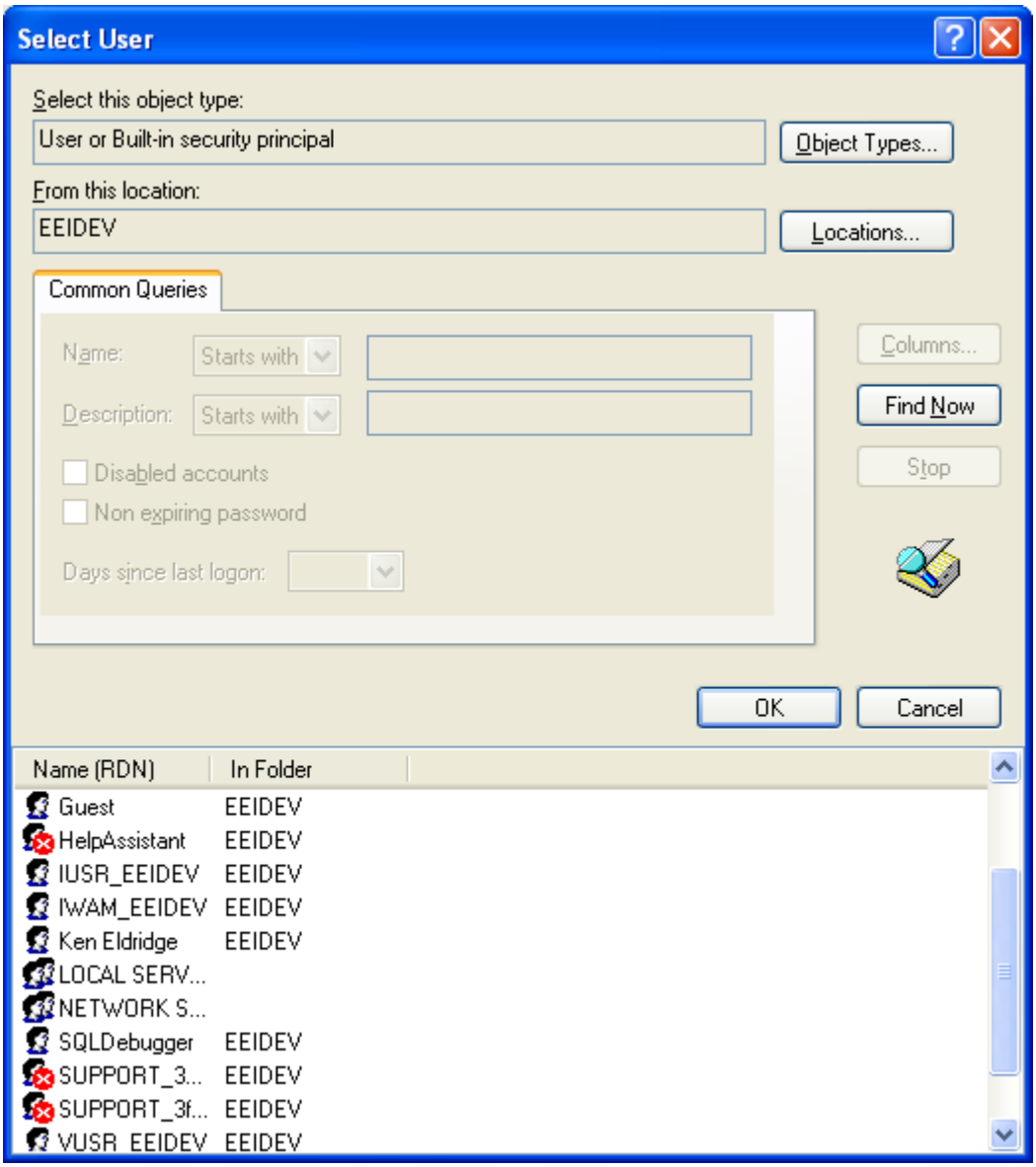
Select the Log On Tab to view the current Logon Account.



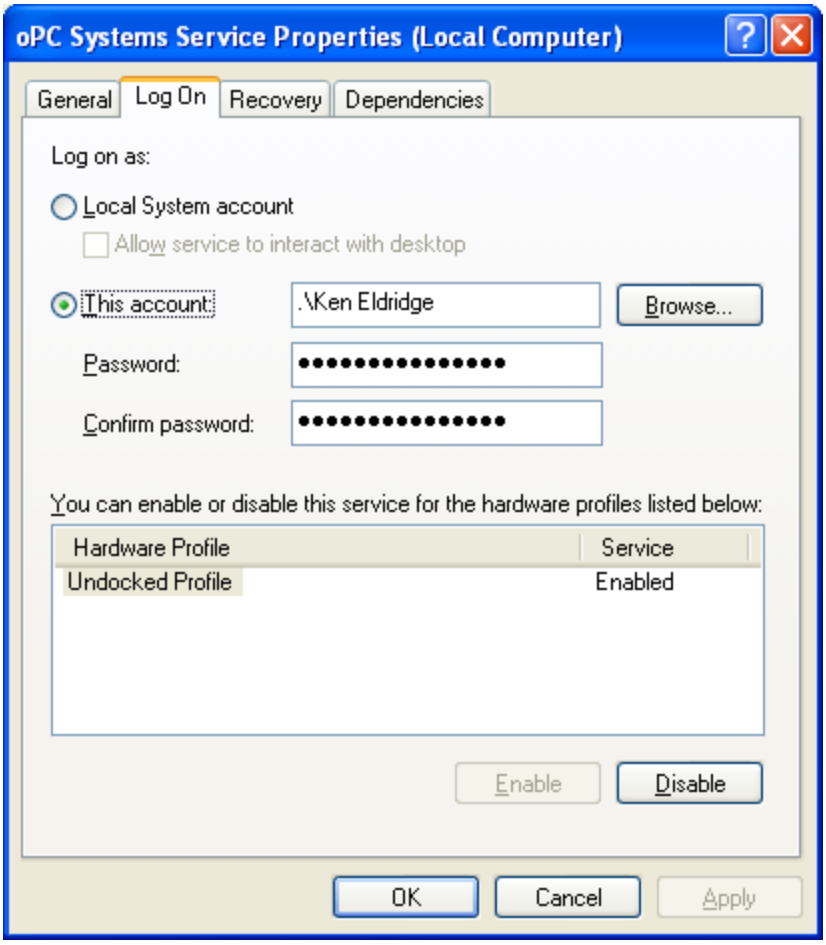
Select This account radio button and use the Browse button to Select User.



Use the Advanced button in the lower left, and then the Find Now button to select the desired Logon User Account.



Select OK twice to return to the Log On Tab dialog.



Enter the Password for the User Account and select OK.

If the Service is already started it must be Stopped and Started for the changes to take effect.

Chapter

2

Chapter 2 - Tags

Configure Tags

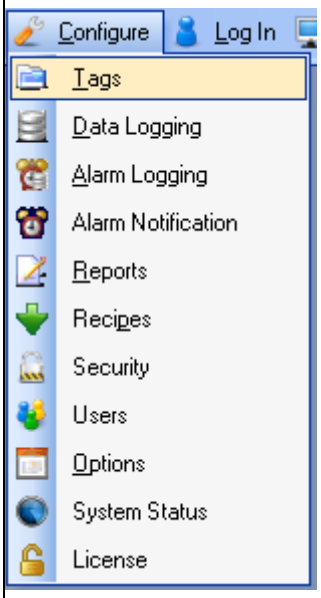
Tags are used to define data source communications, alarm limits, and other realtime signal properties like Time On and Counts for keeping track how long a point has been on and how many times it has transitioned in a given period. Tags are common data sources to all clients. To use data directly from a Visual Studio application without OPC use the default data source of Value as a fixed value that can be changed using the OPC Controls.NET Data component. Tags can also be used to setup Calculations from other local and remote Tags.

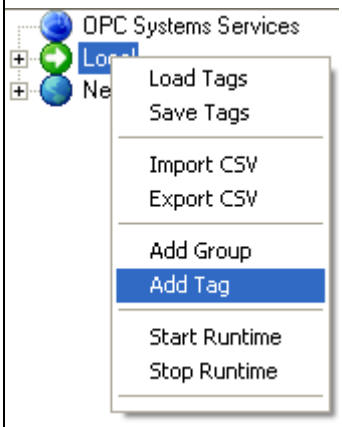
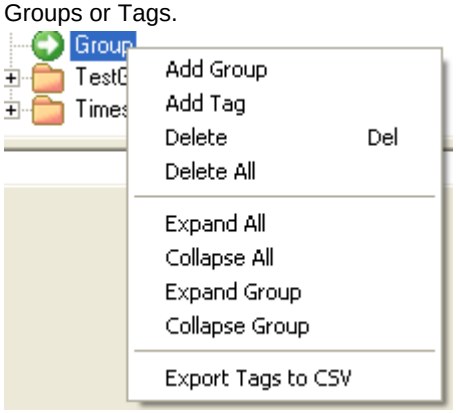
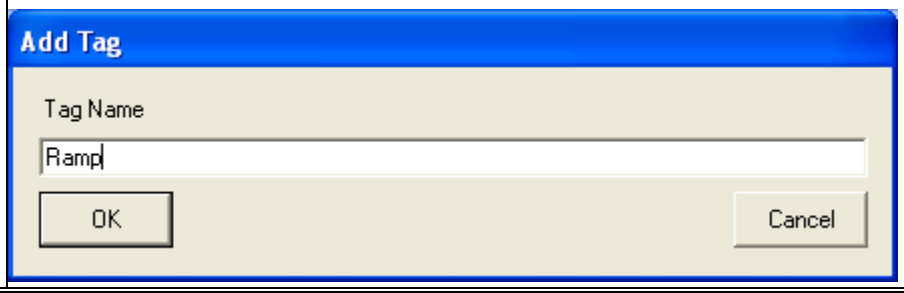
It is possible to connect client components directly to OPC Server Items without configuring Tags, so this step would not be necessary if you plan to use only Direct OPC connections. To learn how Direct OPC connections are defined review each client component step. It is recommended to create at least one OPC Systems.NET Tag to test OPC Server communications and learn what other features a Tag can perform beyond OPC Server connections.

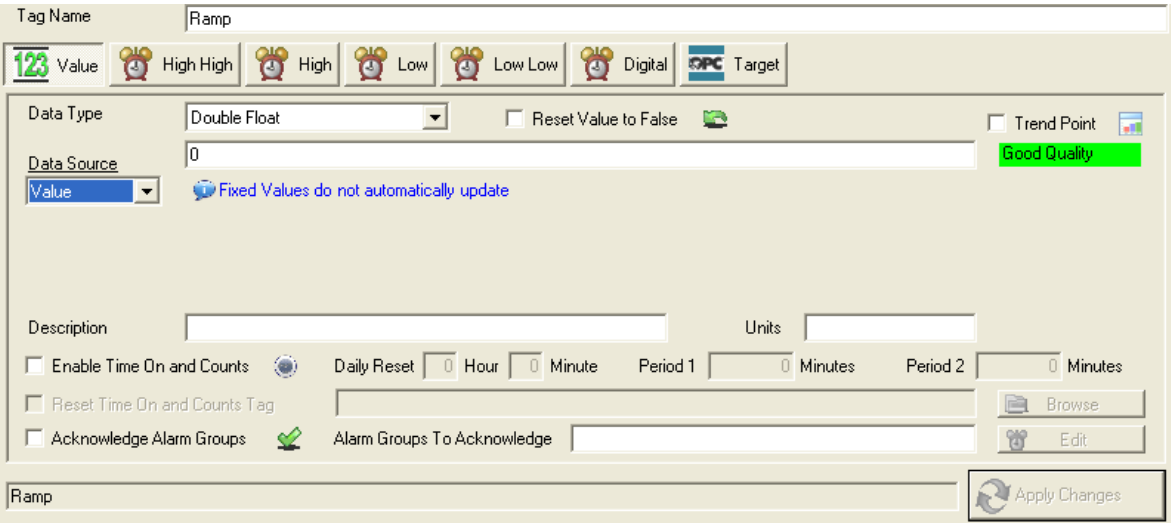
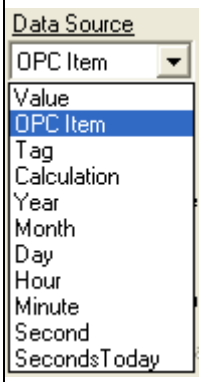
The following section is how to manually add and define Tags using the Configure OPC Systems application. Tags can also be added and modified using the CSV Import and Export selections using the Configure-Tags application to use Excel or other third party Comma Separated Variable editor.

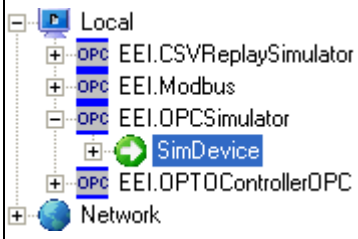
To learn how to programmatically add or modify Tags from your own Visual Studio application refer to the Form FormConfigureCSV in the VB.NET Example on how to add and define multiple tags with one method.

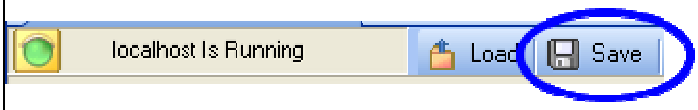
Step	Task
1	Start Configure OPC Systems application. 

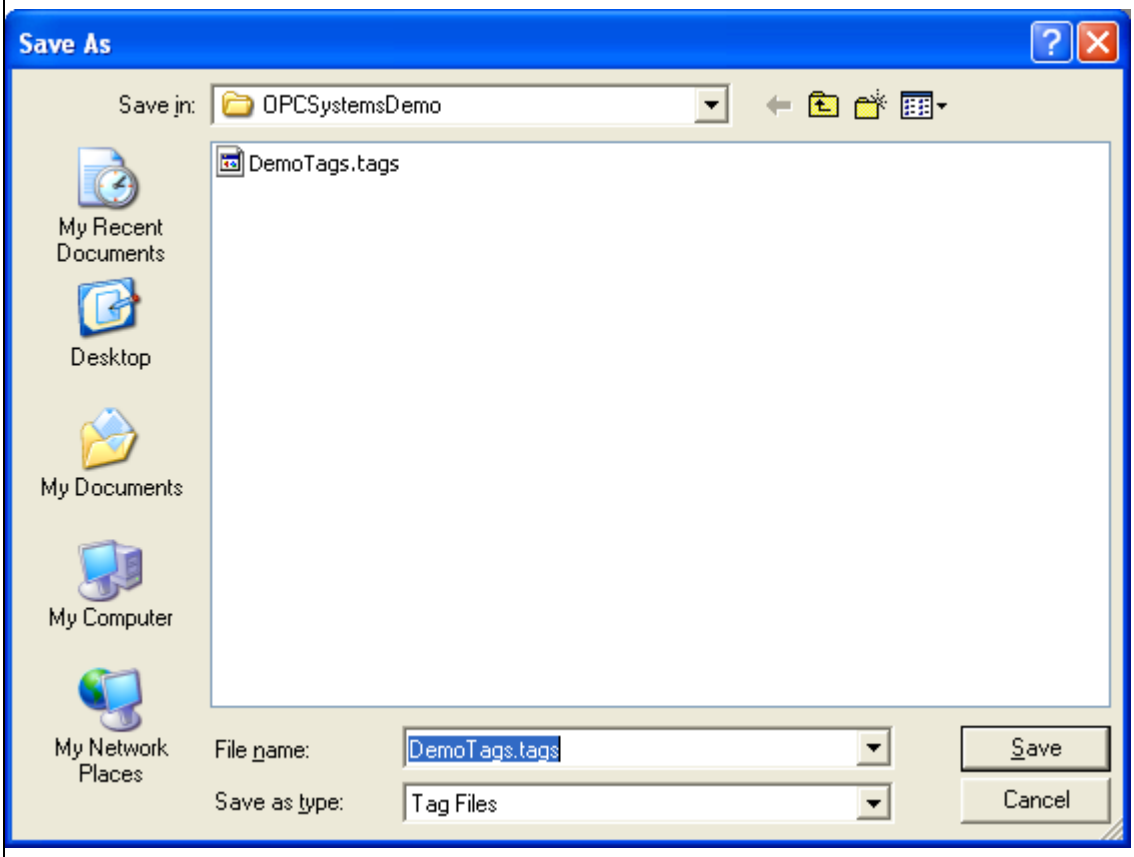
2	Select Configure-Tags. 
3	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left.  **Note** The Configure application can be used to connect to remote systems using the network node name or IP address of the remote node the OPC Systems Service is running on. Simply enter the IP Address or network node name of the remote OPC Systems Service you wish to connect to and click on the Select key. **Note** When selecting a service if you receive a warning dialog that the service cannot be retrieved make sure the OPC Systems Service is started as described in Chapter 1. 

4	Right-Click on the Local OPC Systems Service and select Add Tag.  **Note** You can also add organizational Groups as many levels deep as you prefer and add tags to groups. To do this first add a Group to the root level, then right click on the Group in the right window to add additional Groups or Tags. 
5	Enter the Tag name Ramp in the Add Tag dialog box. 
6	Repeat Steps 4 and 5 with Tag name Sine.
7	Repeat Steps 4 and 5 with Tag name Random.

8	Select Tag Ramp in the right Tag window.  All Tag properties will appear in the lower window. 
9	For the Value Parameter set the Data Source to OPC Item. 
10	Use the OPC Browse button at the right of the OPC Item to browse OPC Servers. 

11	Expand Local to expand EEI.OPCSimulator and select SimDevice. 
12	Select Ramp from the list of OPC Items and select OK to enter the OPC Item EEI.OPCSimulator\SimDevice.Ramp.
13	Enable the Trend Point option in the upper right of the Tag Properties window. 
14	Set the Description field to Ramp.
15	Select the High High Parameter and set the Value field to 80 and enable the High High alarm.
16	Select the High Parameter and set the Value field to 60 and enable the High alarm.
17	Select the Low Parameter and set the Value field to 40 and enable the Low alarm.
18	Select the Low Low Parameter and set the Value field to 20 and enable the Low Low alarm.
19	Select the Apply Changes button in the lower right corner. 

20	Select Tag Random and the Value Parameter.  Repeat steps 8 though 14 and step 19 substituting Ramp for Random as Tag name and OPC Item name. EEI.OPCSimulator\SimDevice.Random.
21	Select Tag Sine and the Value Parameter.  Repeat steps 9 though 19 substituting Ramp for Sine as Tag name and OPC Item name. Use 0.9 for High High Value, 0.8 for High Value, 0.2 for Low Value, and 0.1 for Low Low Value. EEI.OPCSimulator\SimDevice.Sine.
22	Select the Save button on the toolbar at the top. 

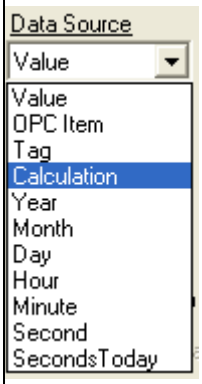
23	Create a directory on the local C:\ drive with the name OPCSystemsDemo. Save the file DemoTags.tags in the directory C:\OPCSystemsDemo. 

Refer to the OPC Systems.NET help file under OPC Systems Configuration- Tags-Parameter Properties for a description of each Tag property.

In order to set the Tag configuration file to automatically load when the OPC Systems Service Starts refer to Configure-Options.

Calculations

You can define Calculation Tags to perform automatic processing of math formulas with any number of local or remote Tags as a data source. You can even use values directly from OPC Servers with the DirectOPC interface, and these OPC Items can also come from the local service or a remote service. If you enable Security on a remote Service to disable all or selected Tags you will need to define the OPC Systems Service User Name and Password under Configure Options, but also recommend to then enable Security to limit access of Read Tags under Configure-Security.

Step	Task
1	Using Configure-Tags create a new Tag with the name Total. Select the local Service or the remote Service with the Ramp, Sine, and Random Tags.
2	Select the new Total Tag and set the Data Source as Calculation. 
3	Use the Edit Calculation button at the right to show the Calculation editor.  Select the Insert Tag button and select Ramp.Value.  Add a + symbol after [Ramp.Value] and insert the Tag Sine.Value. Add a + symbol after [Sine.Value] and insert the Tag Random.Value. The equation should now represent a total of all 3 Tags as the following. [Ramp.Value]+[Sine.Value]+[Random.Value]

	You can select OK and Apply Changes to then see the Total Tag value update to the total of all 3 Tags.

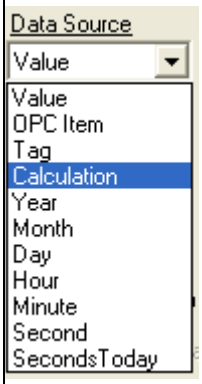
Refer to the Configure-Tags-Calculations section in the OPC Systems.NET Help file for descriptions of each Function for the Calculation engine.

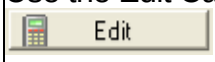
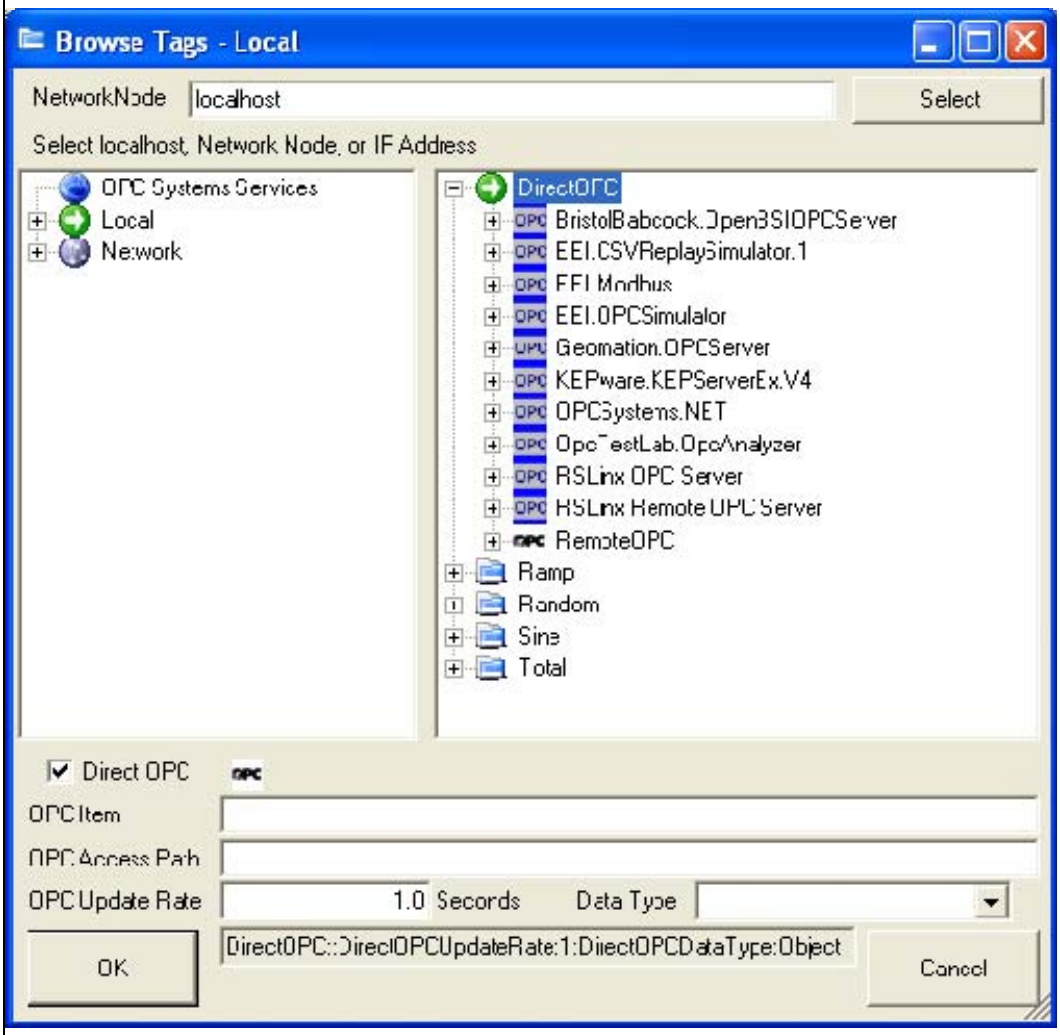
Note

If any tag data source in a calculation is bad the resultant data quality for the calculation tag will also be bad.

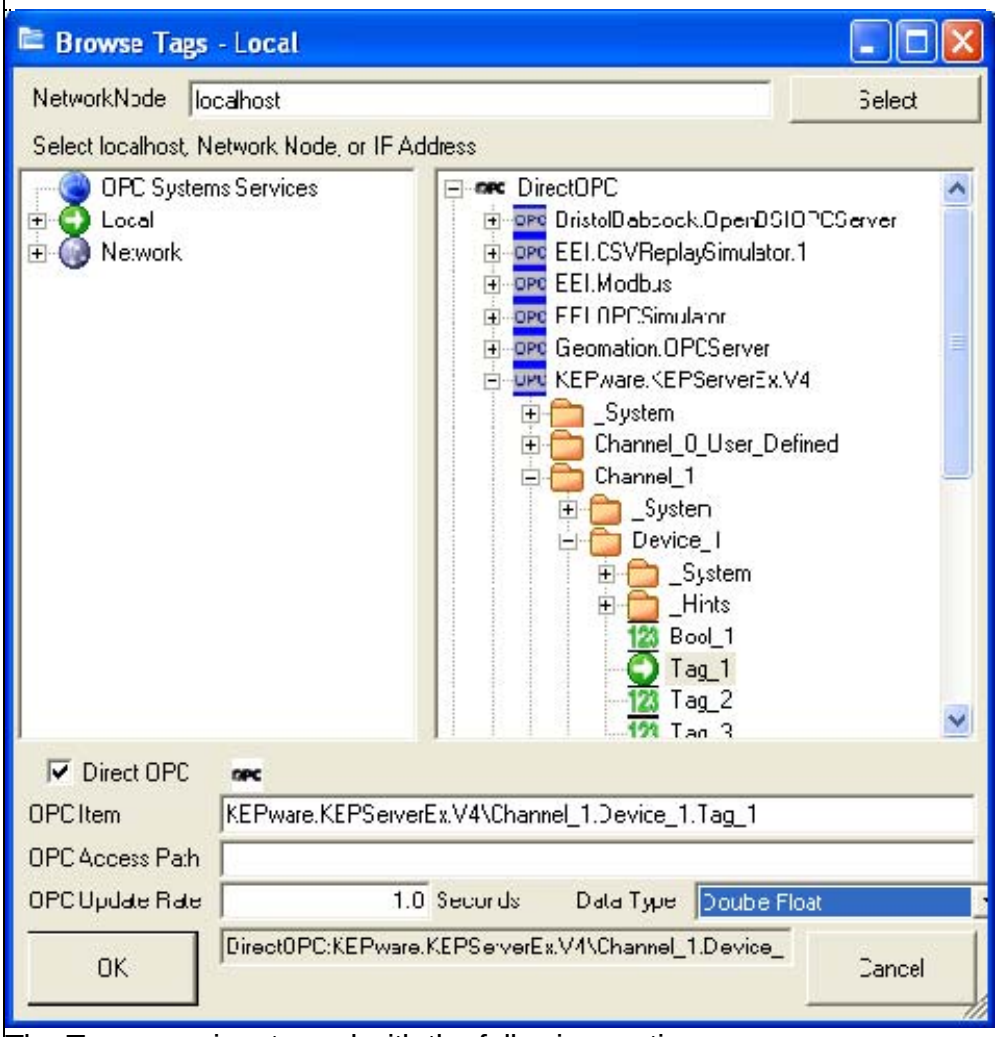
DirectOPC Interface

All client components that use OPC Systems.NET Tags as a data source can also use values directly from OPC Servers with the DirectOPC interface.

Step	Task
1	Using Configure-Tags create a new Tag with the name CalcDirectOPC. Select the local Service or remote Service that has the OPC Server you want to connect to.
2	Select the new CalcDirectOPC Tag and set the Data Source as Calculation. 

3	Use the Edit Calculation button at the right to show the Calculation editor.  Select the Insert Tag button and select the plus icon to the left of DirectOPC.  A list of OPC Servers installed on the system you are currently connected to.

Browse one of your OPC Servers for an OPC Item and set the OPC Update Rate and desired Data Type.



The Tag name is returned with the following portions.

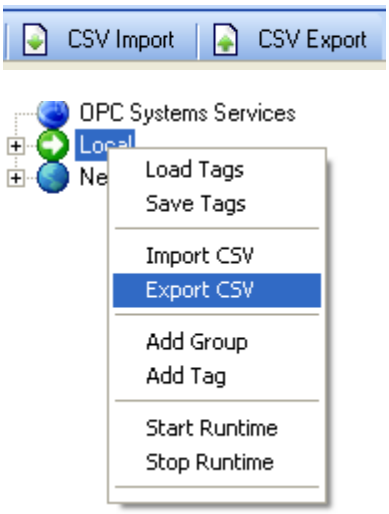
[DirectOPC:KEPware.KEPServerEx.V4\Channel_1.Device_1.Tag_1:DirectOPCUpdateRate:1:DirectOPCDataType:Double Float:DirectOPCAccessPath:]

OPC Server: KEPware.KEPServerEx.V4
OPC Item: Channel_1.Device_1.Tag_1
OPC Update Rate: 1
Data Type: Double Float

All client components can use this alternative syntax for connecting directly to OPC Servers. This provides a direct pass through for OPC Items on local and remote systems.

Tag CSV Export and Import

Tags can be exported to CSV files to be modified using Microsoft Excel. Simply right click on the Local Service and select Export CSV or select the CSV Export button on the toolbar at the top.



Open the CSV file with Microsoft Excel to review or modify the Tag parameters. You can create new Tags by adding additional rows to the file. You can sort and move columns in any order you like. Only the Tag column is required, all other columns are optional. If a property column is deleted new Tags will be created with the default value of the property.

When you have finished the modifications close Microsoft Excel and use Import CSV to load all modifications back to the tag configuration and then save the changes to a tag configuration file as shown in Step 22 of Configure Tags.

Note

You cannot have Excel open with the CSV file during the import or export as Excel will lock the file for exclusive use. First close the file in Excel, and then proceed with the import or export.

Tags Programmatic Interface

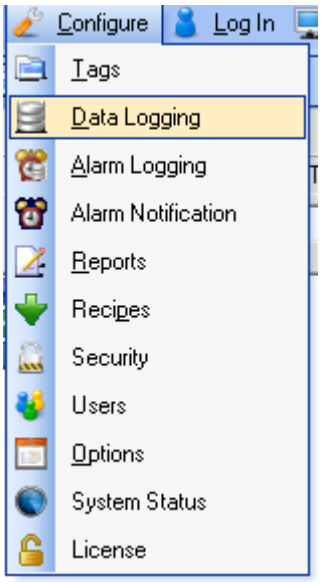
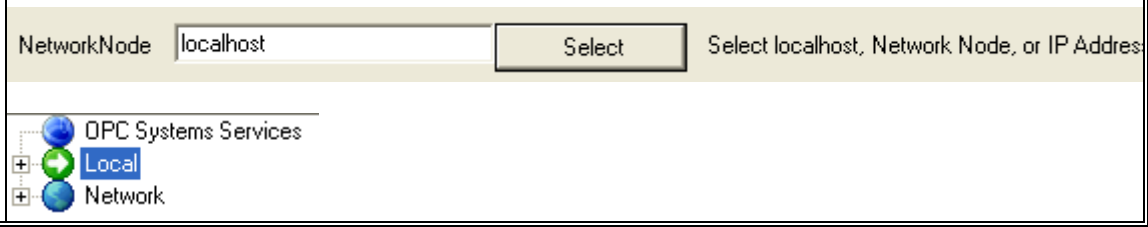
Use the OPC Systems component in your Visual Studio application to programmatically modify tags. Refer to the Form FormConfigureCSV as the best method to add or modify multiple tags Refer to the OPC Systems Component help file for all of the method syntax.

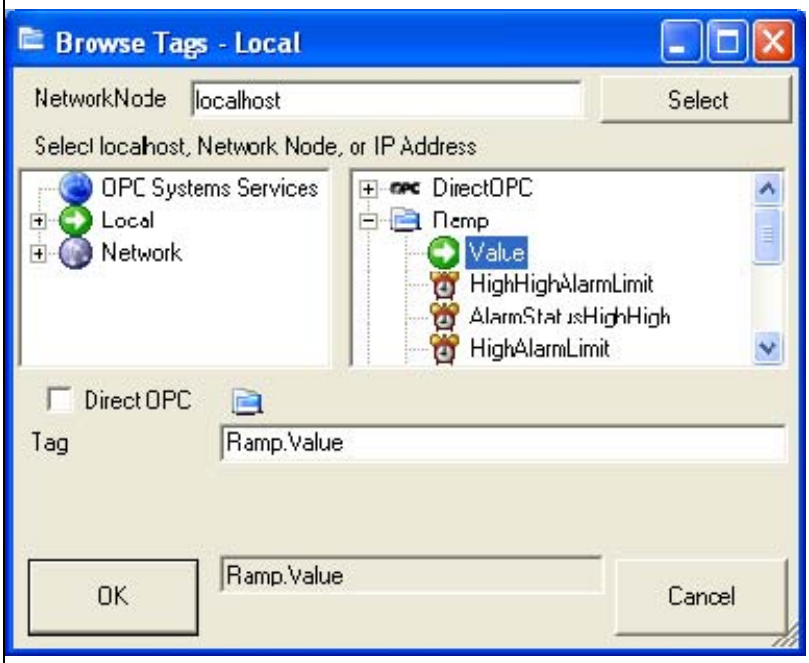
Chapter

3

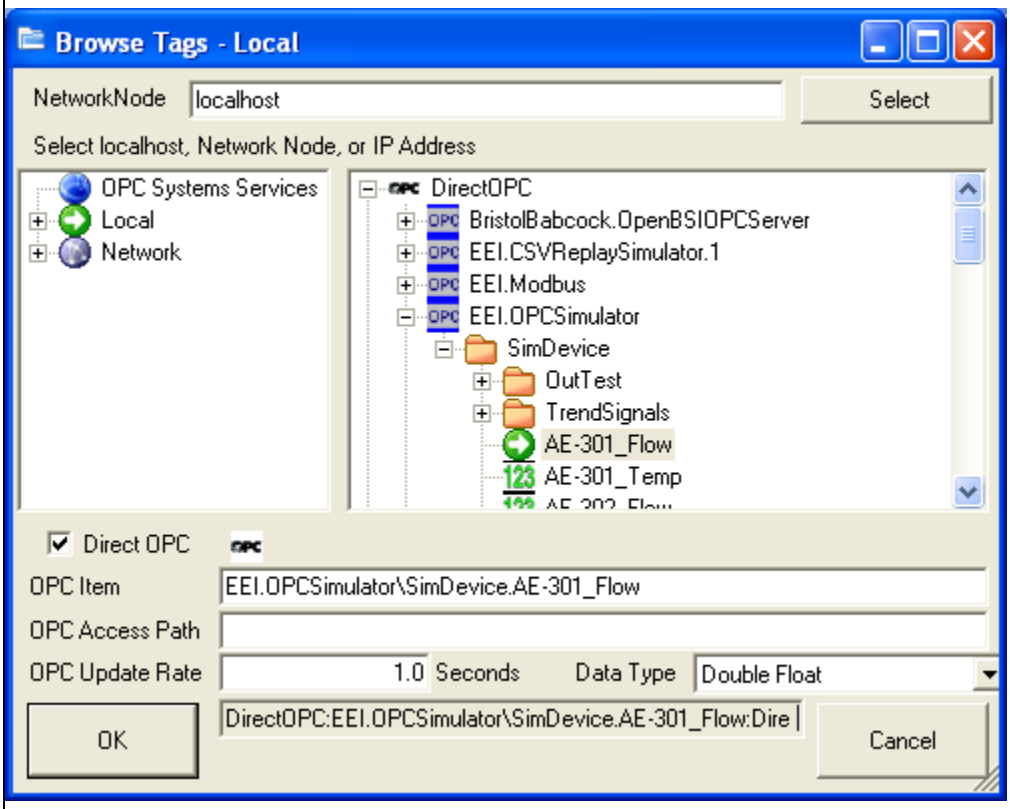
Chapter 3 - Data Logging

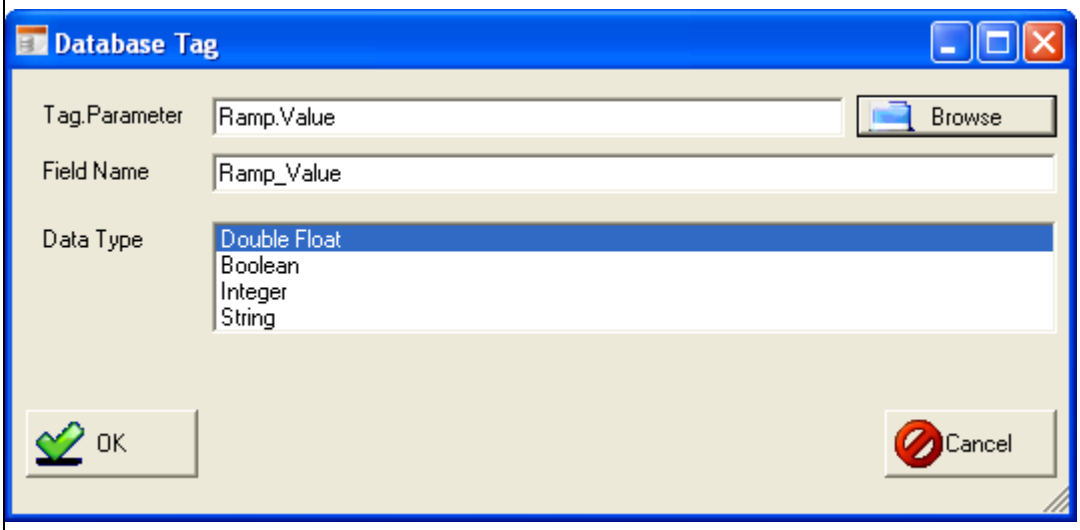
Configure Data Logging

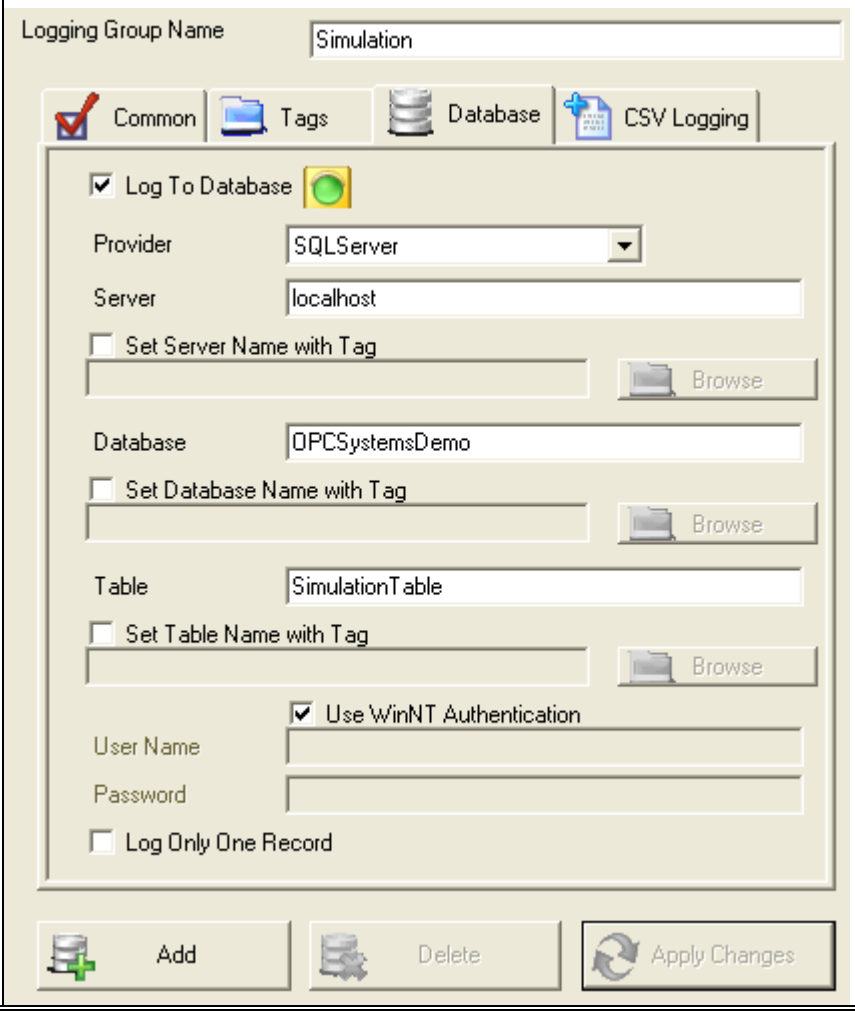
Step	Task
1	Start Configure OPC Systems application if it is not already running. 
2	Select Configure-Data Logging. 
3	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left. 

4	Enter the Logging Group Name of Simulation in the field in the upper right. 
5	Check Logging Active in the Common Properties Tab.  We use the default logging type of Continuous at a 1 second rate.
6	Select the Tags Tab. 
7	Select the Add Field button. 
8	Select the Tag Parameter browse button to select the Tag Parameter Ramp.Value.  

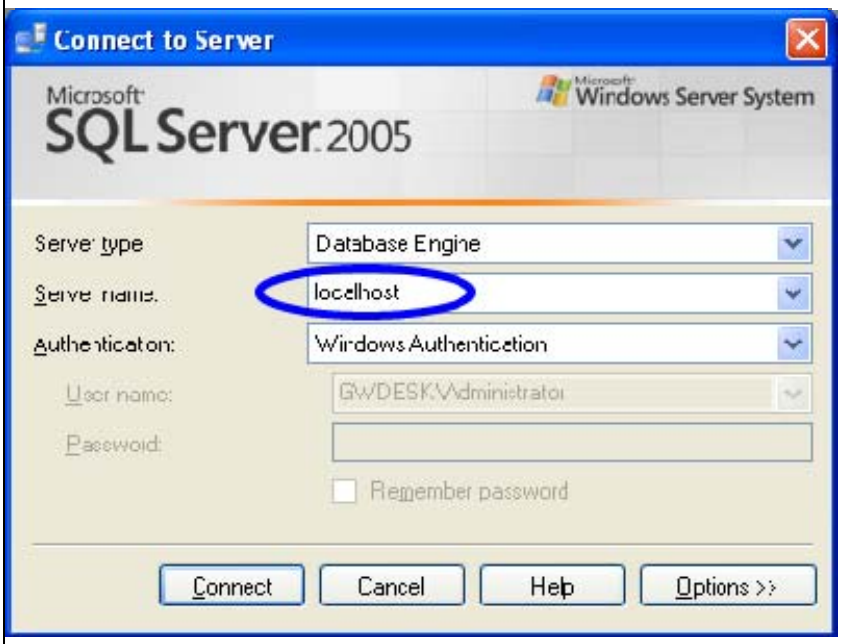
You could also optionally use the DirectOPC selection to connect directly to OPC Items on OPC Servers.





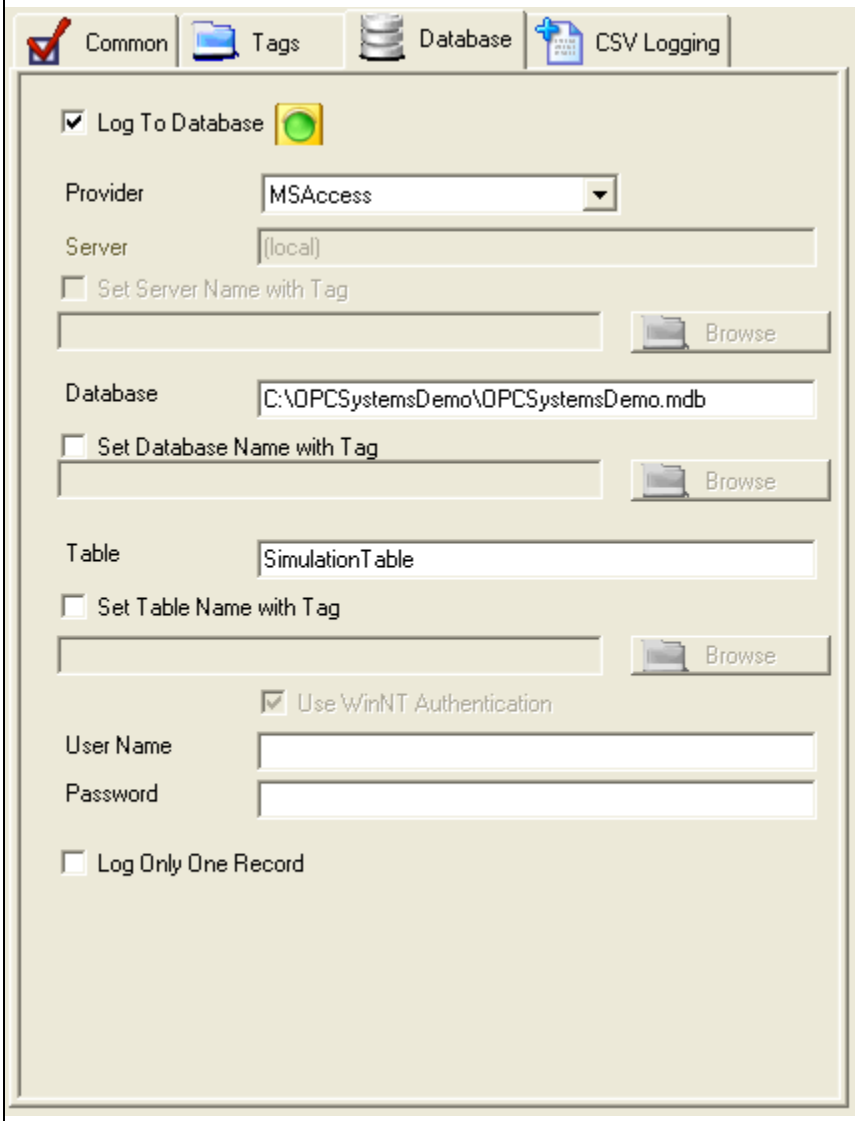
9	Repeat steps 7 and 8 for the Tag Parameter Random.Value. Repeat steps 7 and 8 for the Tag Parameter Sine.Value. Field Name / Data Type / Tag.Parameter Ramp_Value / Double / Ramp.Value Random_Value / Double / Random.Value Sine_Value / Double / Sine.Value
10	Select the Database Tab.
11	Use the following configuration for SQL Server. The Database and Table will automatically be created for you. 

The Server name may need to be adjusted from localhost to the proper Server name of the SQL Server engine you wish to log to. This Server name can be found when first bringing up the Connect dialog of the SQL Server Management Studio. Also the login method can be with Windows Authentication or SQL Server mode. Contact your database administrator if you are unsure of what login type to use.



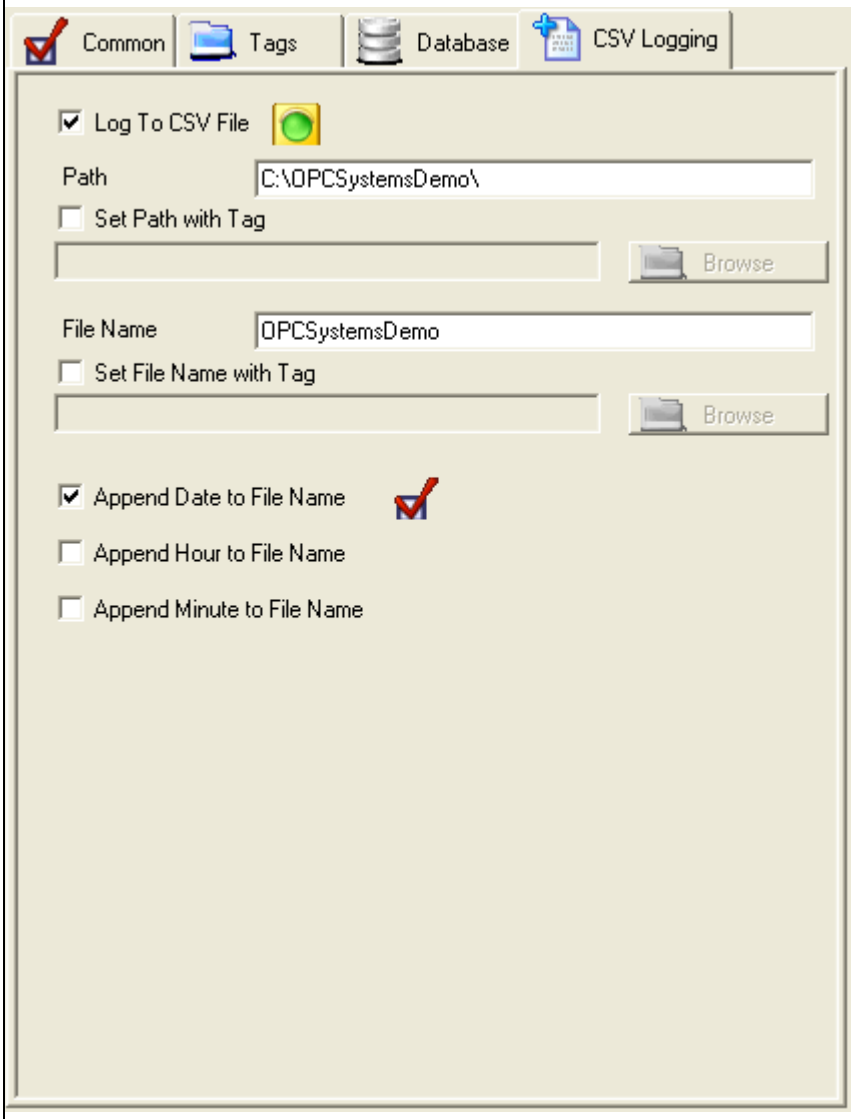
You can download a free version of SQL Server Express from www.microsoft.com.

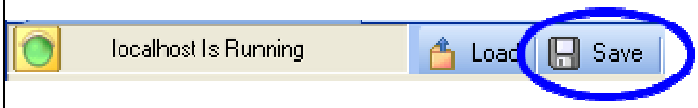
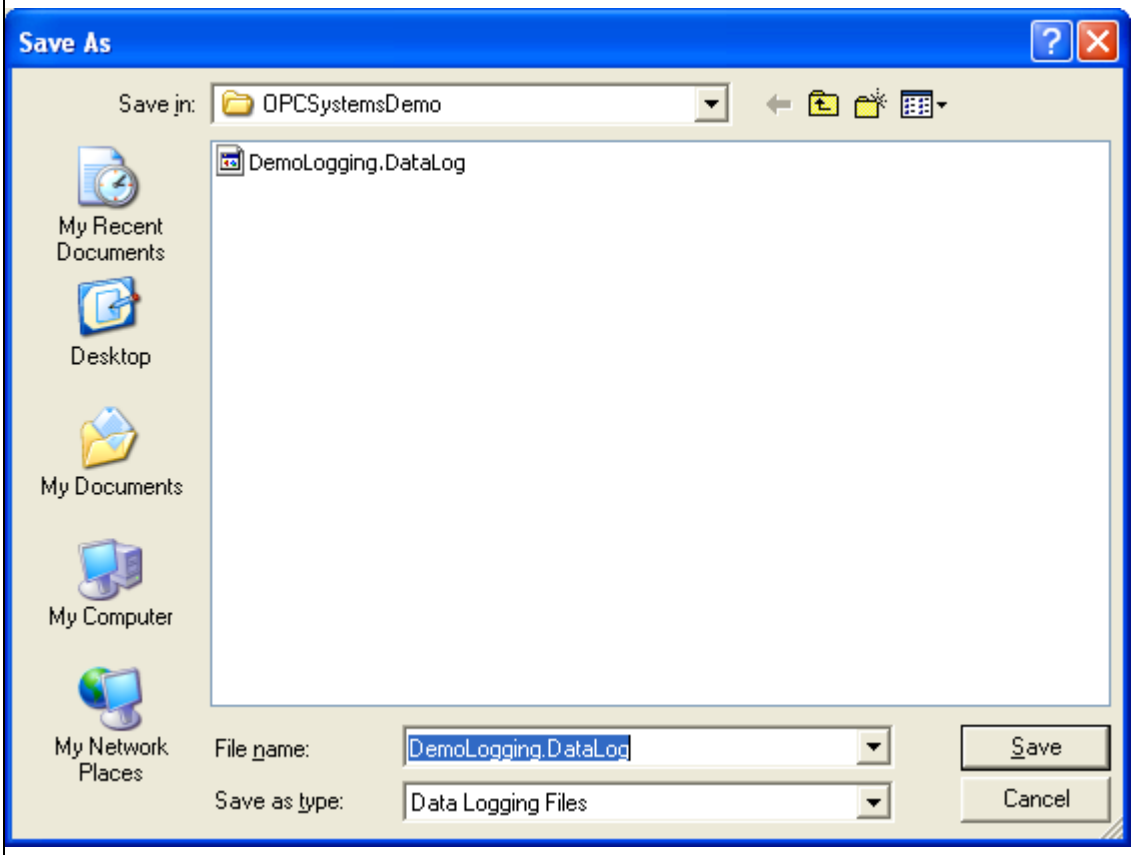
If you prefer to use Microsoft Access instead of SQL Server use the following configuration for Microsoft Access. The Database and Table will be automatically created for you. Make sure to create the directory OPCSystemsDemo on your C drive or specify a different Database path that will exist.



Use the following configuration for CSV text logging.

Make sure to create the directory OPCSystemsDemo on your C drive or specify a different Database path that will exist.



12	Select the Add button to add the Data Logging group. 
13	Select the Save button on the toolbar at the top. 
14	Save the file DemoLogging.DataLog in the directory C:\OPCSystemsDemo. 

Refer to the OPC Systems.NET help file under OPC Systems Configuration- Data Logging for a description of each property.

Data Logging High Speed Data from Visual Studio Applications

OPC Systems.NET will support data logging from Visual Studio application with resolution to 100 nanoseconds.

In order to log values directly from a Visual Studio application you will need to implement the OPC Controls.NET Data Component as demonstrated in the VB.NET Example under FormWriteValues. You can log values just as if the values were from an OPC Server, just setup Tags with a data source of Value, the desired data type, and then from your Visual Studio application use the WriteTags method of the OPC Controls.NET Data component. If you also include TimeStamps array in the WriteTags method the service will use the TimeStamps you write instead of the CPU clock time when writing values to the database.

You can also setup data logging groups to be triggered based on event and assign a Boolean Tag as the Trigger Tag. From your Visual Studio application your code in simple form would look like the following.

```
Dim sourceTimeStamps(2) As Date
Dim timeNow As Date = Now
sourceTimeStamps(0) = timeNow
timeNow = timeNow.AddTicks(1)
sourceTimeStamps(1) = timeNow
timeNow = timeNow.AddTicks(1)
sourceTimeStamps(2) = timeNow
Dim TagsToWrite(14) As String
Dim ValuesToWrite(14) As Object
Dim TimeStampToWrite(14) As Date
TagsToWrite(0) = "Value01.Value"
ValuesToWrite(0) = 1
TimeStampToWrite(0) = sourceTimeStamps(0)
TagsToWrite(1) = "Value02.Value"
ValuesToWrite(1) = 1
TimeStampToWrite(1) = sourceTimeStamps(0)
TagsToWrite(2) = "Value03.Value"
ValuesToWrite(2) = 1
TimeStampToWrite(2) = sourceTimeStamps(0)
TagsToWrite(3) = "Trigger.Value"
ValuesToWrite(3) = True
TimeStampToWrite(3) = sourceTimeStamps(0)
TagsToWrite(4) = "Trigger.Value"
ValuesToWrite(4) = False
TimeStampToWrite(4) = sourceTimeStamps(0)

TagsToWrite(5) = "Value01.Value"
ValuesToWrite(5) = 2
TimeStampToWrite(5) = sourceTimeStamps(1)
TagsToWrite(6) = "Value02.Value"
ValuesToWrite(6) = 2
```

```
        TimeStampToWrite(6) = sourceTimeStamps(1)
        TagsToWrite(7) = "Value03.Value"
        ValuesToWrite(7) = 2
        TimeStampToWrite(7) = sourceTimeStamps(1)
        TagsToWrite(8) = "Trigger.Value"
        ValuesToWrite(8) = True
        TimeStampToWrite(8) = sourceTimeStamps(1)
        TagsToWrite(9) = "Trigger.Value"
        ValuesToWrite(9) = False
        TimeStampToWrite(9) = sourceTimeStamps(1)

        TagsToWrite(10) = "Value01.Value"
        ValuesToWrite(10) = 3
        TimeStampToWrite(10) = sourceTimeStamps(2)
        TagsToWrite(11) = "Value02.Value"
        ValuesToWrite(11) = 3
        TimeStampToWrite(11) = sourceTimeStamps(2)
        TagsToWrite(12) = "Value03.Value"
        ValuesToWrite(12) = 3
        TimeStampToWrite(12) = sourceTimeStamps(2)
        TagsToWrite(13) = "Trigger.Value"
        ValuesToWrite(13) = True
        TimeStampToWrite(13) = sourceTimeStamps(2)
        TagsToWrite(14) = "Trigger.Value"
        ValuesToWrite(14) = False
        TimeStampToWrite(14) = sourceTimeStamps(2)

        OpcControlsData1.WriteTags(TagsToWrite, ValuesToWrite,
TimeStampToWrite)
```

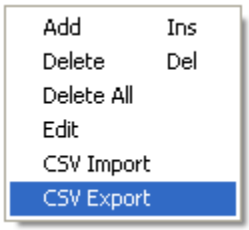
The result in the database for this one WriteTags method would be 3 records.

DateAndTime	ms	us	ns	Value01	Value02	Value03
3/10/2009 11:04:50 AM	984	375	0	1	1	1
3/10/2009 11:04:50 AM	984	375	100	2	2	2
3/10/2009 11:04:50 AM	984	375	200	3	3	3

Data Logging CSV Export and Import

All logging groups can be exported to a CSV file along with all database field names to individual sub files by right clicking on the Local service and select Export CSV.

You can also export just the individual fields for a logging group by right clicking in the Tags field list.



Note

You cannot have Excel open with the CSV file during the import as Excel will lock the file for exclusive use. First close the file in Excel, and then proceed with the import.

Data Logging Programmatic Interface

Use the OPC Systems component in your Visual Studio application to programmatic modify data logging groups. Refer to the Configure Data Logging Form and the Configure CSV Form in the VB.NET example for an example and how to add and modify data logging groups. Refer to the OPC Systems Component help file for all of the method syntax.

Chapter

4

Chapter 4 - Trending

Trending Components

There are 3 trending components that can be used to trend real-time data from the tag configuration.

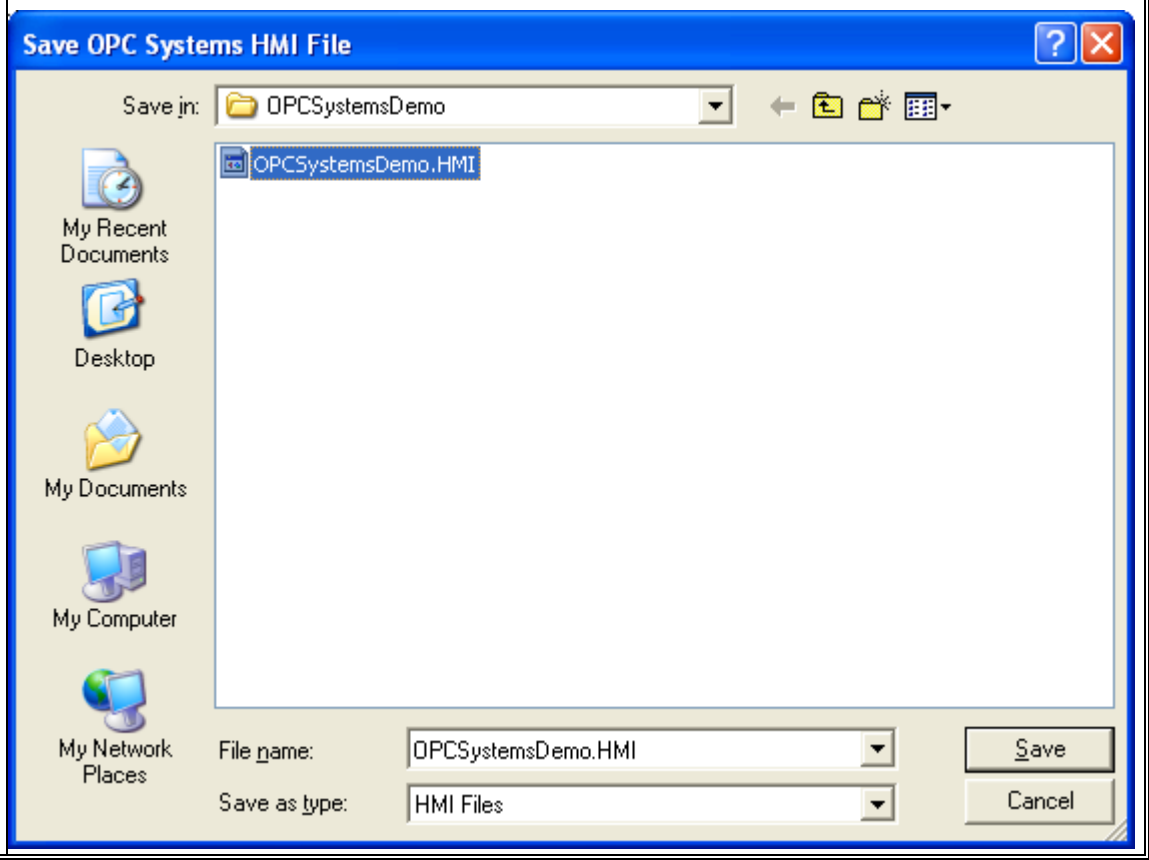
1. OPC Trend .NET 100% managed component for WinForm applications.
2. OPC Web Trend.NET 100% managed ASP.NET component for Web applications.
3. OPC Trend ActiveX control for legacy applications.

All 3 types of components can be run locally or remotely for live data from any service containing an OPC Trend.NET license or OPC Web Trend.NET license.

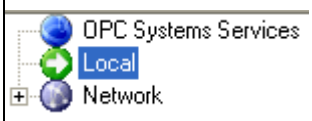
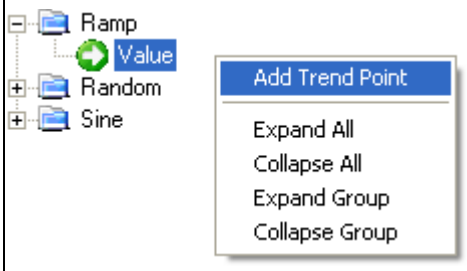
All 3 types of components also support historical replay from any service containing an OPC Database.NET license with logging groups defined as in Chapter 3.

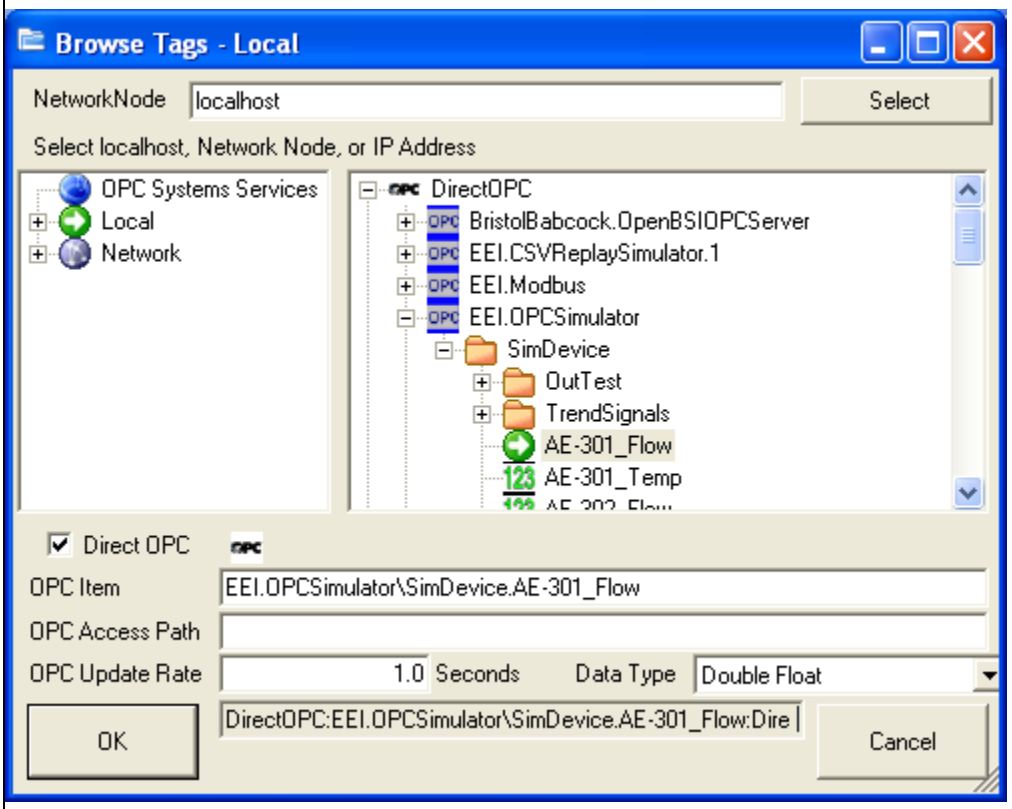
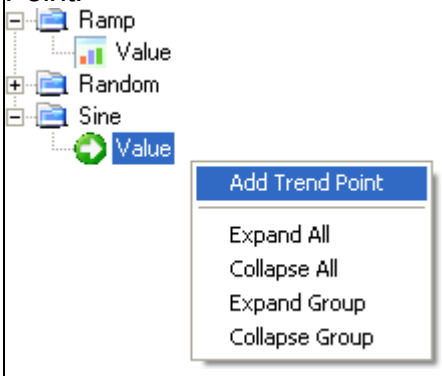
OPC Systems HMI

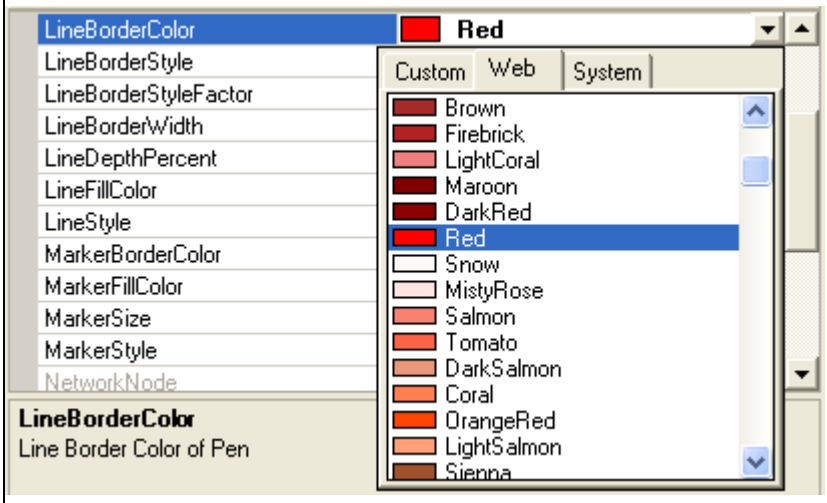
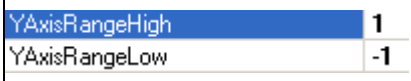
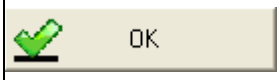
The OPC Systems HMI container allows you to use the OPC Trend .NET and OPC Alarm.NET components directly without using Visual Studio.

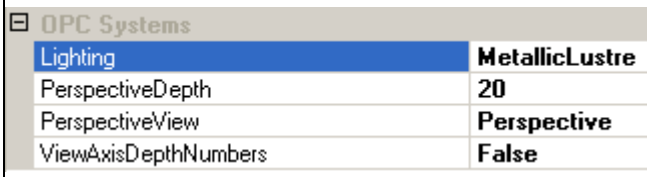
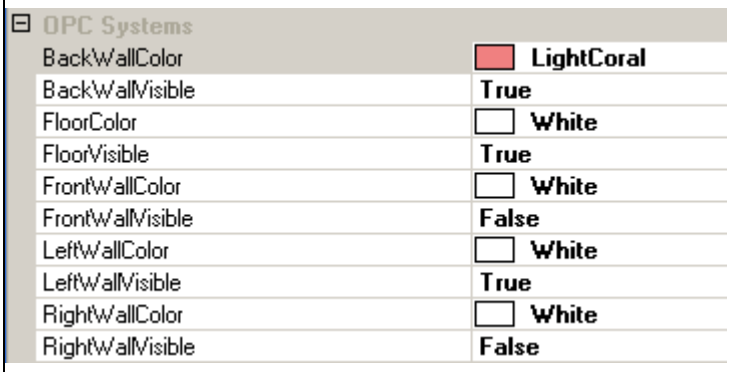
Step	Task
1	Start OPC Systems HMI application.
2	Select File-New and save the file OPCSystemsDemo in the C:\OPCSystemsDemo\ directory. 

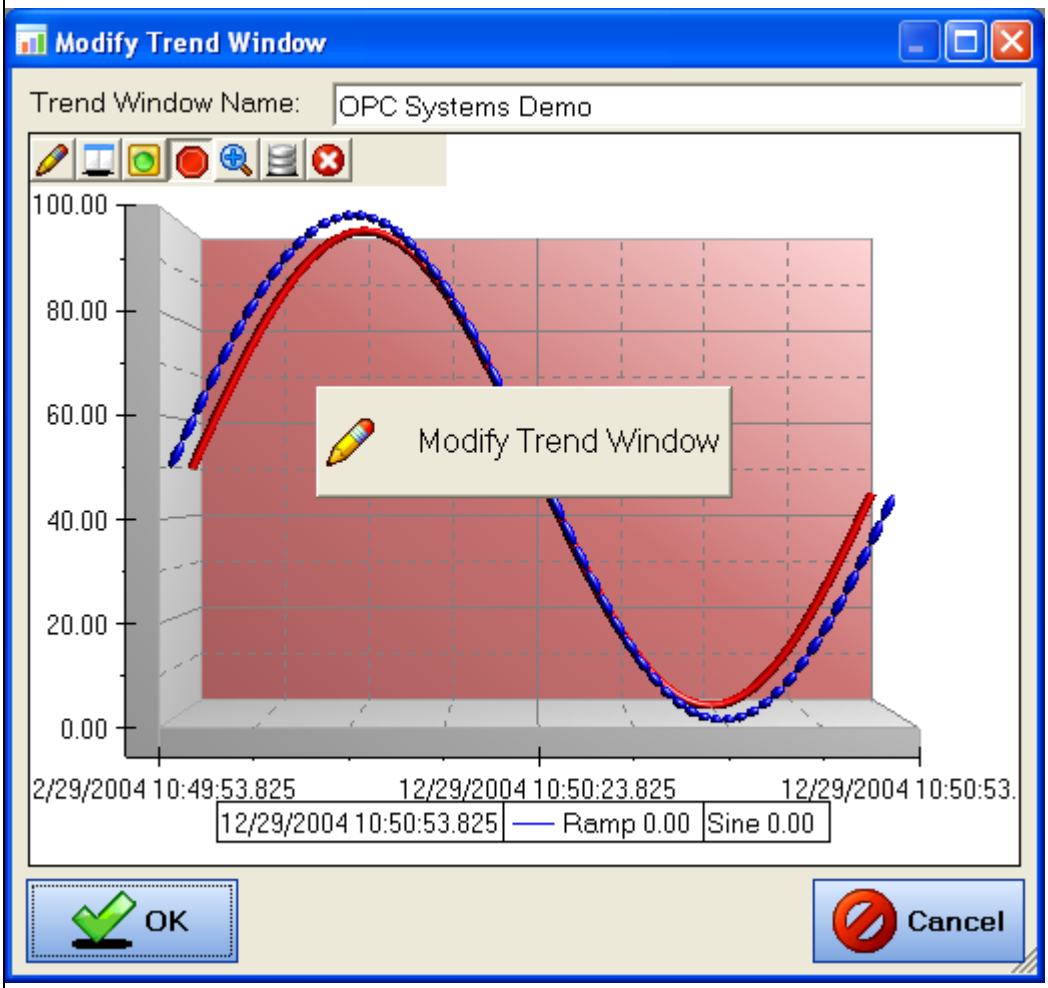
Configure Trend

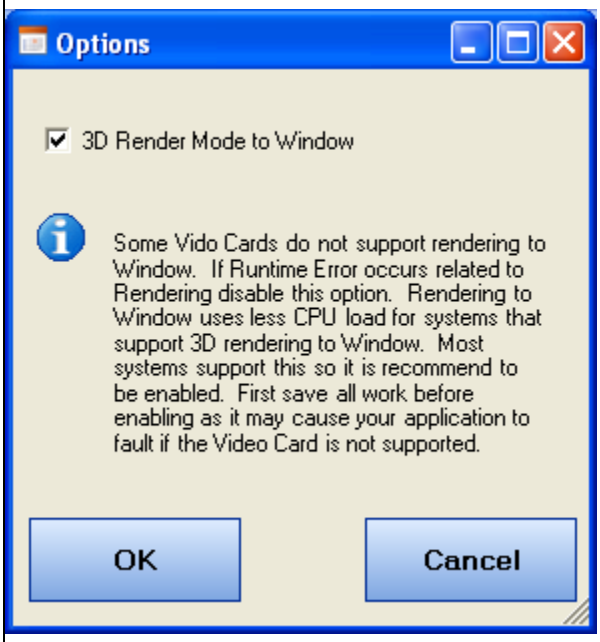
Step	Task
1	Start OPC Systems HMI application if it is not already running.
2	Select the Add button under the Trends Navigator bar on the left. 
3	Change the Trend Window Name to OPC Systems Demo.
4	Select the Modify Trend Window button and select Pens. 
5	Select the Local OPC Systems Service in the Trend Point Tags dialog.  *Note: if you desire to setup the Trend Window to work on remote application enter the OPC Systems Service source IP Address, Network Node Name, or registered Internet domain name and to select the desired local or remote OPC Systems Service. In this way the configuration can be deployed to multiple remote computers for remote trending.
6	Expand the Ramp Tag and select the Value Parameter and right click to Add Trend Point.  Only Parameters that have been enabled as Trend Point in the Tag configuration will appear for trending.

	You can also optionally select DirectOPC to connect directly to OPC Server Items without configuring Tags. 
7	Expand the Sine Tag and select the Value Parameter and right click to Add Trend Point. 

8	Set the LineBorderColor of Sine.Value Pen to Red. 
9	Set the YAxisRangeHigh to 1 and YAxisRangeLow to -1 of Sine.Value Pen. 
10	Select OK in the lower left of the Trend Point Tags dialog. 
11	Select the Modify Trend Window button and select Update Rates. 
12	Set the TimeFrame to 60 seconds. 

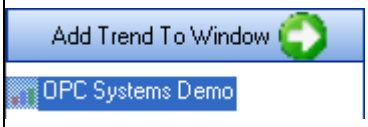
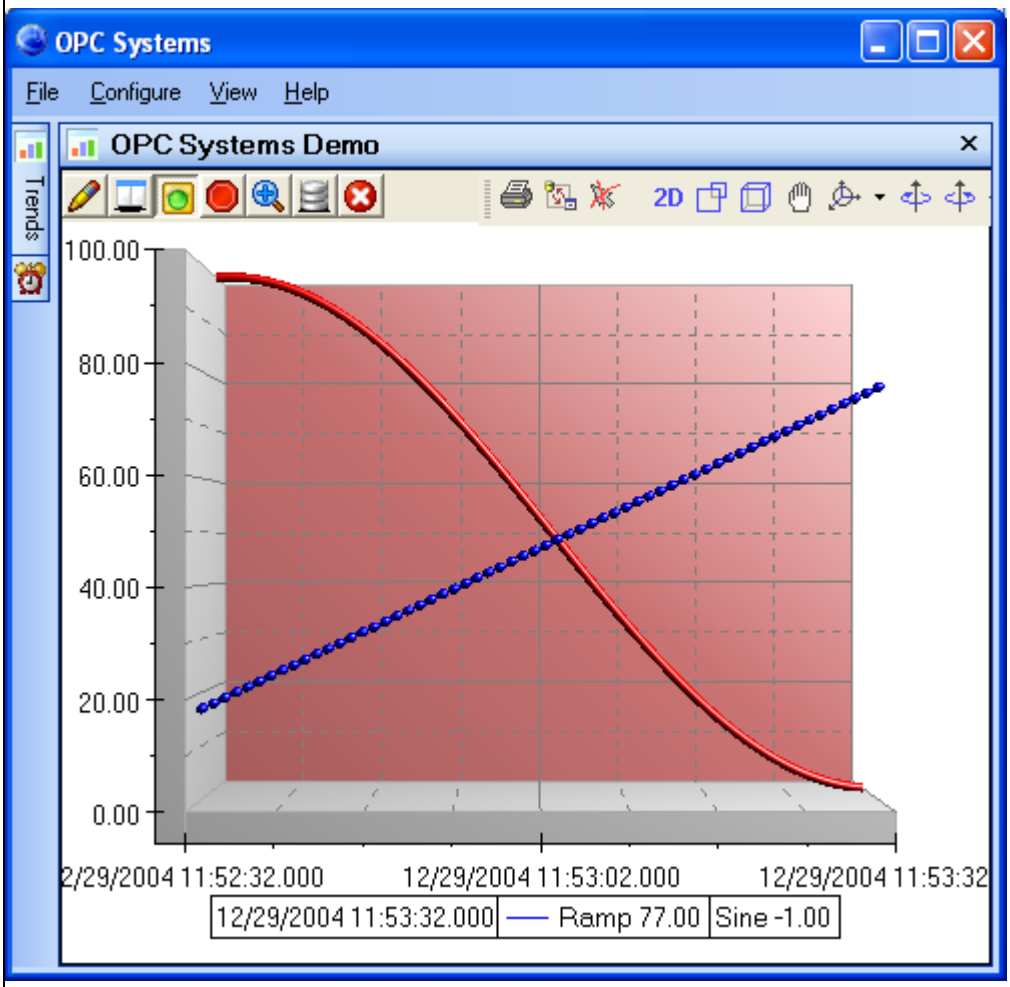
13	Select the Modify Trend Window button and select View. 
14	Set the Lighting to MetallicLustre 
15	Select the Modify Trend Window button and select Walls. 
16	Set the BackWallColor to LightCoral. 

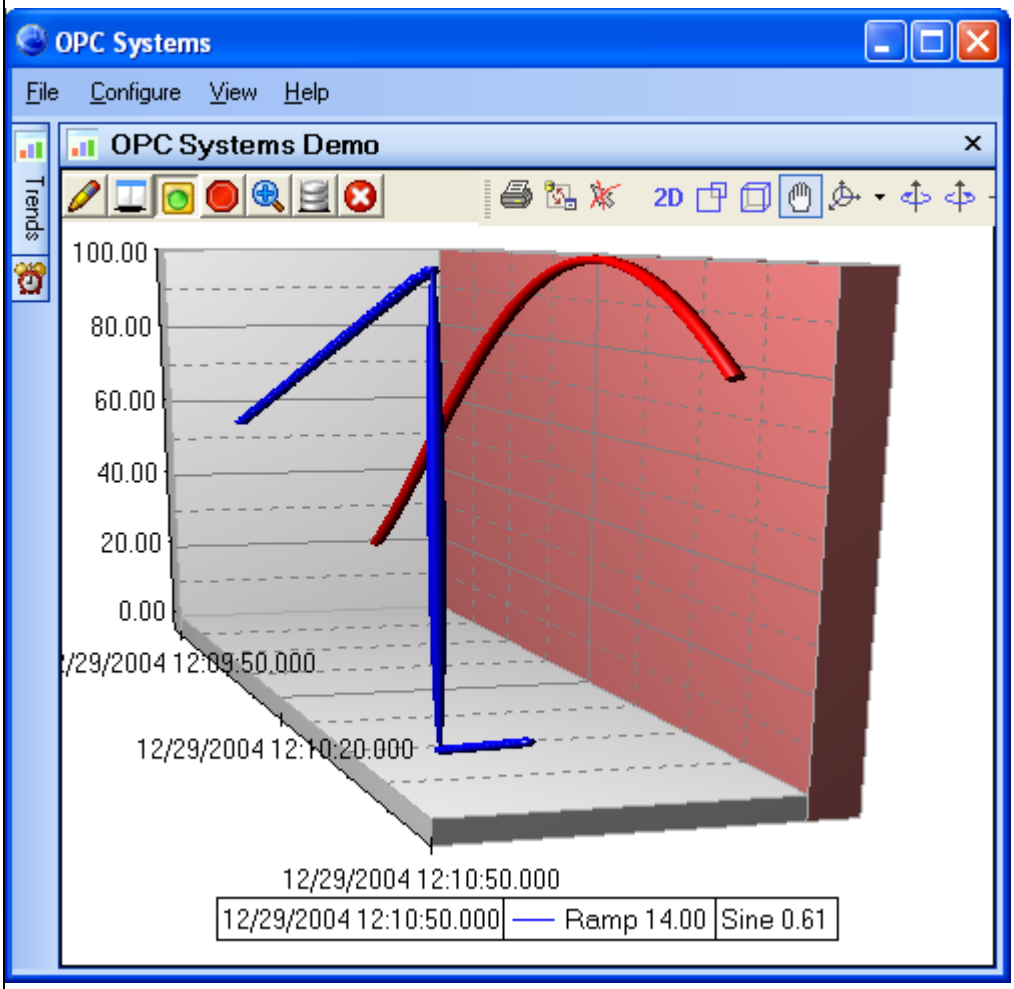
17	The Trend Window should appear as below. 
18	Click OK in the lower left corner. 
19	Select File Save  This is only necessary when you Add or Delete a Trend Window, Alarm Window, or View Window. Modifications to a Trend or Alarm Window will automatically be saved in its own configuration file.

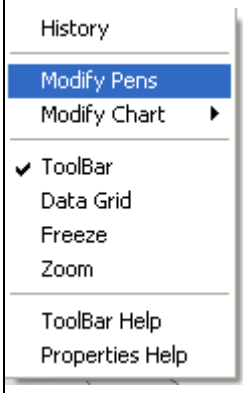
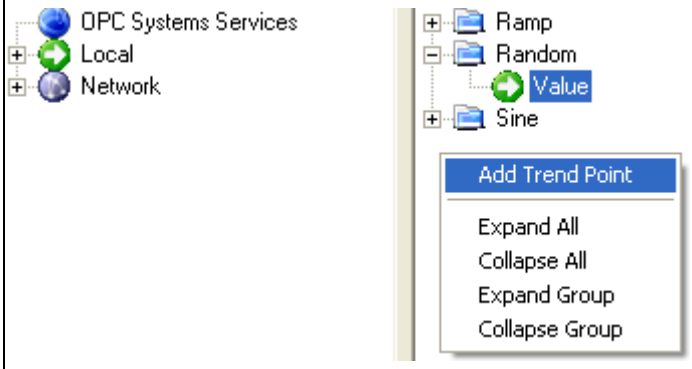
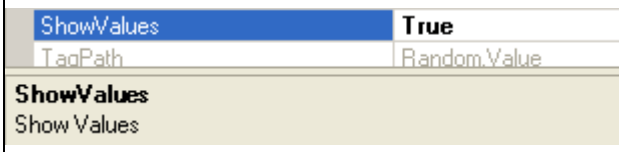
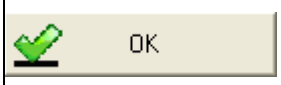
20	To test in the PC system supports 3D rendering to move processing to the graphic card select Configure-Options and enable the 3D Render Mode to Window.  Select Yes to Save the Render Mode when prompted.
21	Select the OPC Systems Demo Trend Window and select the Modify button.  If the Window appears correctly without the OPC Systems HMI application halting your systems supports 3D rendering to Window and no further action is required except to Cancel the Trend Window modification.  If the OPC Systems HMI application halts be sure to select Configure-Options once again and uncheck the 3D Render Mode to Window. Option and select OK.

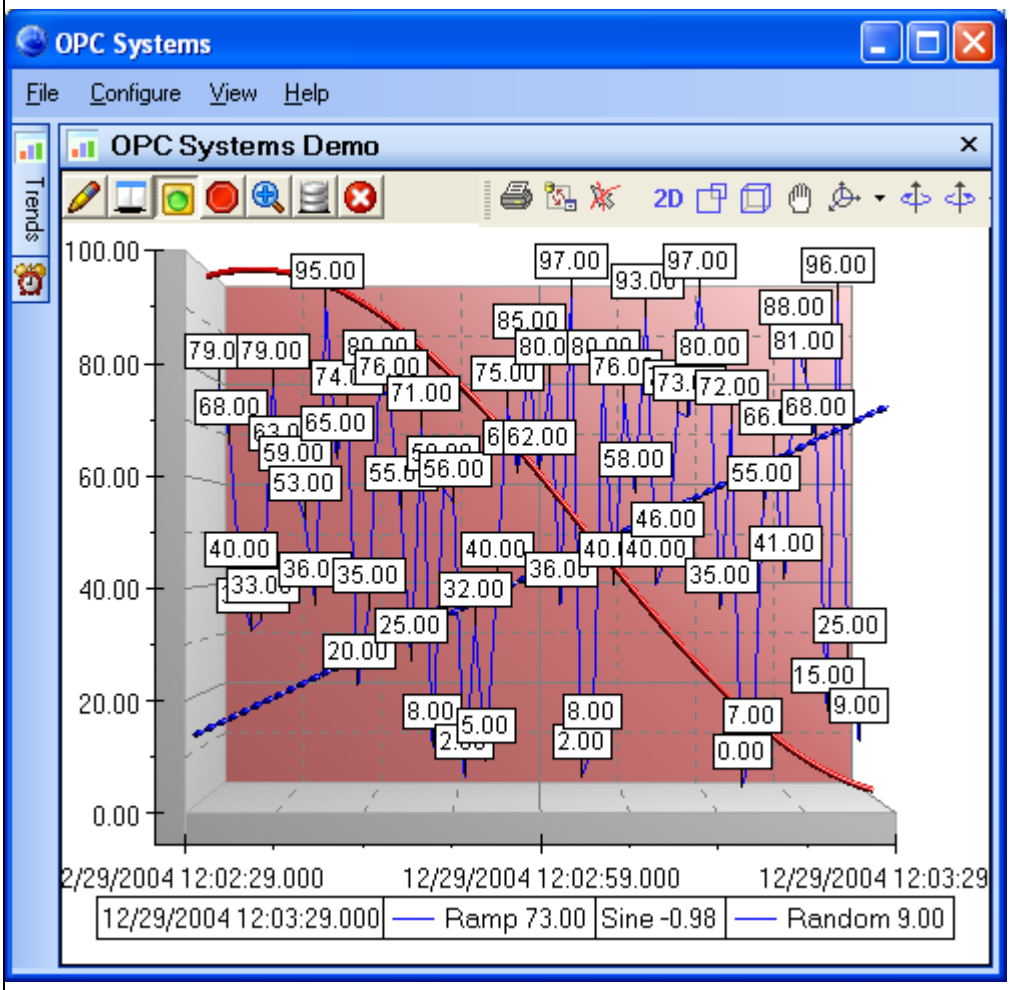
22	Your Trend Window is now ready to Run Trend Window in OPC Systems HMI.

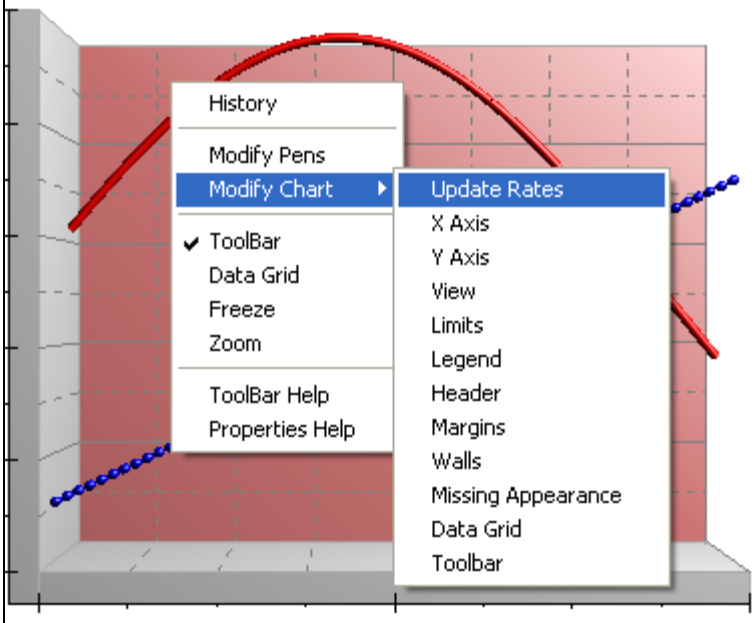
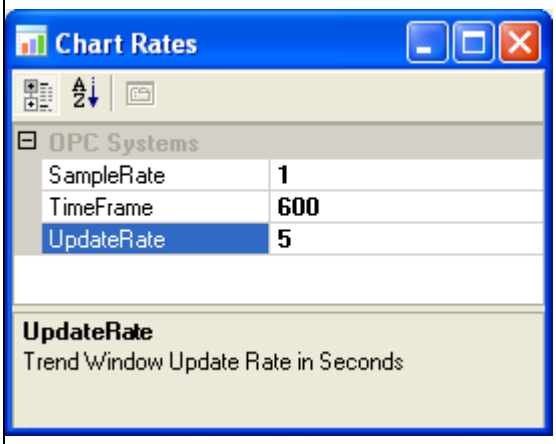
Running the Trend Window in OPC Systems HMI

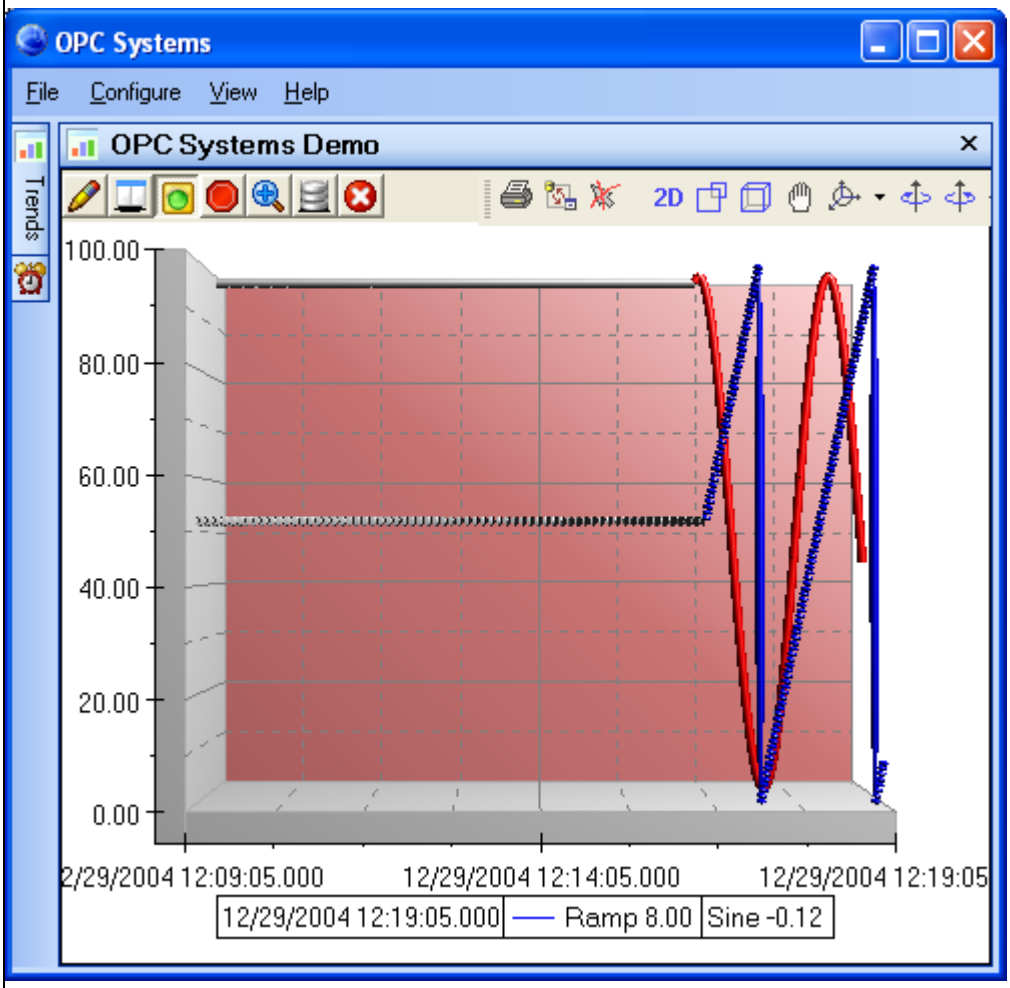
Step	Task
1	Start OPC Systems HMI application if it is not already running.
2	Select the Trends button in the lower left of the Navigator bar and select the OPC Systems Demo Trend Window and select the Add Trend To Window button. 
3	The Trend Window should appear as below with real time update from the OPC Systems Service. 

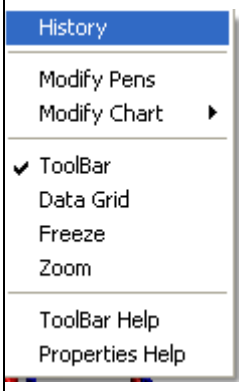
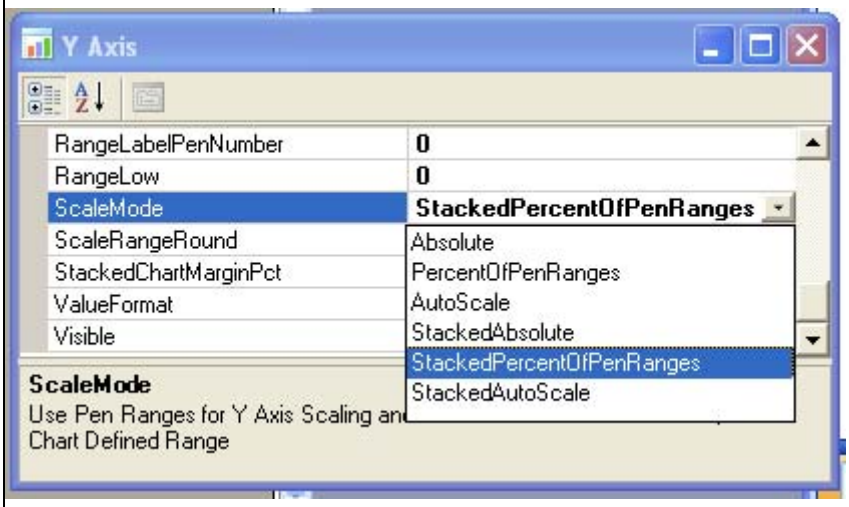
	If data does not appear within a few seconds make sure to open up the Firewall for port 58723 for the OPCSystemsHMI.exe and the OPCSystemsService.exe. Then restart the OPC Systems Service and try again. Contact Open Automation Software for OPC setup for XP document and help setting up OPC Systems.NET. www.opcsystems.com.
4	 Select the Rotate View icon on the toolbar and move the mouse to change the view perspective to see the 3D trending. 

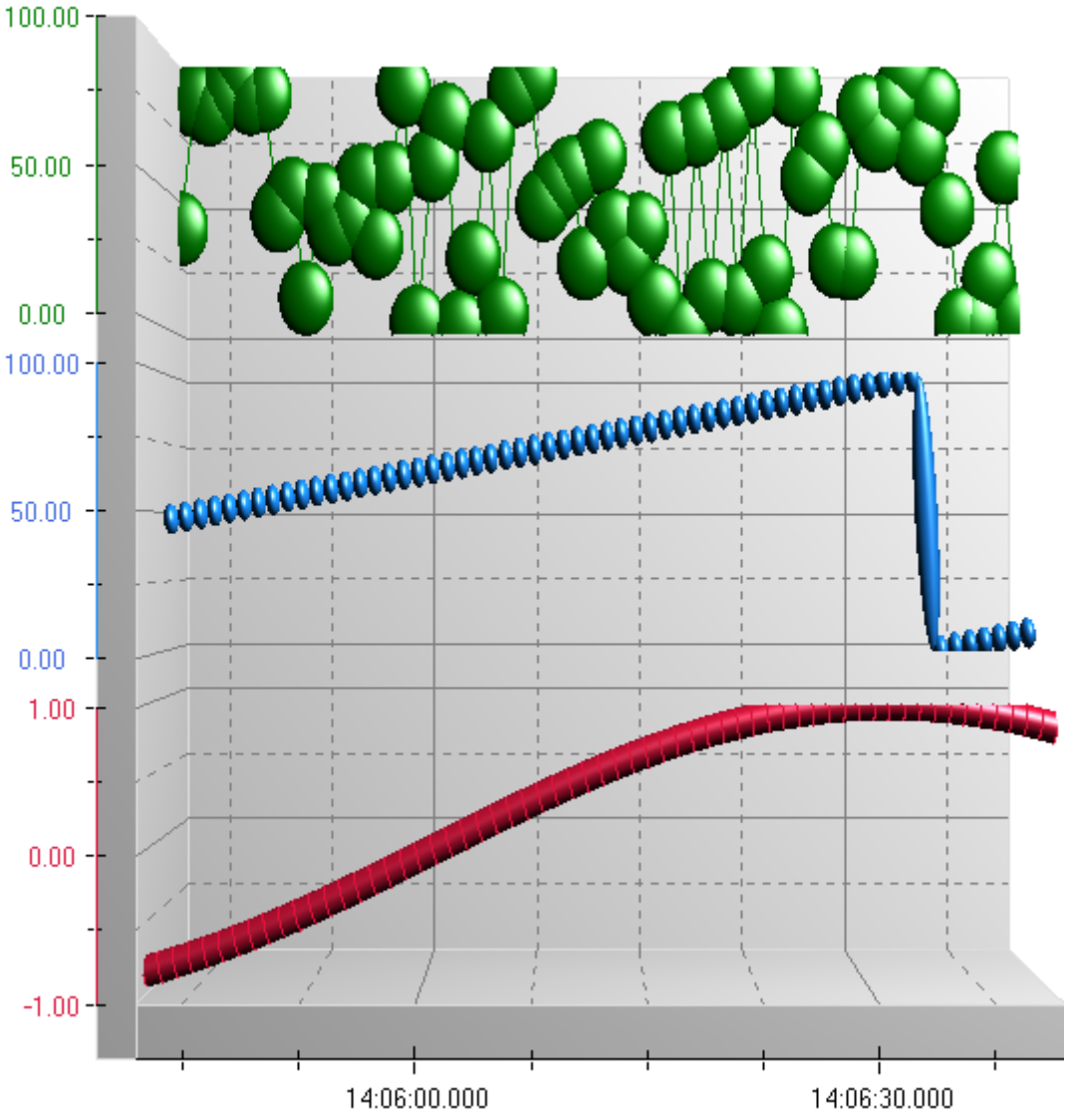
5	Right Click on the Trend Window and select Modify Pens. 
6	Select the Local OPC Systems Service and expand the Random Tag and select the Value Parameter and right click to Add Trend Point. 
7	Select Random.Value in the lower left Pen list and set the ShowValue attribute to True. 
8	Select OK in the lower left of the Trend Point Tags dialog. 

9	Notice that the values for the Random Pen have been buffered by the OPC Systems Service even though it had not previously been added to a Trend Window.  You can remove Pens by simply selecting Modify Pens and delete the Pen from Pen list and select OK.

10	Right Click on the Trend Window and select Modify Chart-Update Rates. 
11	Set the TimeFrame to 600. 

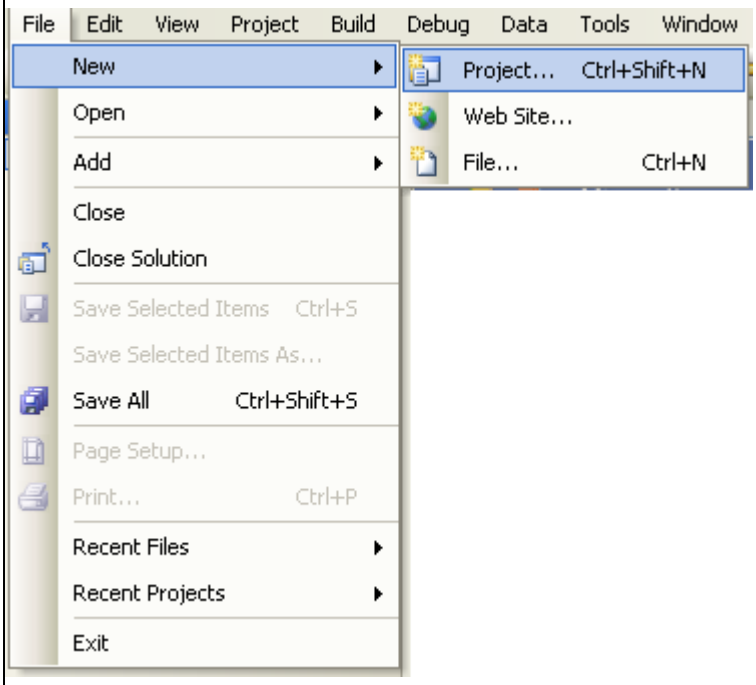
12	This will cause a new Time Frame in the OPC Systems Service for all Trend Points with a 1 second Sample Rate. The Trend Window Update Rate is set to 6 seconds as more samples will take more CPU load with fast update rates in 3D mode. 
13	Select File Save if you wish to save the on-line Window name and arrangement retentive. All modifications to the Trend Window properties will be retentive after this even if you do not select Save again. Only when a Window is added or removed do you need to select Save.

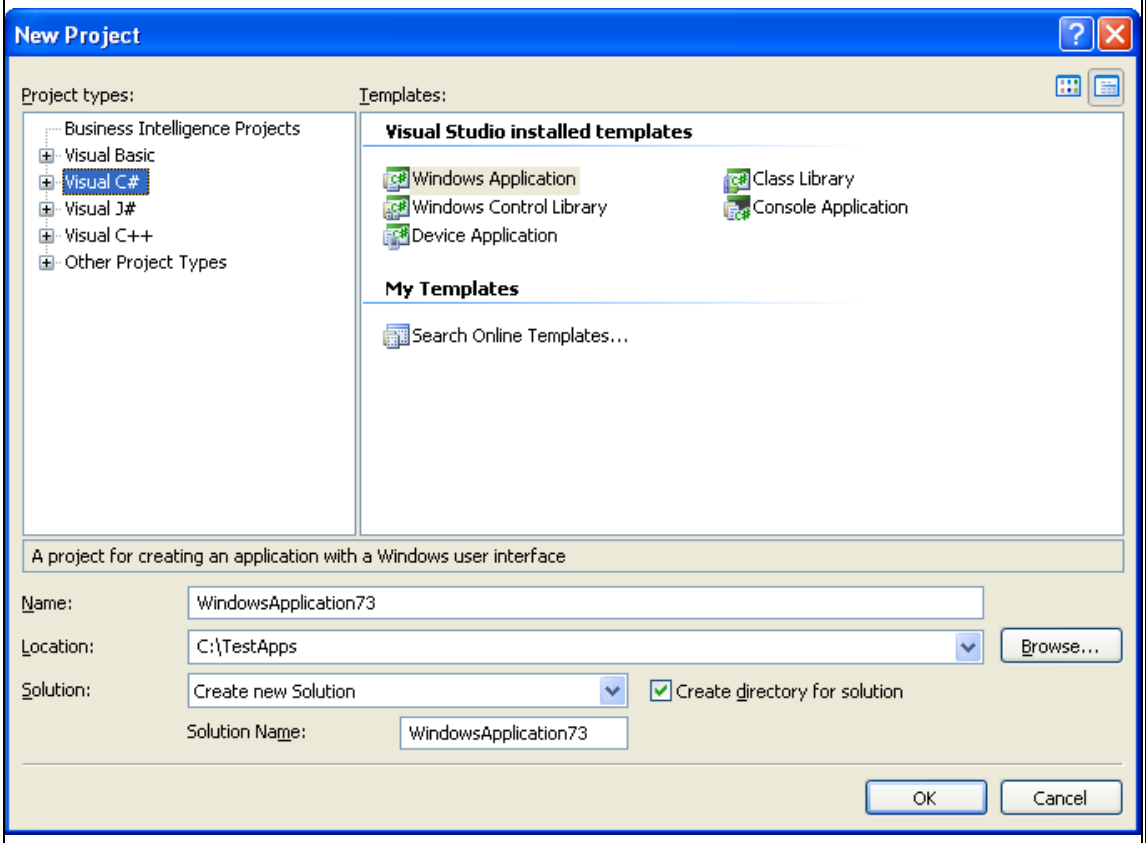
14	If you setup Data Logging in this example right click on the Trend Window and select History to retrieve historical data if you have setup Data Logging previously. 
15	Right click on the trend window and choose Modify Chart-Y Axis and change the Scale Mode from PercentOfPenRanges to StackedPercentOfPenRanges 

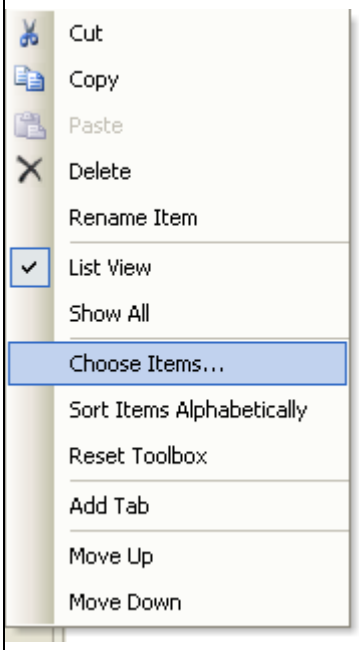
16	Right click on the trend window and choose Modify Pens and set the StackedChartNumber property in each Pen to a unique number for each Pen. Select OK to confirm the changes to the Pens. 
17	Explore the Trend Window Toolbar buttons and other features of the Trend Window including the Data Grid, Data Cursors, Data Zoom, This same trend control can also be added to your Visual Studio application.

Add Trend to Visual Studio Application

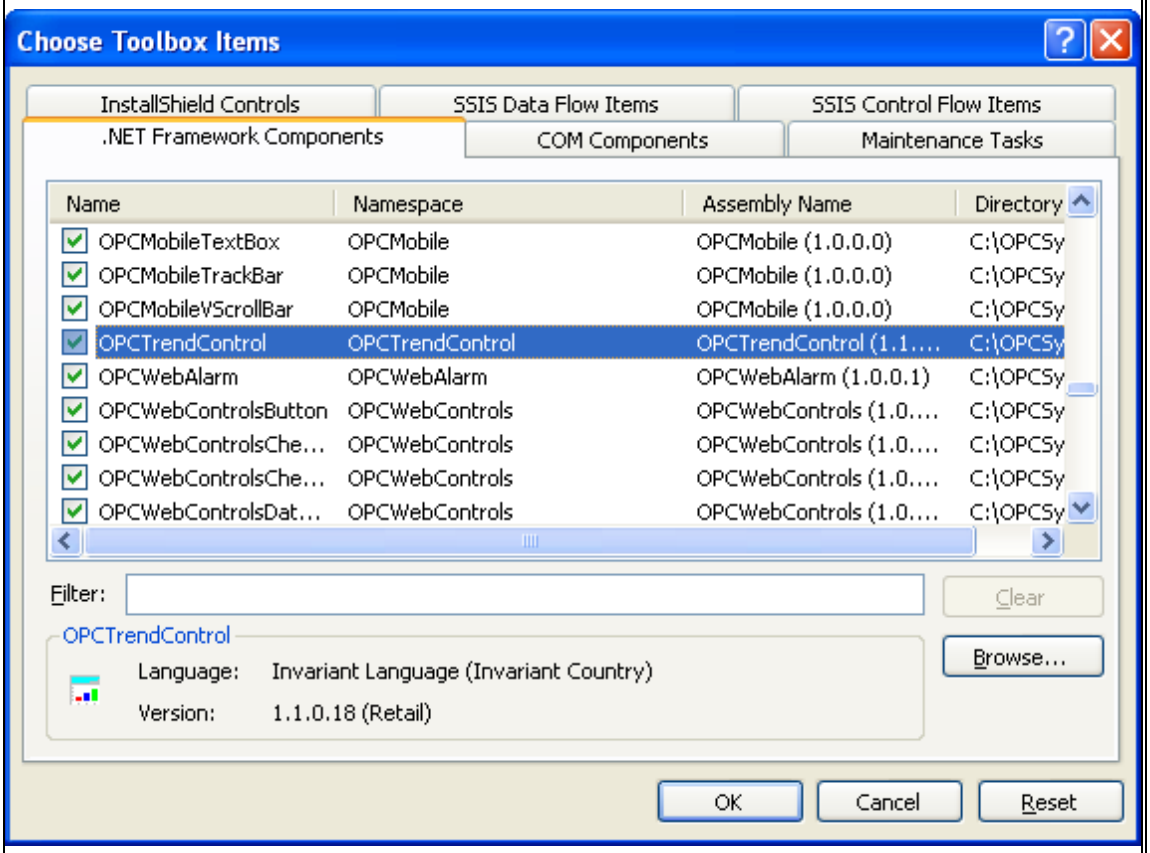
The following steps can be used to add a trend window to a C#, C++, or Visual Basic.NET application. Refer to the VB.NET example for programmatic interface of adding and removing pens. All properties are programmatically accessible. The following example demonstrates the trend window with no code required. The 1.1 Framework version of the trend window can also be used with Visual Studio 2003.

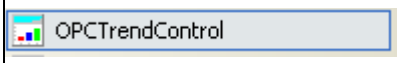
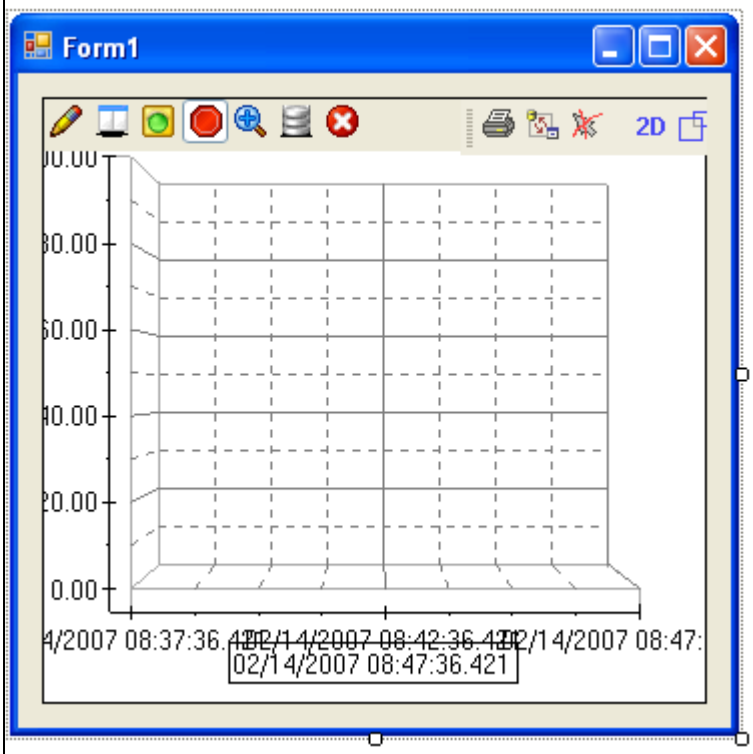
Step	Task
1	Start Visual Studio 2005 or Visual Studio 2008 and select File->New->Project to create a new C#, C++, or VB. 

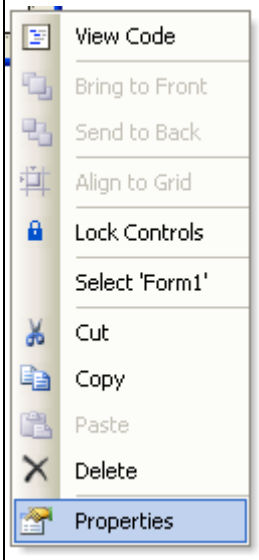
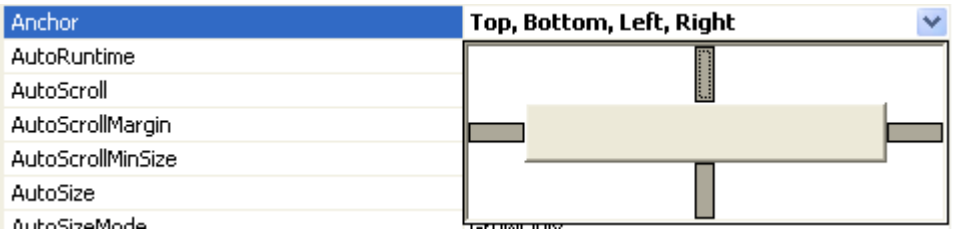
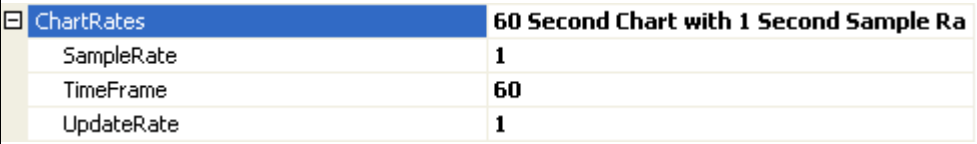
2	Select Windows Application as the project type. 

3	From the Toolbox if OPCTrendControl is not available right click in the Toolbox and select Choose Items. If it is available to step 4. 

From the .NET Framework Components select OPCTrendControl and then select OK.



4	Add the OPCTrendControl component onto the Form.  Resize both the form and trend window to the desired size. 

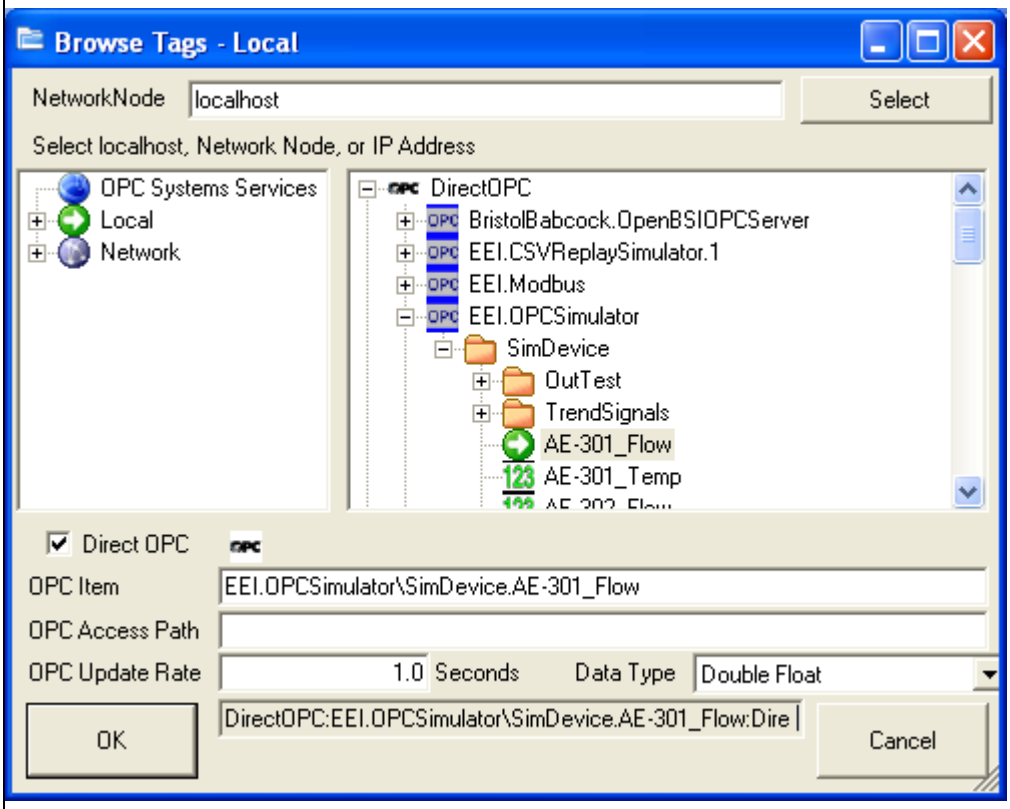
5	Right click on the trend window and select Properties. 
6	Set the Anchor property to Top, Bottom, Left, Right. 
7	Expand the ChartRates property and set the TimeFrame to 60 seconds. 
8	Select the Pens property and click on the small grey square with the 3 dots at the right. 

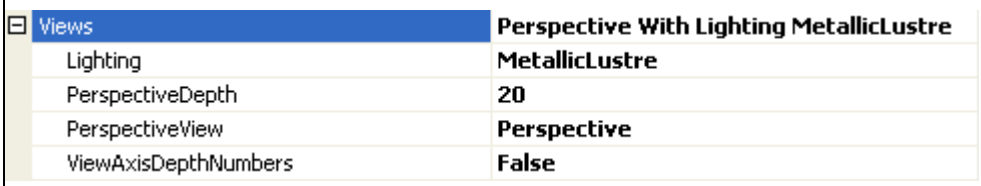
9	Select the Local OPC Systems Service to display a list of available Tags to

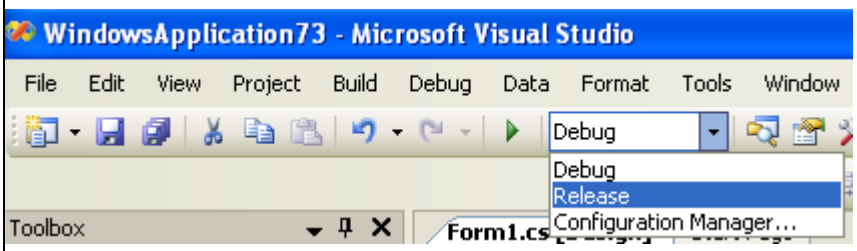
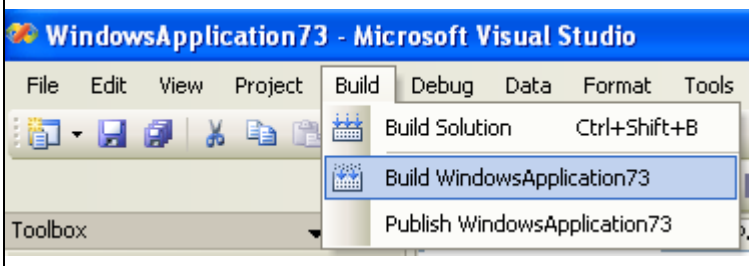
Note: If you do not see the desired Tag in this list go back to Configure-Tags and enable the Trend Point property for the Tags you wish to trend.

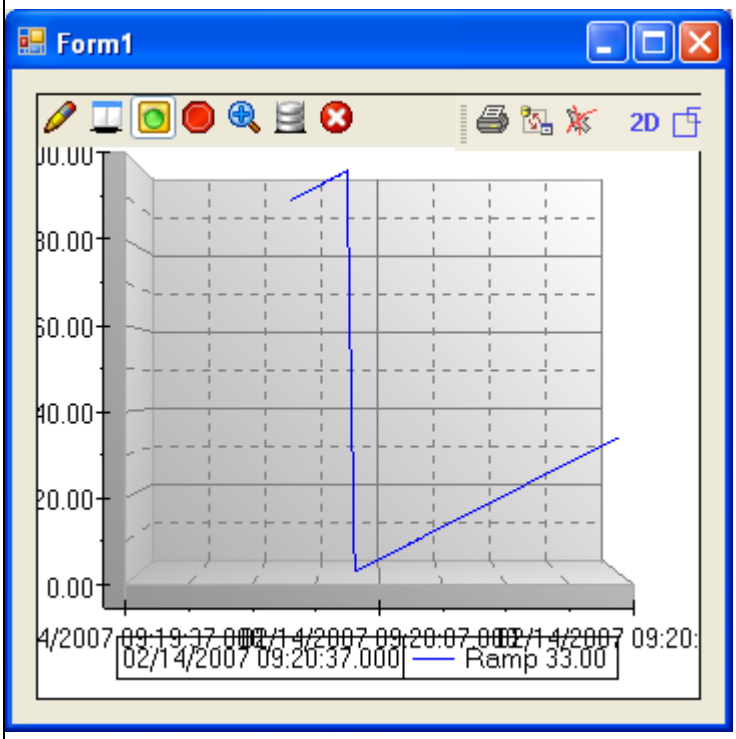
Note: If you want the application to be deployed across a network to remote PCs select the Network Node or enter an IP Address in the NetworkNode field and use the Select button to include the network node or IP Address of the OPC Systems Service source.

You can also use the DirectOPC interface to connect directly to OPC Server Items.



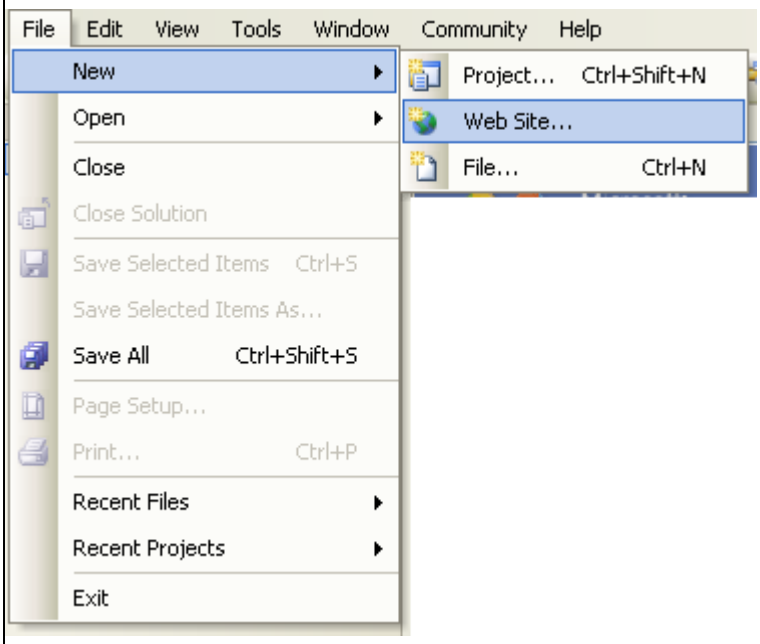
10	Expand the Ramp Tag and select Value.  Select the Add Pen button or right click on Value and select Add Pen. 
11	The pen Ramp.Value will appear in the lower left list of pens. You can select the pen to change of the pen properties that appear to the right. The YAxisRangeHigh and YAxisRangeLow properties are important when the trend windows YAxis.ScaleMode property is set to PercentOfPenRanges.
12	Select OK from the Pens dialog.
13	If the PC(s) that will run this have a video card that can support OpenGL rendering set the RenderMode3D property to True.  If you are unsure leave this property to False for now.
14	Expand the Views property and set the trend window Lighting and PerspectiveView to the desired settings. 

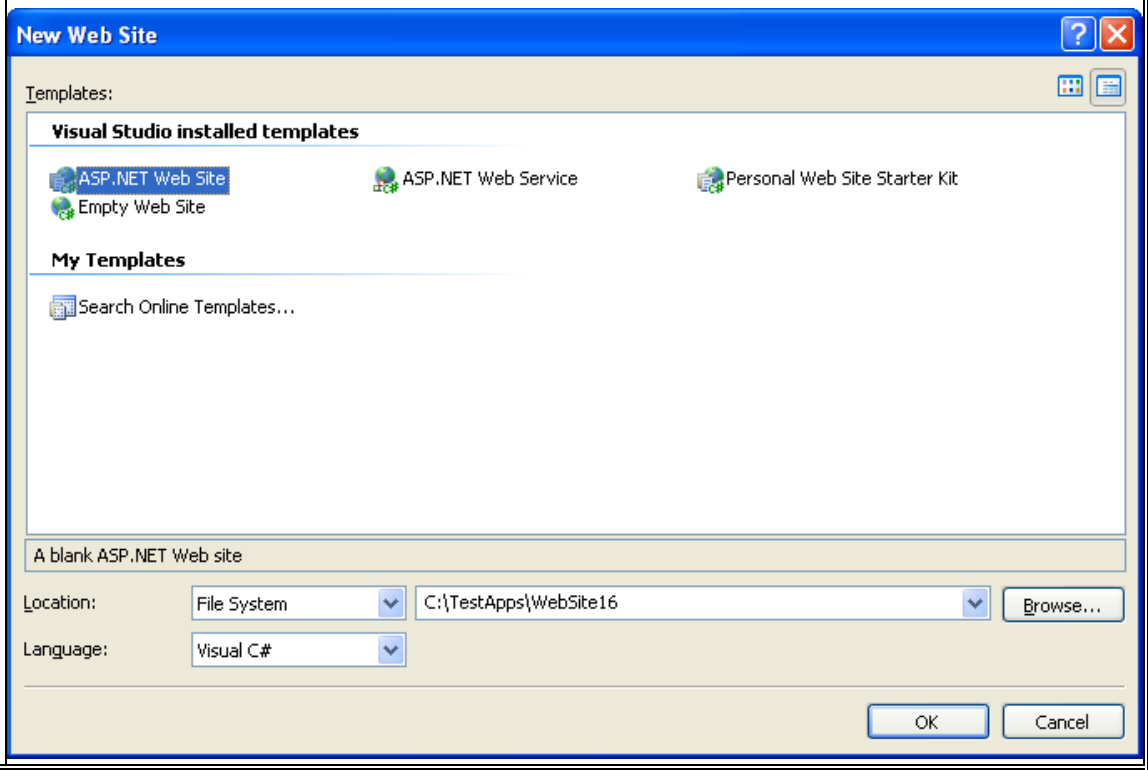
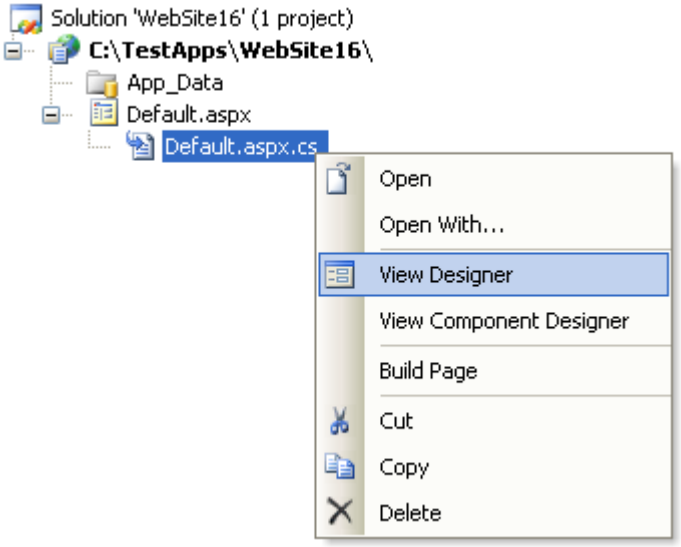
15	If you desire for the operators changes to the trend window during runtime to remain set the ConfgiurationFile property to a valid file path. Make sure each system the application will run on that the directory path is valid.  **Note:** If you set this property to a file make sure you deploy the file with the application in the directory you specify.. Leave this property blank if you wish to have the default properties set during configuration remain on the application restarting.
16	Set the compile mode on the Visual Studio toolbar to Release. 
17	Select Build from the VS menu and select to Build the application. 

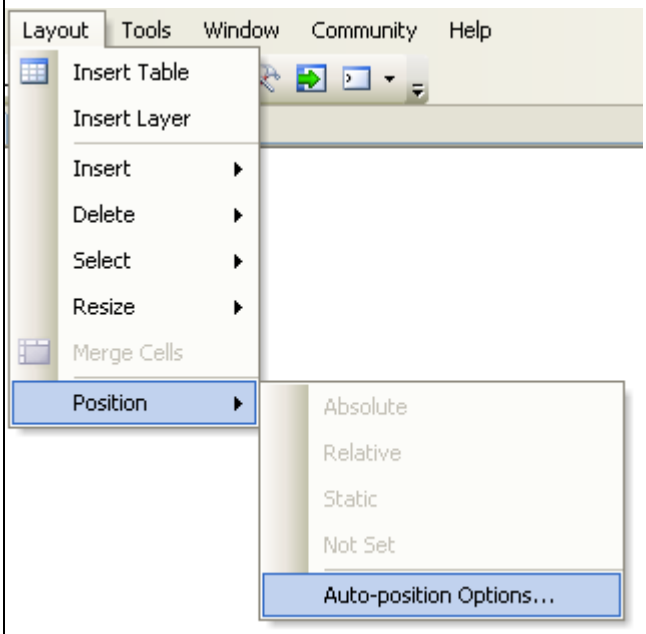
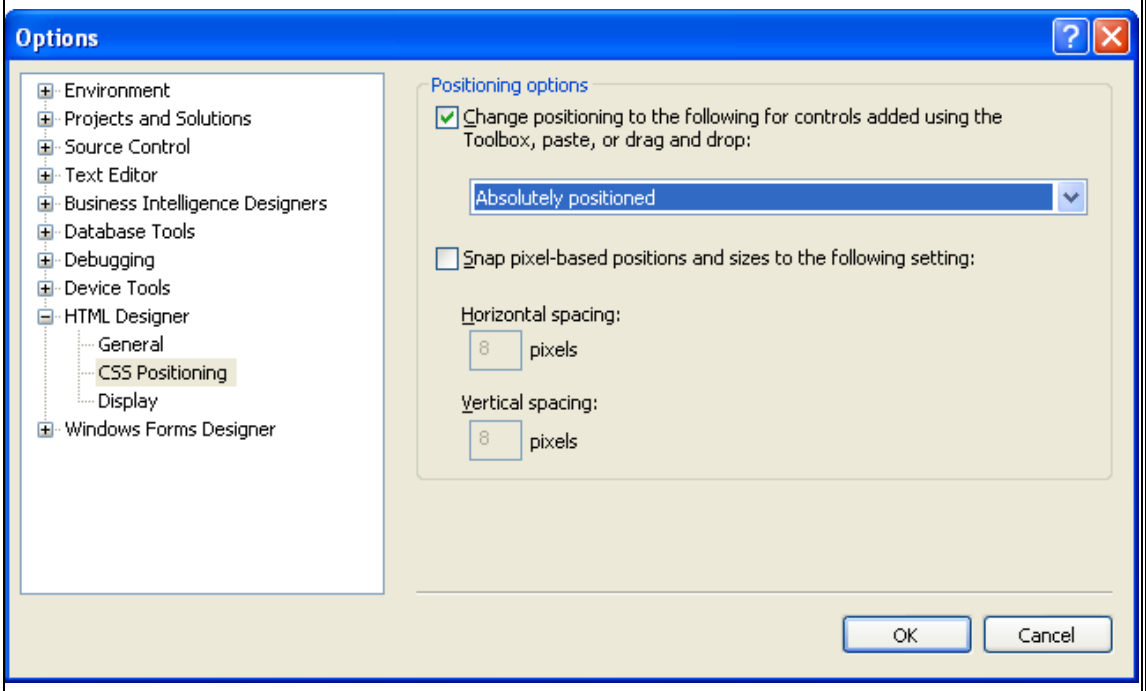
18	Use Windows Explorer to browse for the application located in the bin\Release directory and run the application. 
19	To deploy the application to remote nodes first make sure the Pen selection as described in step 9 is set to a Network Node or IP Address. Then simply copy the files in the bin\Release directory to the target systems or follow the Smart Client deployment section in this help file to deploy your application using Click Once Deployment.

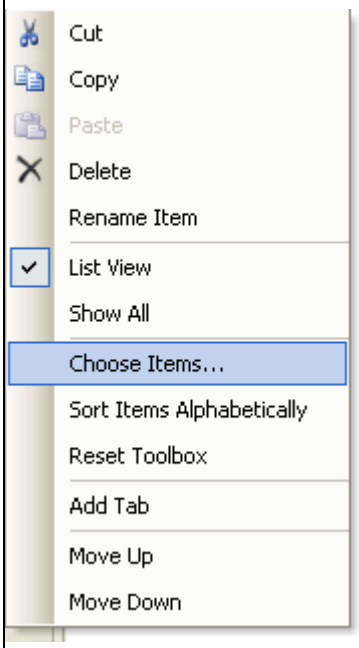
Add Trend to ASP.NET Web Application

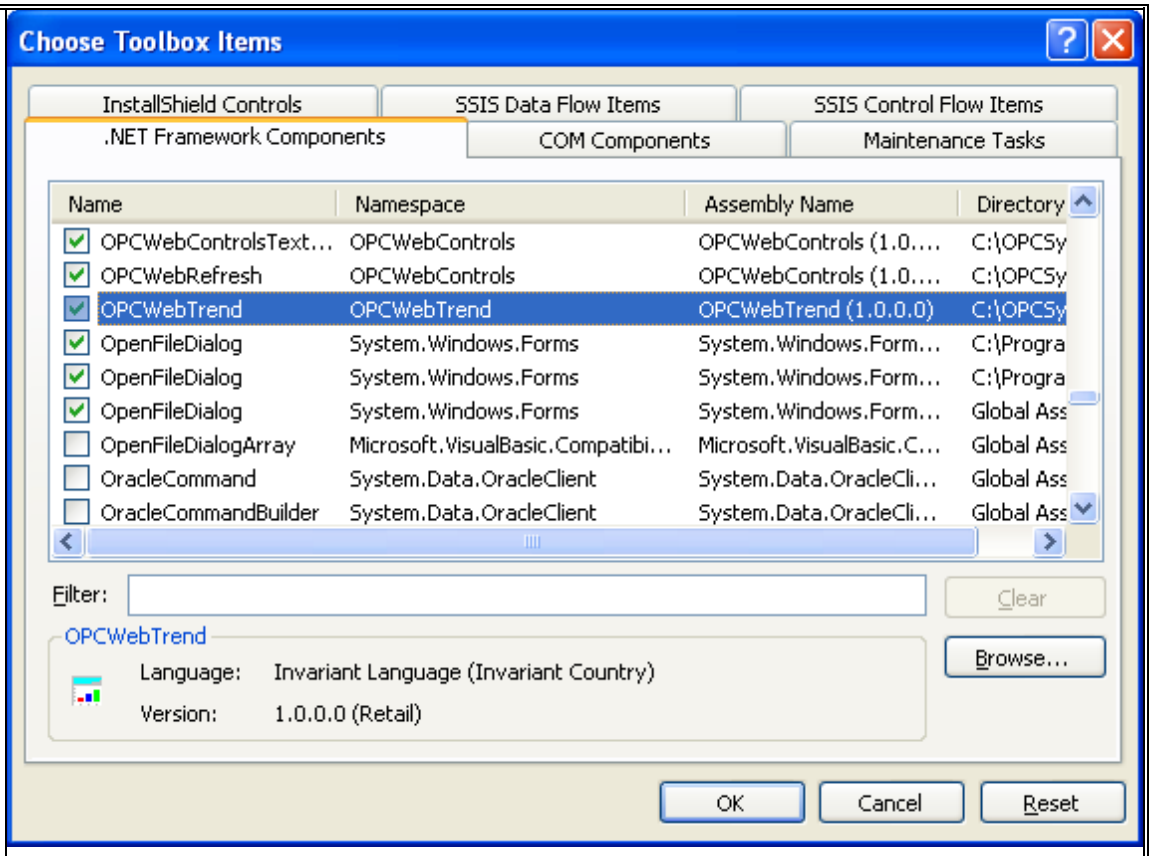
The following steps can be used to add a trend window to a C#, C++, or Visual Basic.NET web application. All properties are programmatically accessible. The following example demonstrates the trend window with no code required. The trend window can also be used with Visual Studio 2003 web applications.

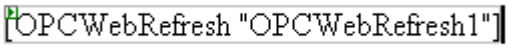
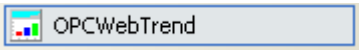
Step	Task
1	Start Visual Studio 2005 and select File->New->Web Site to create a new C#, J#, or VB ASP.NET web application. 

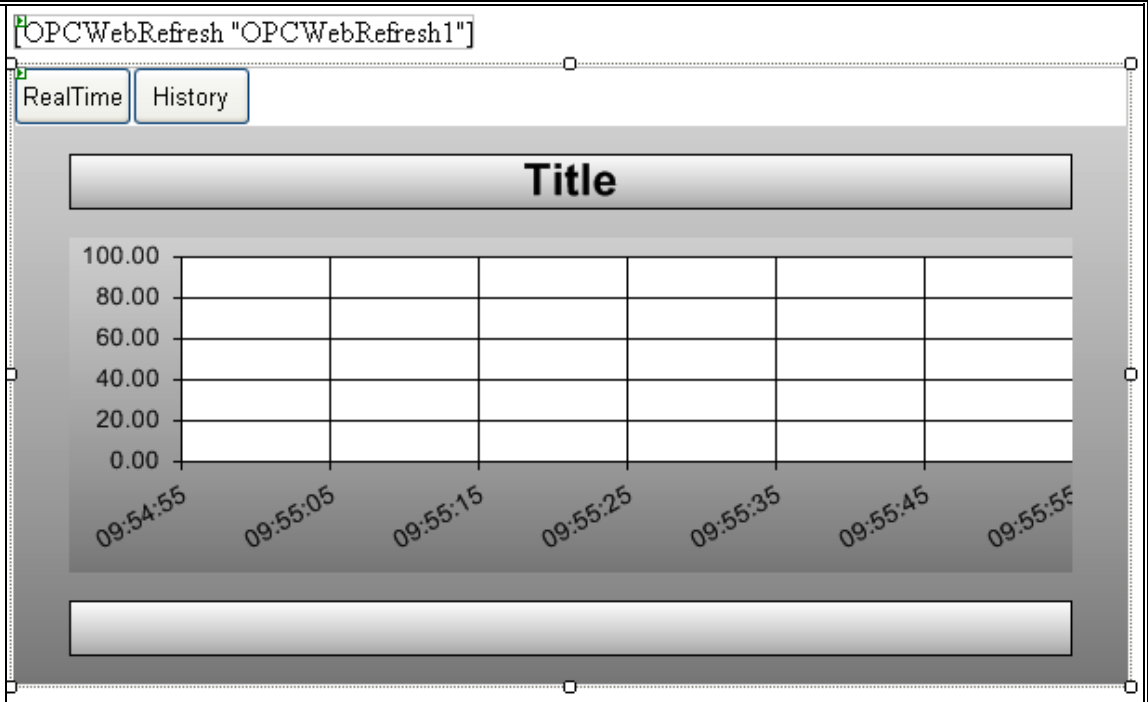
2	Select ASP.NET Web Site and specify your desired Language in the lower right. 
3	Expand the Default.aspx web page in the Solution Explorer and select View Designer. **Note:** If you do not see the Solution Explorer select View->Solution Explorer. 

4	From the VS menu select Layout->Position->Auto-position Options. 
5	Under the HTML Designer set the CSS Positioning to Absolutely positioned. **Note:** You are welcome to use any positioning style you desire, this simply pointed out for new web developers. 

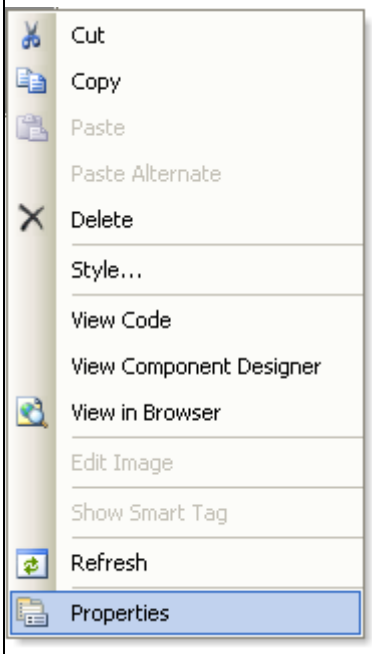
6	From the Toolbox if OPCWebTrend and OPCWebRefresh is not available right click in the Toolbox and select Choose Items. If it is available to step 4.  From the .NET Framework Components select OPCWebTrend, OPCWebControlsLabel, and OPCWebRefresh and then select OK.

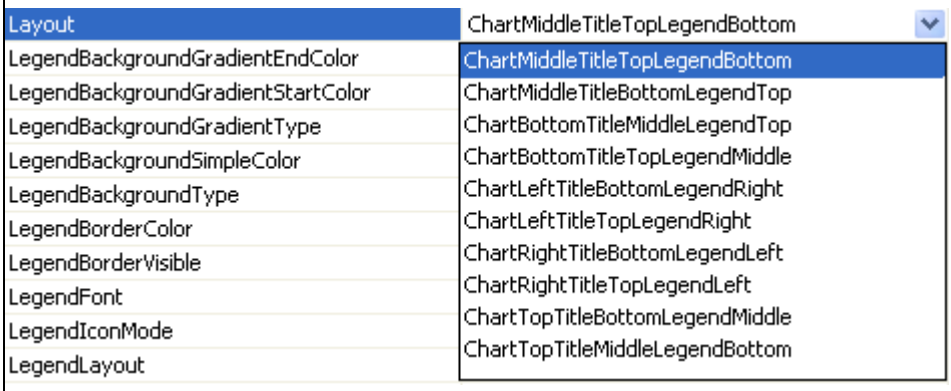
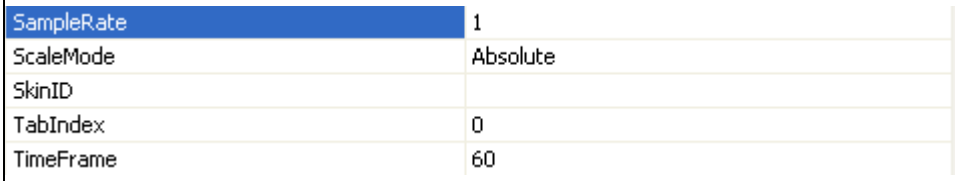


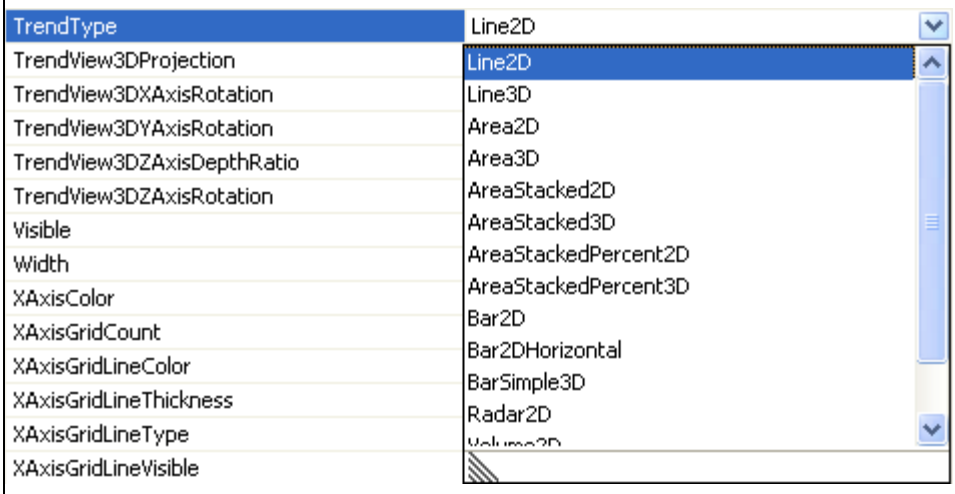
7	Add one OPCWebRefresh control to the WebForm.   To make sure your web.config file is modified for proper AJAX runtime add one OPCWebControlsLabel to the WebForm and then delete it.
8	Add the OPCWebTrend component onto the WebForm.  Resize the trend window to the desired size.

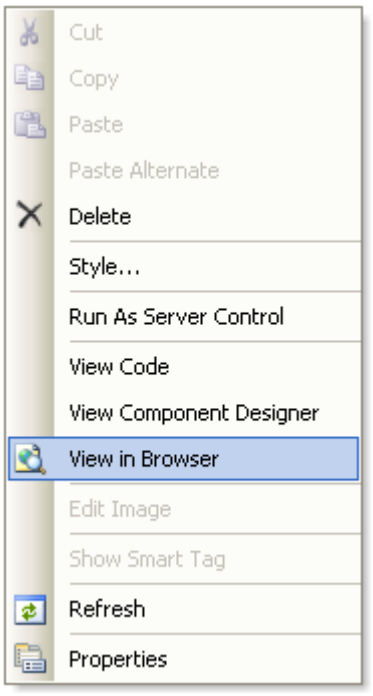


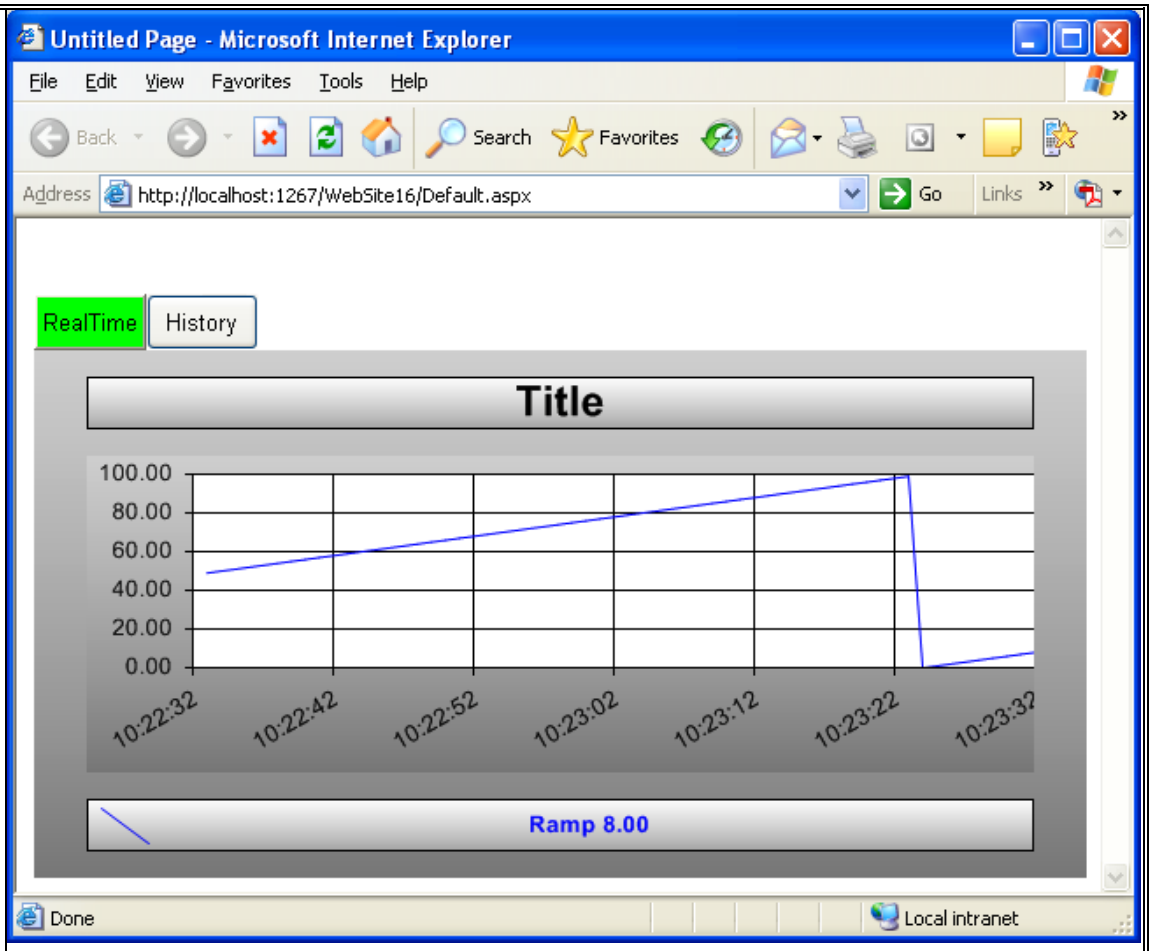
9 Right click on the trend window and select Properties.



10	Set the Layout style to the desired setting. 
11	Set the SampleRate and TimeFrame properties to the desired settings. The default is 60 seconds at a 1 second sample rate.. 
12	Select the Pens property and click on the small grey square with the 3 dots at the right. 
13	Select the Local OPC Systems Service to display a list of available Tags to  **Note:** If you do not see the desired Tag in this list go back to Configure-Tags and enable the Trend Point property for the Tags you wish to trend. **Note:** If you want the web application to be deployed across a network to remote PCs select the Network Node or enter an IP Address in the NetworkNode field and use the Select button to include the network node or IP Address of the OPC Systems Service source. If the web application will be running on the local system and simply need access to the web application through remote browsers use the Local service.

14	Expand the Ramp Tag and select Value.  Select the Add Pen button or right click on Value and select Add Pen.  **Note:** You can also use the DirectOPC interface to connect to OPC Server Items directly without having to create OPC Systems.NET Tags.
15	The pen Ramp.Value will appear in the lower left list of pens. You can select the pen to change of the pen properties that appear to the right. The YAxisRangeHigh and YAxisRangeLow properties are important when the trend windows YAxis.ScaleMode property is set to PercentOfPenRanges.
16	Select OK from the Pens dialog.
17	Set the TrendType property to the desired setting. 

18	Right click on the WebForm and select View in Browser. 

	
19	Set the application for Release mode, modify the web.config file an ensure that the Debug parameter is set to false, and build your application. Typically the application files are under the Inetpub\wwwroot directory. Refer to the OPC Web Trend.NET section in the OPC Systems.NET help file for Special considerations and Deployment steps if you encounter any issues.

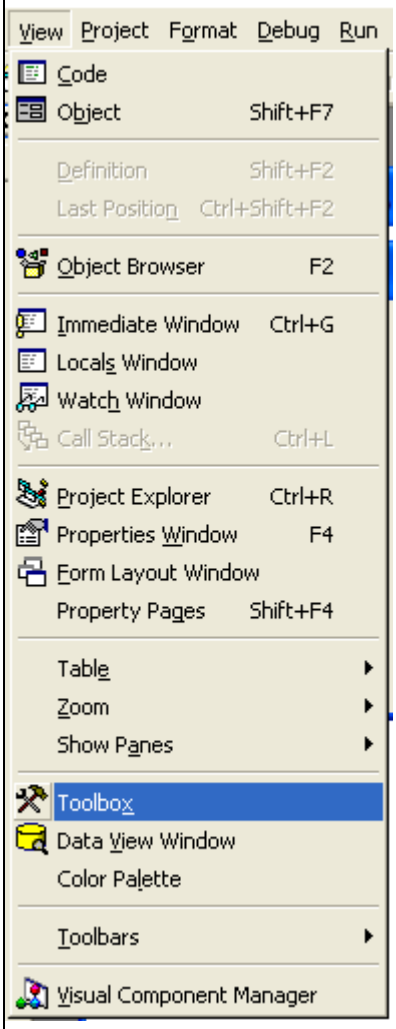
Add Trend ActiveX to Legacy Application

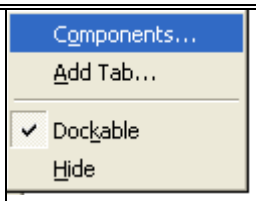
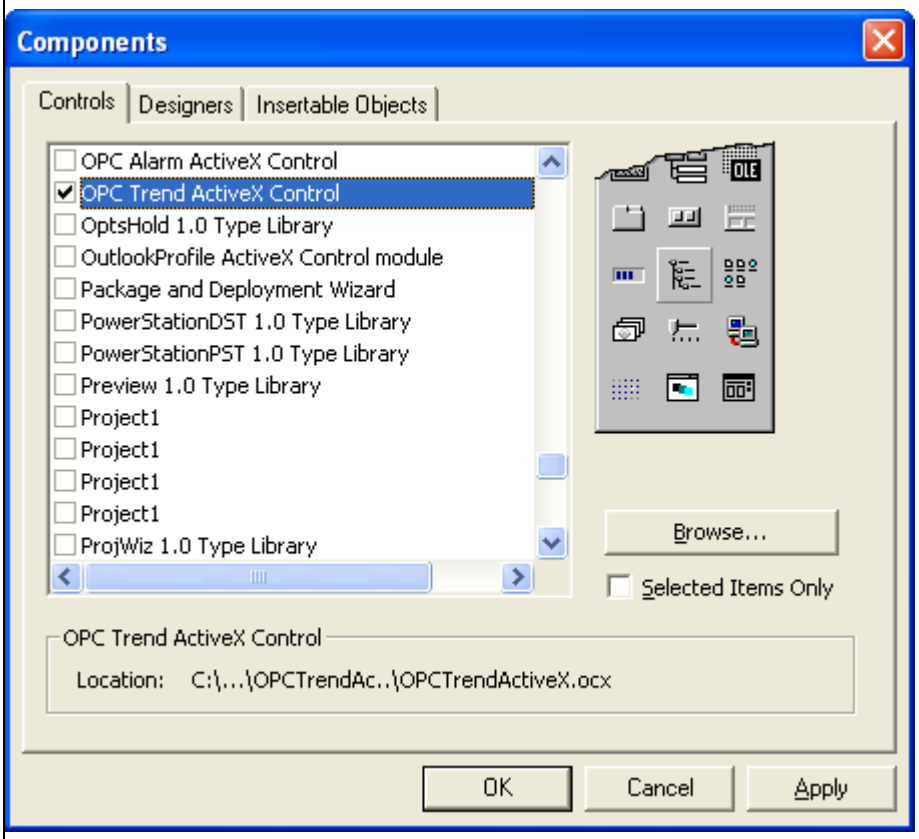
The OPC Trend ActiveX control can be integrated directly into any ActiveX container, most commonly HMI graphic applications or Visual Studio 6.0.

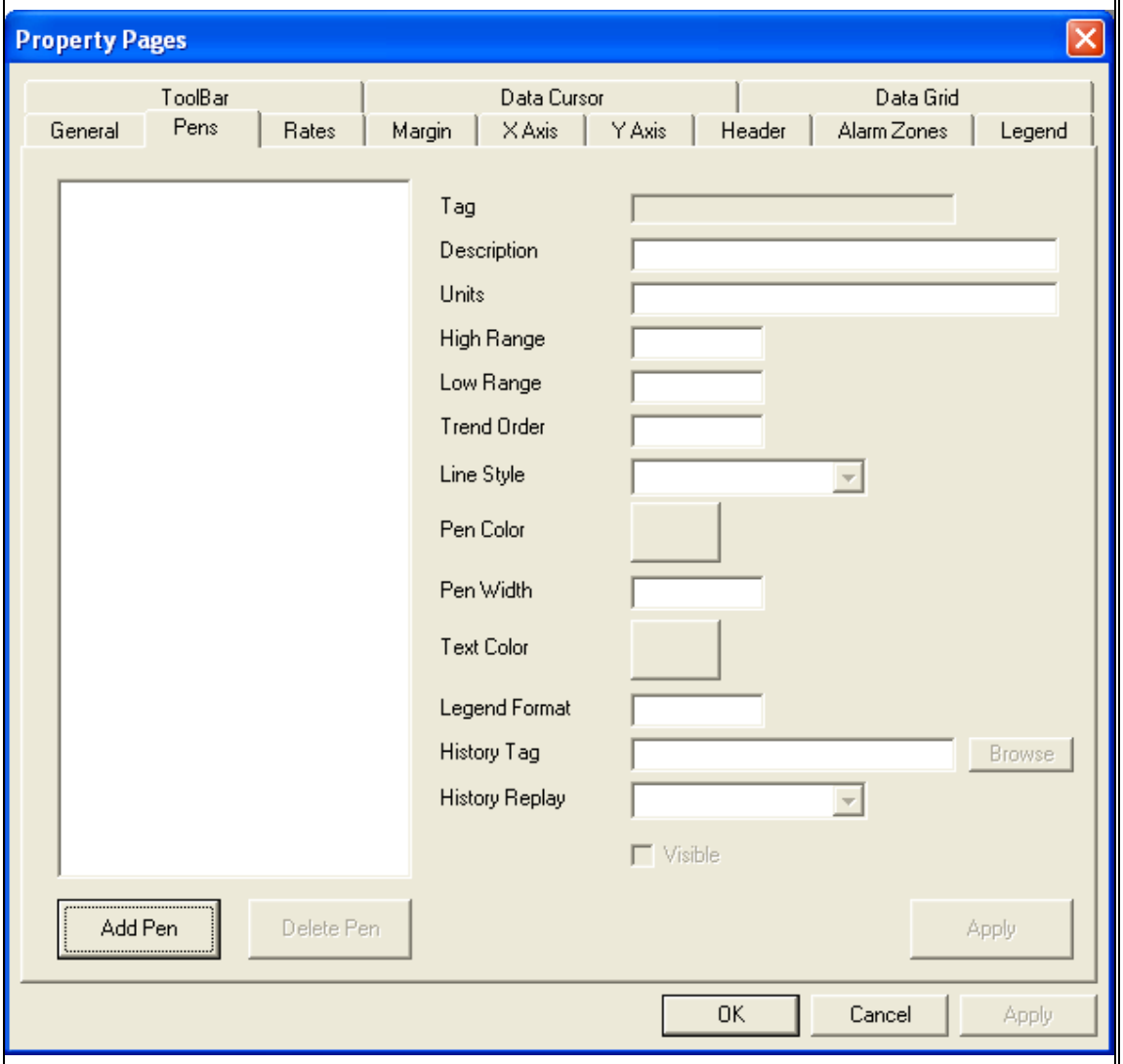
Note: ActiveX controls are not 100% managed and you should use the .NET controls for Visual Studio.NET applications and in more update to date containers that support .NET controls.

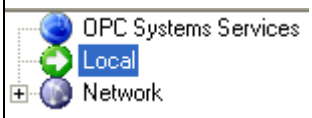
Below is an example of using the OPC Trend ActiveX Control integrated into Visual Basic 6.0 or other ActiveX container of your choice.

Step	Task
1	Start Visual Basic 6.0 with a new or existing Project or other ActiveX container application of your choice. If using other ActiveX container than Visual Studio simply select the appropriate method of the application to add an ActiveX control and select OPCTrendActiveX.OPCTrend and skip to Step 6.

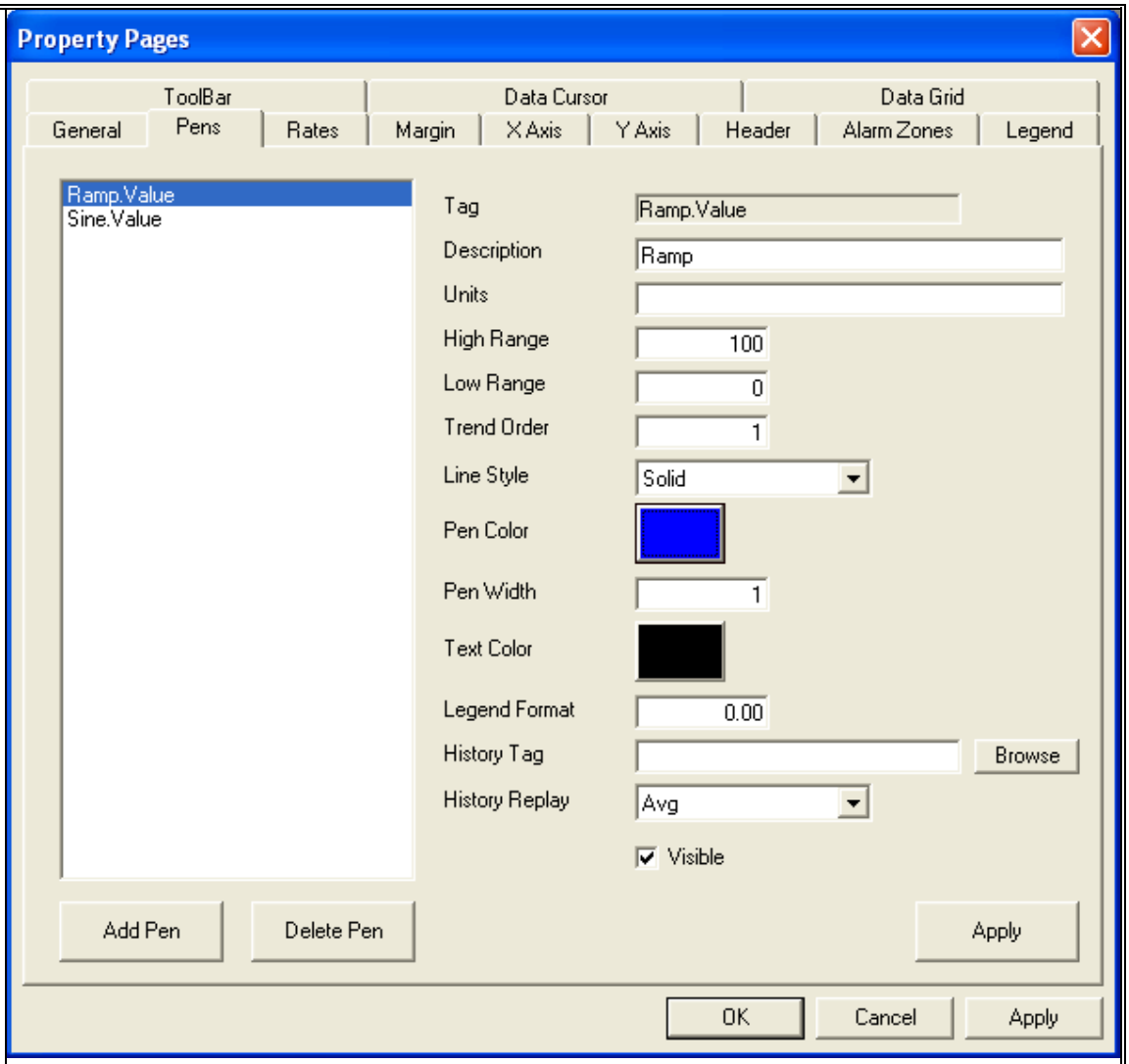
2	To add the OPC Trend.ActiveX Control to your Visual Studio 6.0 development system select View-Toolbox from within Visual Studio. 
3	Right-Click on the Toolbox and select Components.

	
4	Select OPC Trend ActiveX Control from the list of Controls 
5	From the ToolBox select the OPC Trend ActiveX Control icon and add the trend to any Form.
6	Resize the Trend Window to any size your desire. Please note that some of the Runtime Properties will not be available for modifying during runtime if the Trend Window is too small.

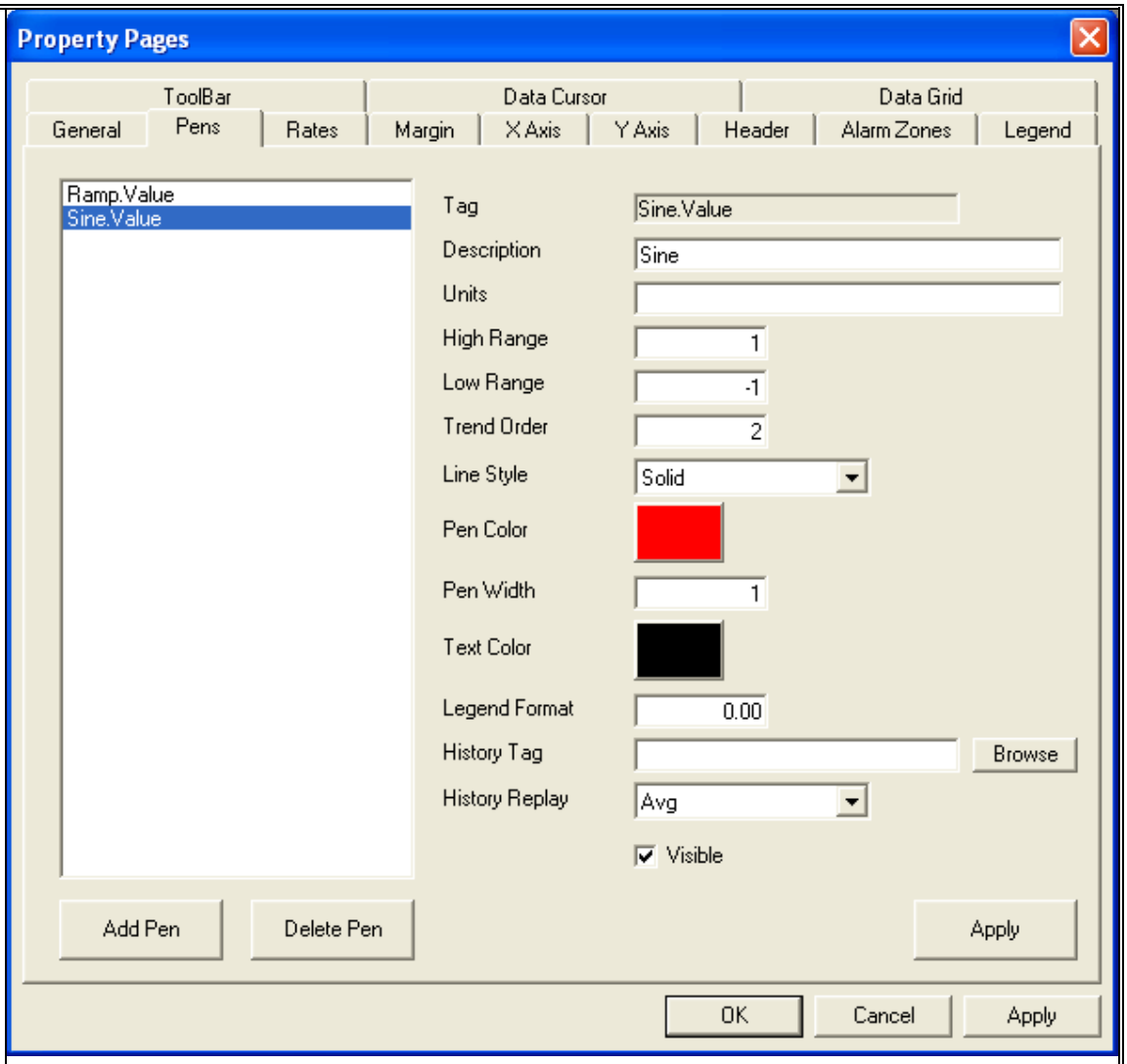
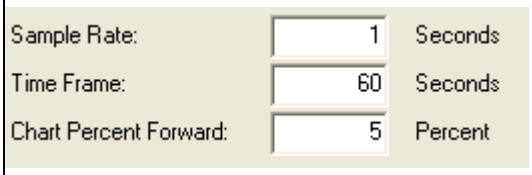
7	Select Properties of Trend Window and select Pens. 
8	Select Add Pen.

	Select the Local OPC Systems Service in the Trend Point Tags dialog.  *Note: if you desire to setup the Trend Window to work on remote application select the Network browse to select the appropriate network name of the local or remote OPC Systems Service.
6	Expand the Ramp Tag and select the Value Parameter and select OK in the lower left of the dialog.  Only Parameters that have been enabled as Trend Point in the Tag configuration will appear for trending.
7	Select Add Pen and expand the Sine Tag and select the Value Parameter and select OK in the lower left of the dialog. 

8	Select Ramp.Value from the Pen list and set the Pen Color to Blue, then select Apply button for the Pen.



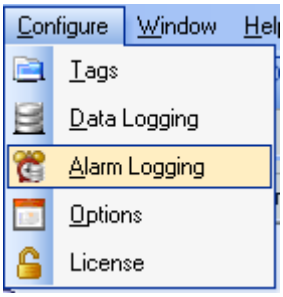
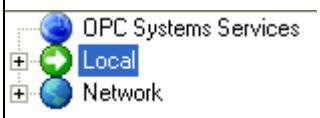
9	Select Sine.Value from the Pen list and set the High Range to 1 and Low Range to -1, then select the Apply button for the Pen..

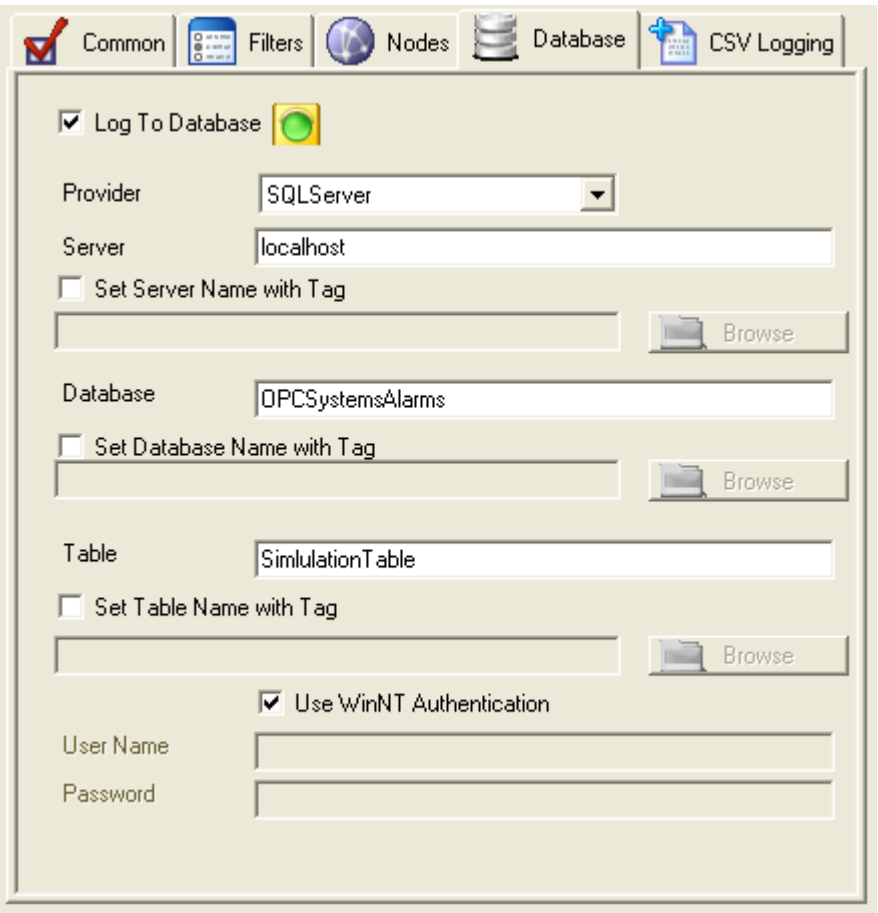
	
10	Select the Rates Property Tab and set the set the TimeFrame to 60 seconds. 
11	Select OK at the bottom of the Property Page.
12	Save your application and compile if needed and you are now ready to run the application.

Chapter 5

Chapter 5 - Alarming

Configure Alarm Logging

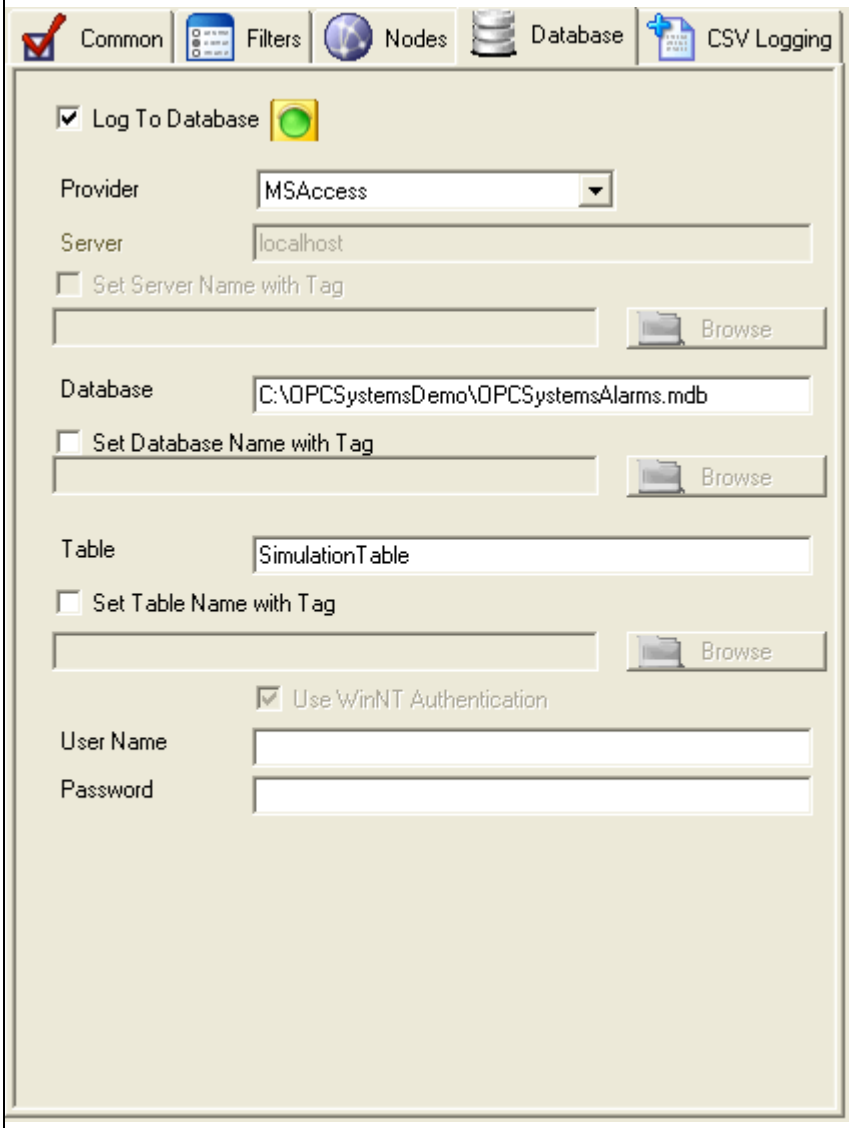
Step	Task
1	Start Configure OPC Systems application. 
2	Select Configure-Alarm Logging. 
3	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left.   **Note** The Configure application can be used to connect to remote systems using the network node name or IP address of the remote node the OPC Systems Service is running on. Simply enter the IP Address or network node name of the remote OPC Systems Service you wish to connect to and click on the Select key.

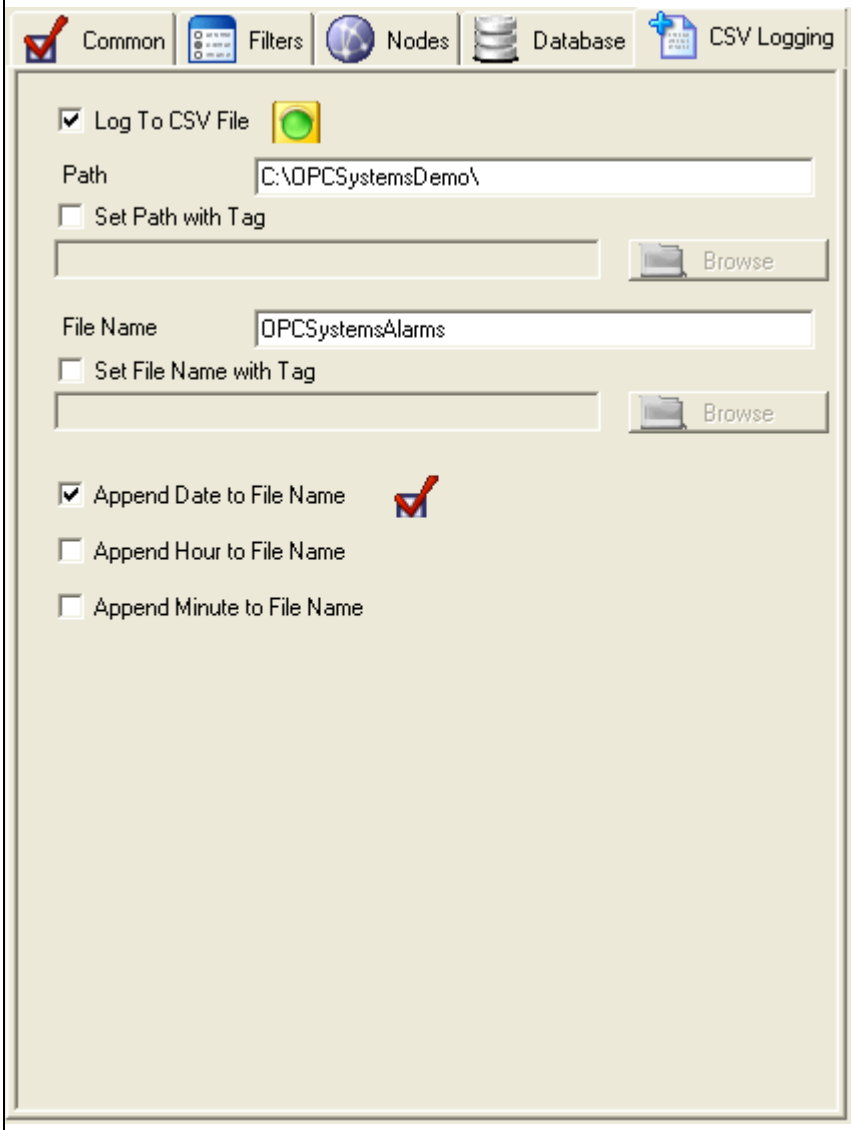
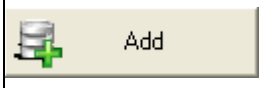
4	Enter the Logging Group Name of Simulation in the field in the upper right. 
5	Check Logging Active in the Common Properties Tab. 
6	Select the Database Tab. 
7	Use the following configuration for SQL Server Desktop. The Database and Table will automatically be created for you.  The Server name may need to be adjusted from localhost to the proper Server name of the SQL Server engine you wish to log to. This Server name can be found when first bringing up the Connect dialog of the SQL Server Management Studio.

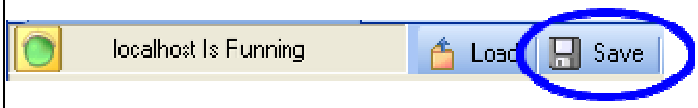
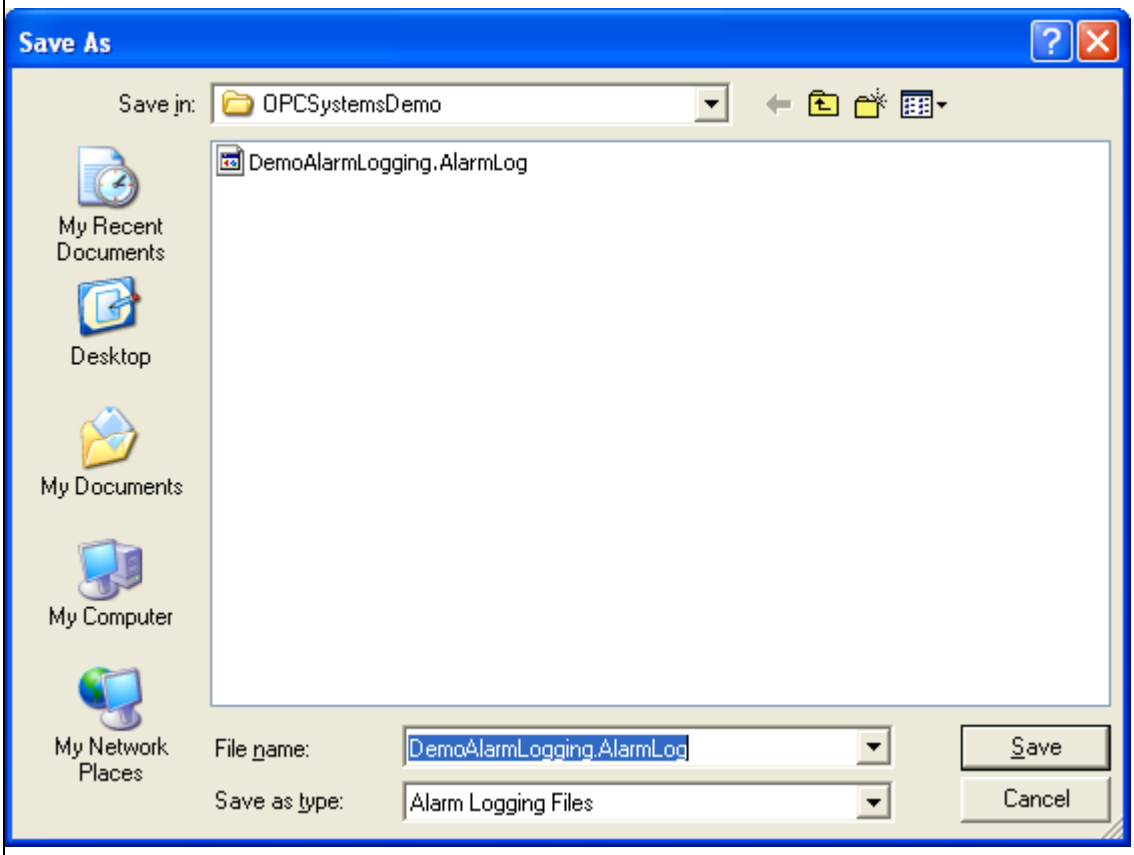
Also the login method can be with Windows Authentication or SQL Server mode. Contact your database administrator if you are unsure of what login type to use.

You can download a free version of SQL Server Express from www.microsoft.com.

Use the following configuration for Microsoft Access. The Database and Table will be automatically created for you.

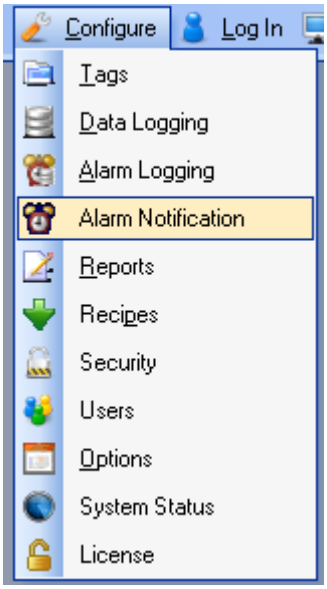
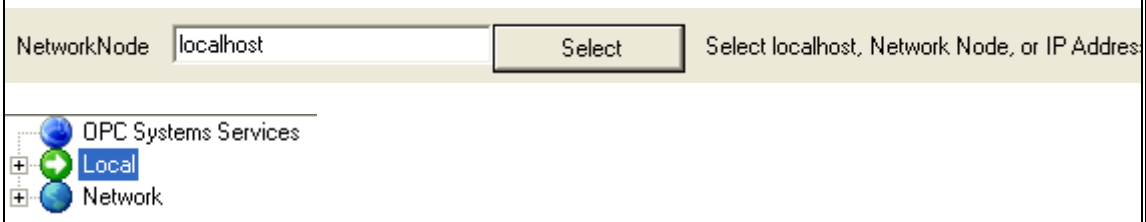


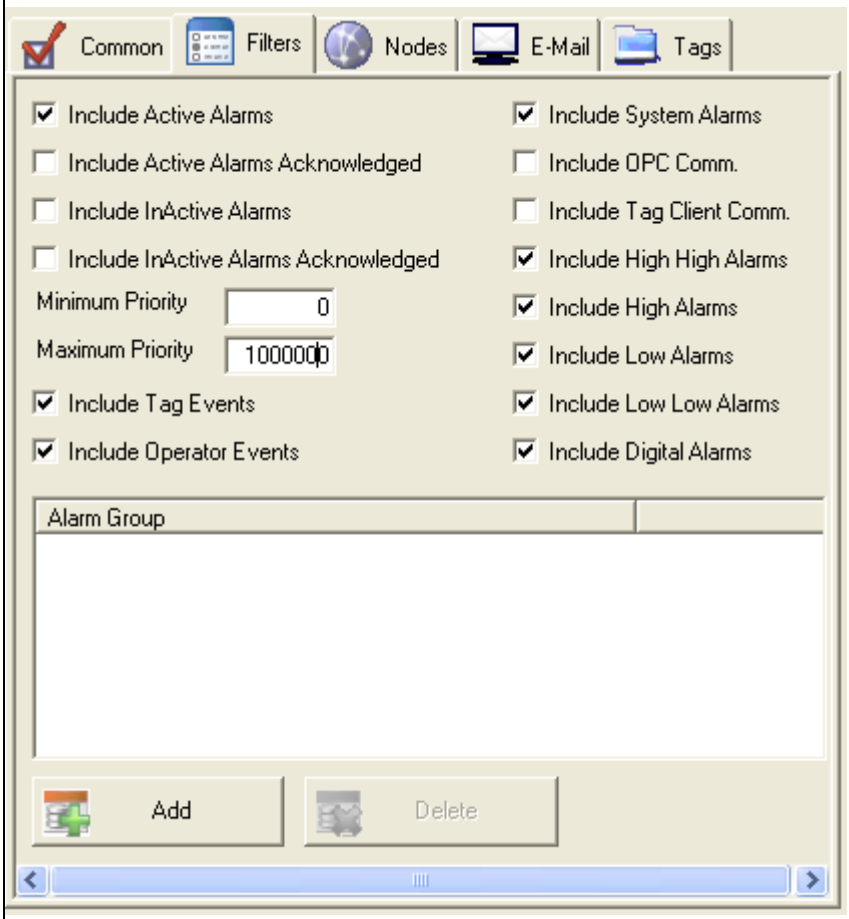
	Use the following configuration for CSV text logging. **Note:** There is no historical replay to alarm windows with CSV logging. 
8	Select the Add button to add the Alarm Logging group. 

9	Select the Save button on the toolbar at the top. 
10	Save the file DemoAlarmLogging.AlarmLog in the directory C:\OPCSystemsDemo.  You can specify for this configuration to load automatically when the Service starts using Configure-Options which is described also in this Training Guide.

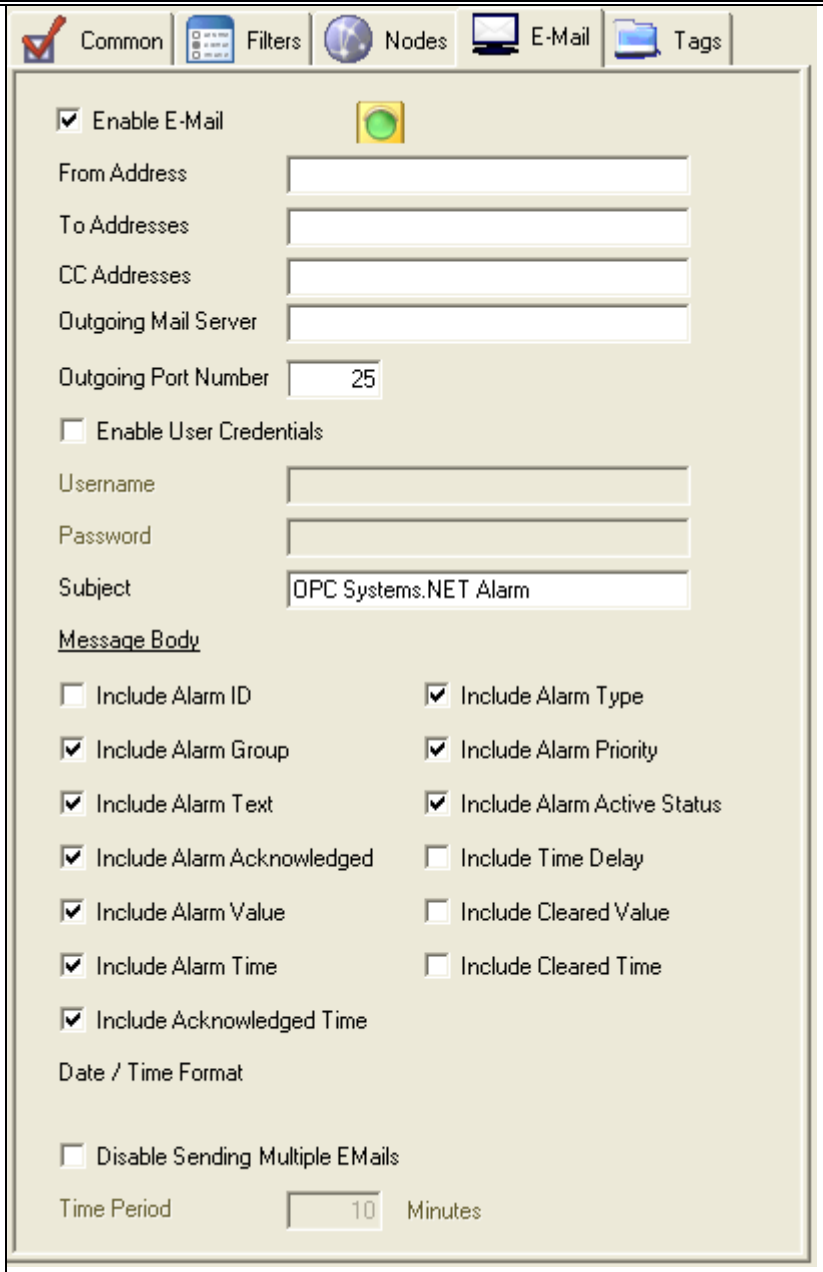
Configure Alarm Notification

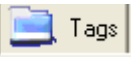
The Alarm Notification feature is used to summarize alarm totals and send e-mails based on specified filter criteria of alarm conditions.

Step	Task
1	Start Configure OPC Systems application.  Load the DemoTags Tag configuration or use the new Tag configuration you have created.
2	Select Configure-Alarm Notification. 
3	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left. 

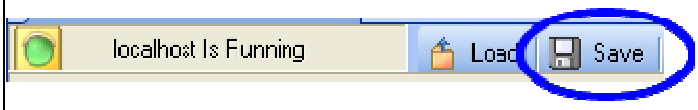
4	Enter the Notifcation Group Name of Simulation in the field in the upper right. 
5	Check Notification Active in the Common Properties Tab. 
6	Select the Filters Tab.  Notice how you can define filtering based on Alarm Priority, Alarm Groups, and Alarm Types. 

7	Select the E-Mail Tab  You can define to send alarms that meet the filter criteria to an e-mail account. Use a semicolon to separate multiple e-mail addresses.



8	Select the Tags Tab  With the Tags properties you can assign to automatically write alarm statistics to OPC Systems.NET Tags. Useful to drive an external alarm horn through a PLC or display how many alarms are currently active.

	
9	Select Configure-Tags and add a Boolean Tag with the name Alarm Is Active. Add an Integer Tag with the name Number Of Alarms Active. Add an Integer Tag with the name Number Of Alarms In Last 24 Hours.
10	Return to Configure-Alarm Notification.

	Enable the property to Set Tag When Alarm Active. Assign the Tag name Alarm Is Active.Value Enable the property to Set Integer Tag with Alarm Active Count. Assign the Tag name Number Of Alarms Active. Enable the property to Set Integer Tag With Number Of Alarms In A Given Period. Assign the Tag name Number Of Alarms In Last 24 Hours.
11	Select the Add button in the lower left to add the Alarm Notification group. 
12	Select the Save button on the toolbar at the top. 
13	Save the file DemoAlarmNotification.AlarmNotification in the directory C:\OPCSystemsDemo. You can specify for this configuration to load automatically when the Service starts using Configure-Options which is described also in this Training Guide.
14	Return to Configure-Tags and select each tag and not that the current number of alarms and active state are updated automatically to the appropriate tag.

Alarming Components

There are 3 alarming components that can be used to view real-time alarms from the tag configuration.

4. OPC Alarm .NET 100% managed component for WinForm applications.

5. OPC Web Alarm.NET 100% managed ASP.NET component for Web applications.

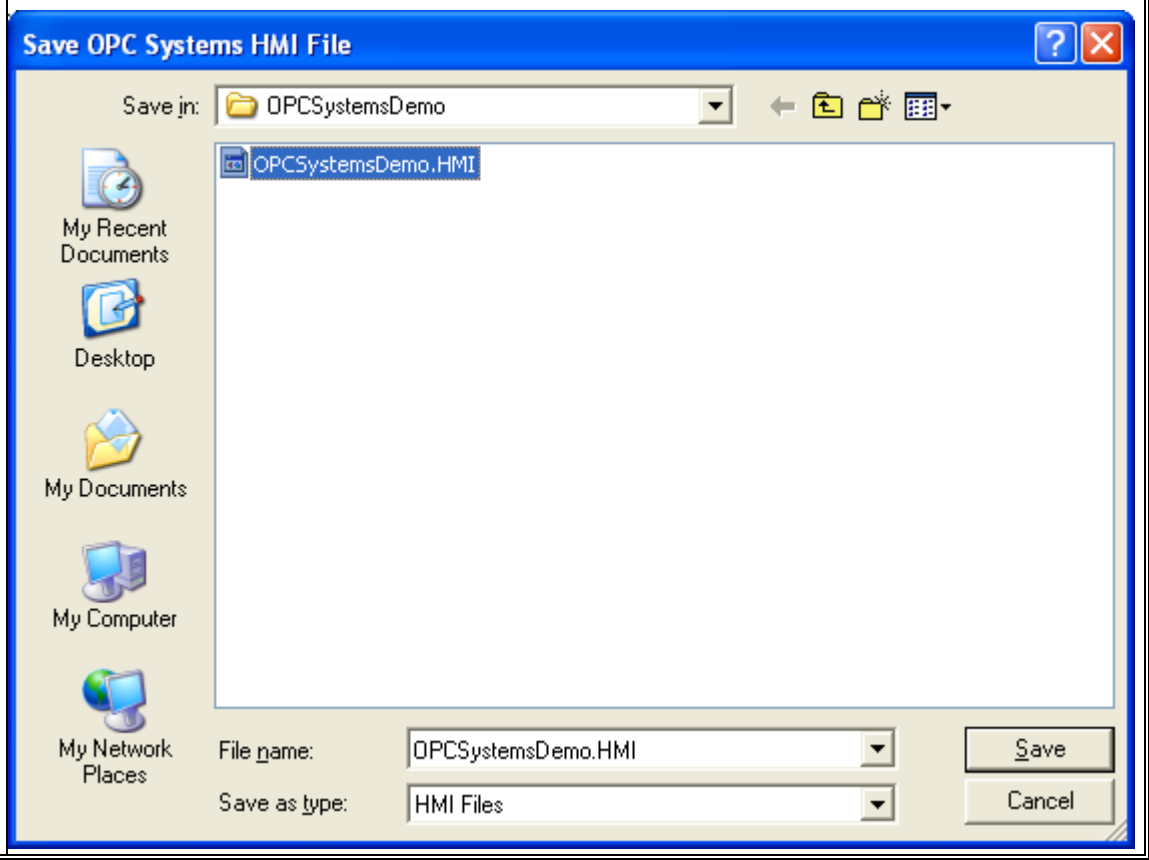
6. OPC Alarm ActiveX control for legacy applications.

All 3 types of components can be run locally or remotely for live data from any service containing an OPC Alarm.NET license or OPC Web Trend.NET license.

All 3 types of components also support historical replay from any service containing an OPC Alarm.NET or OPC Web Alarm.NET license with alarm logging groups defined as in this chapter under Configure Alarm Logging.

OPC Systems HMI

The OPC Systems HMI container allows you to use the OPC Trend .NET and OPC Alarm.NET components directly without using Visual Studio.

Step	Task
1	Start OPC Systems HMI application.
2	Select File-New and save the file OPCSystemsDemo in the C:\OPCSystemsDemo\ directory. 

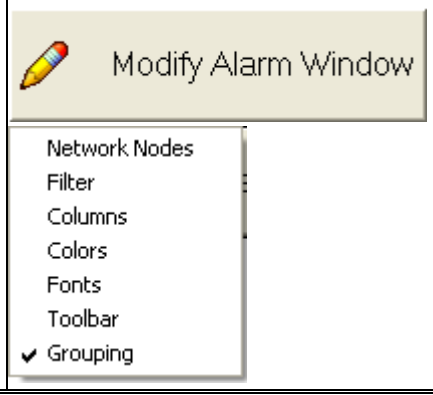
Configure Alarm Window

The OPC Alarm.NET component can be integrated directly into any .NET container, most commonly Visual Studio.NET. or OPC Systems HMI.

The OPC Web Alarm.NET component can be integrated directly into any ASP.NET Web application.

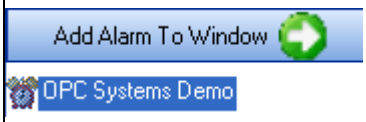
The OPC Alarm ActiveX control can be integrated directly into any ActiveX container, most commonly HMI graphic applications or Visual Studio 6.0. Refer to the Configure Alarm ActiveX section of Quick Start section of the OPC Systems.NET help file for an example. Keep in mind that ActiveX controls are not 100% managed.

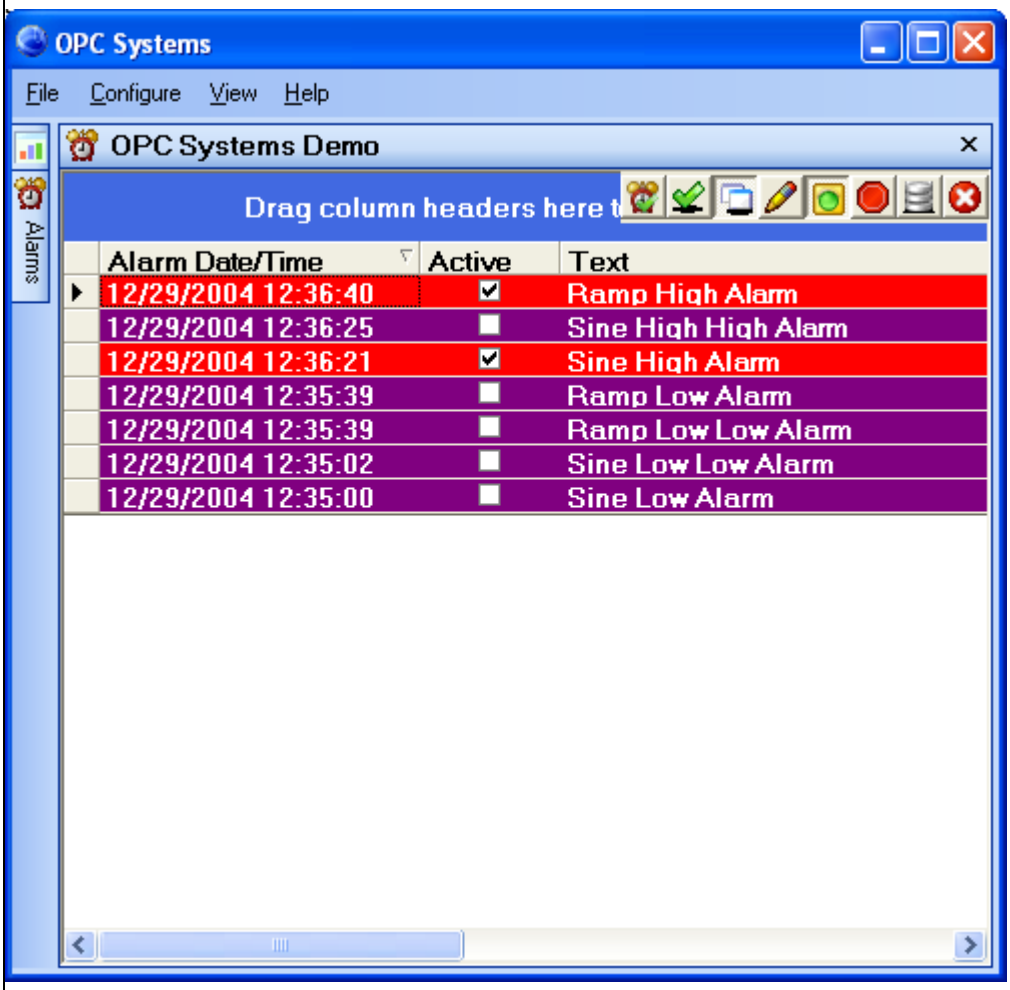
Below is an example of using the OPC Alarm.NET component integrated in the OPC Systems HMI .NET container.

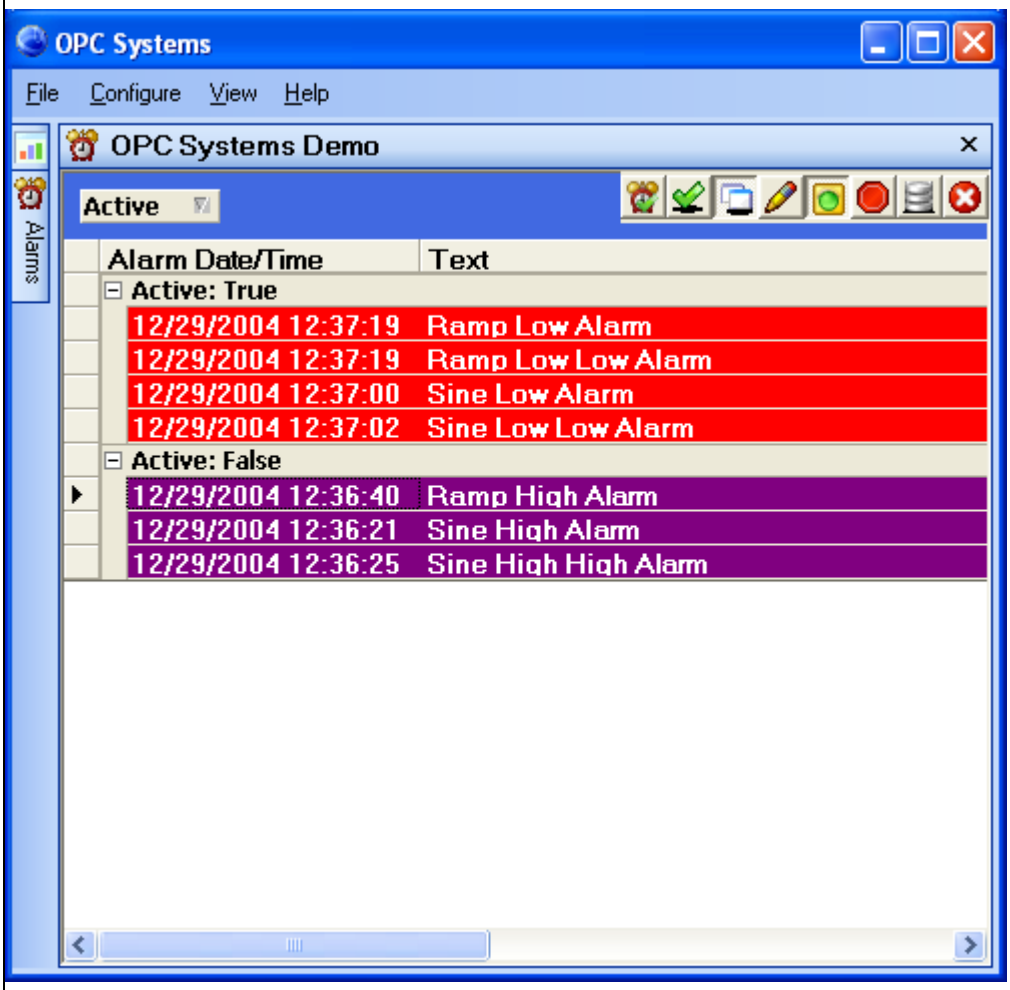
Step	Task
1	Start OPC Systems HMI application if it is not already running and select the Alarms button in the lower left of the Navigator Bar. 
2	Select the Add button under the Alarms Navigator bar on the left. 
3	Change the Alarm Window Name to OPC Systems Demo.
4	Select the Modify Alarm Window button to see that you can add other OPC Systems Nodes, modify Filters, Colors, or other attributes. 

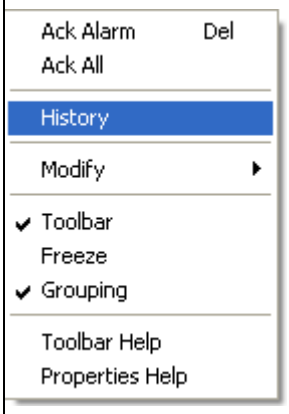
	By default the localhost node is selected and Filters include all alarms except alarms that have been acknowledged and no longer active, OPC communication failures, and Tag Client communications to other OPC Systems Services. You can adjust the Filter as you desire for Priority, Alarm Groups, and Alarm Types. *Note: if you desire to setup the Alarm Window to work on remote application select the Network Nodes to select the appropriate network names of the local or remote OPC Systems Services.
5	Click OK in the lower left corner. 
6	Select File Save  This is only necessary when you Add or Delete a Trend Window, Alarm Window, or View Window. Modifications to a Trend or Alarm Window will automatically be saved in its own configuration file.
7	Your Alarm Window is now ready to Run Alarm Window in OPC Systems HMI.

Running the Alarm Window

Step	Task
1	Start OPC Systems HMI application if it is not already running.
2	Select the Alarms button in the lower left of the Navigator bar and select the OPC Systems Demo Alarm Window and select the Add Alarm To Window button. 
3	If alarm data does not appear within a few seconds make sure to open up the Firewall for port 58723. Then restart the OPC Systems Service and try again. Contact Open Automation Software for OPC setup for XP document and help setting up OPC Systems.NET. www.opcsystems.com.

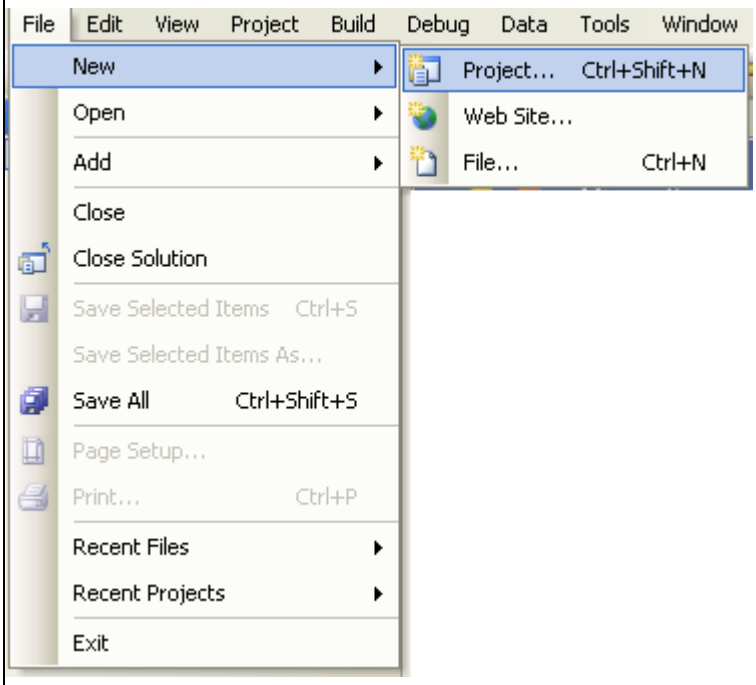
4	The Alarm Window should appear as below with real time update from the OPC Systems Service. 

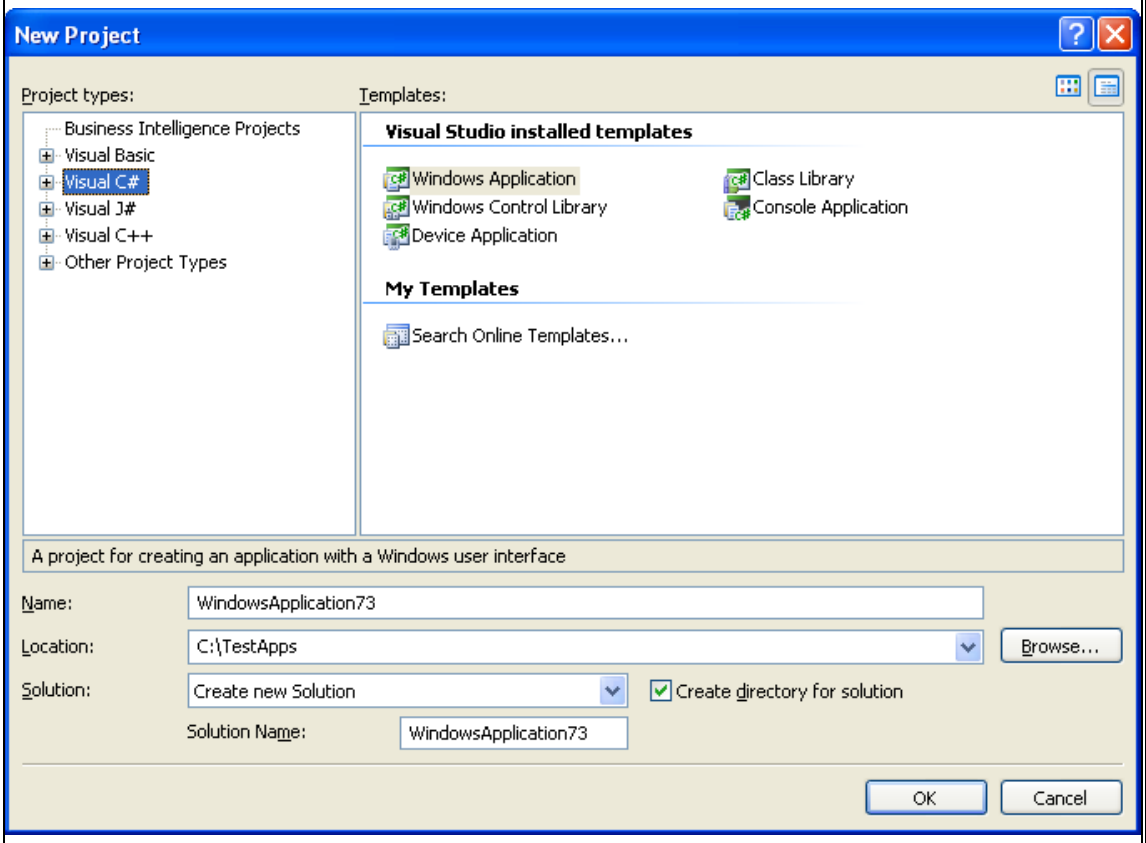
5	Select the Active Column and drag it to the header to group alarms as Active and Non-Active. 
6	Double Click on an Alarm Record to acknowledge. With the current filtering once an alarm is no longer active and acknowledged it will disappear from the Alarm Window. Use Ack All button to Acknowledge all alarms in the Alarm Window.
7	 Select File Save if you wish to save the on-line Window name and arrangement retentive. All modifications to the Trend Window properties will be retentive after this even if you do not select Save again. Only when a Window is added or removed do you need to select Save.

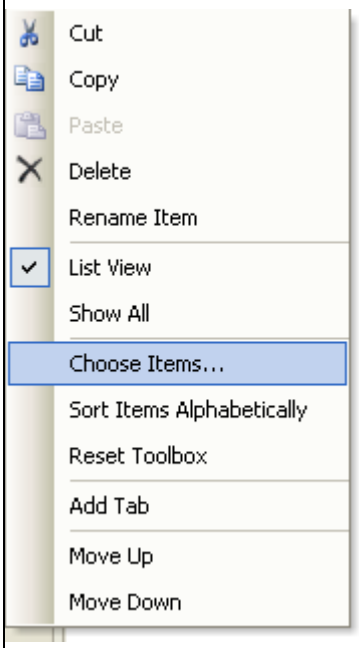
8	If you setup Alarm Logging in this example right click on the Alarm Window and select History to retrieve historical alarms. 
9	Explore the Alarm Window Toolbar buttons and right click on the Alarm Window to explore features. 

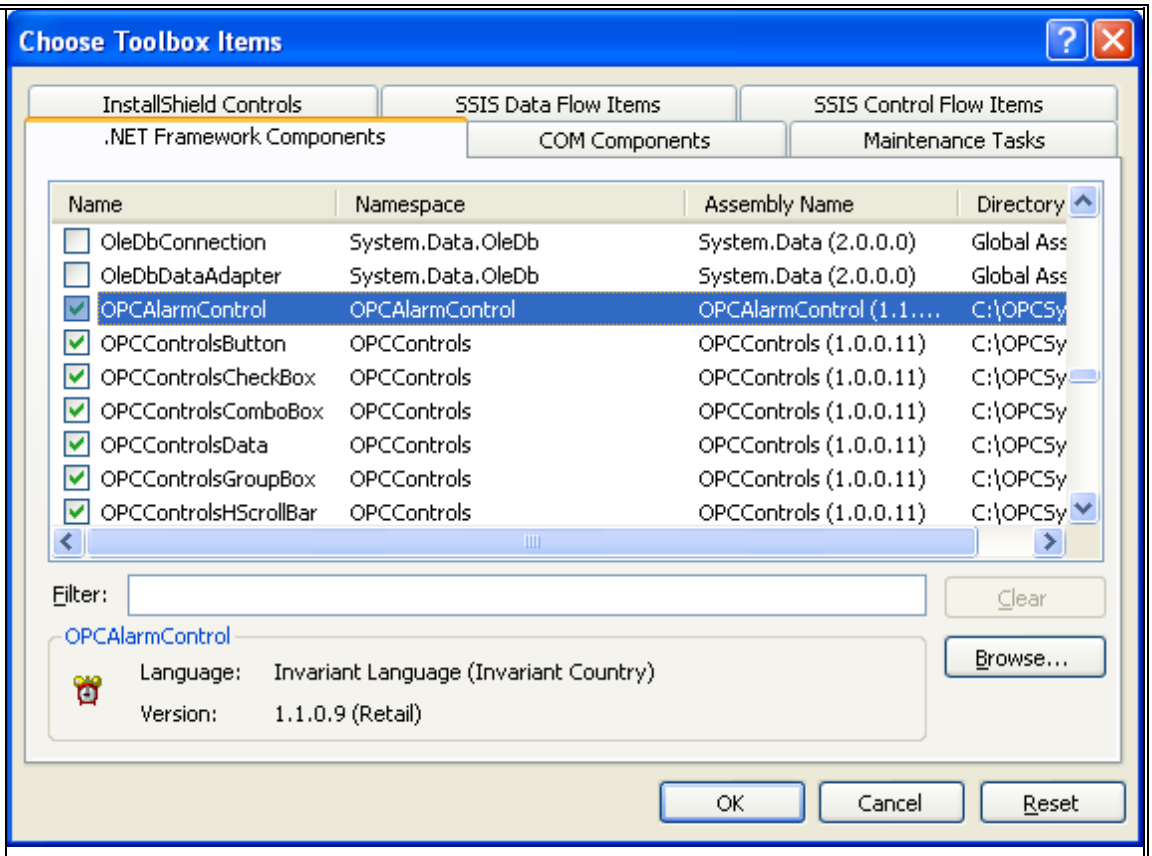
Add Alarm Window to Visual Studio Application

The following steps can be used to add an alarm window to a C#, C++, or Visual Basic.NET application. All properties are programmatically accessible. The following example demonstrates the alarm window with no code required. The alarm window can also be used with Visual Studio 2003 with the older .NET Framework 1.1 version of OPC Systems.NET.

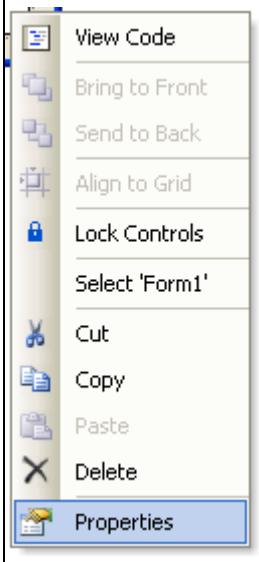
Step	Task
1	Start Visual Studio 2005 or Visual Studio 2008 and select File->New->Project to create a new C#, C++, or VB. 

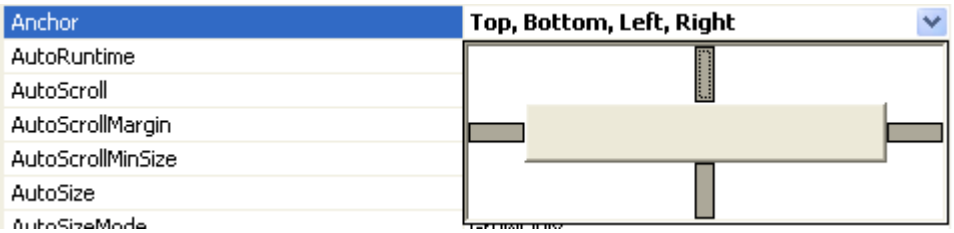
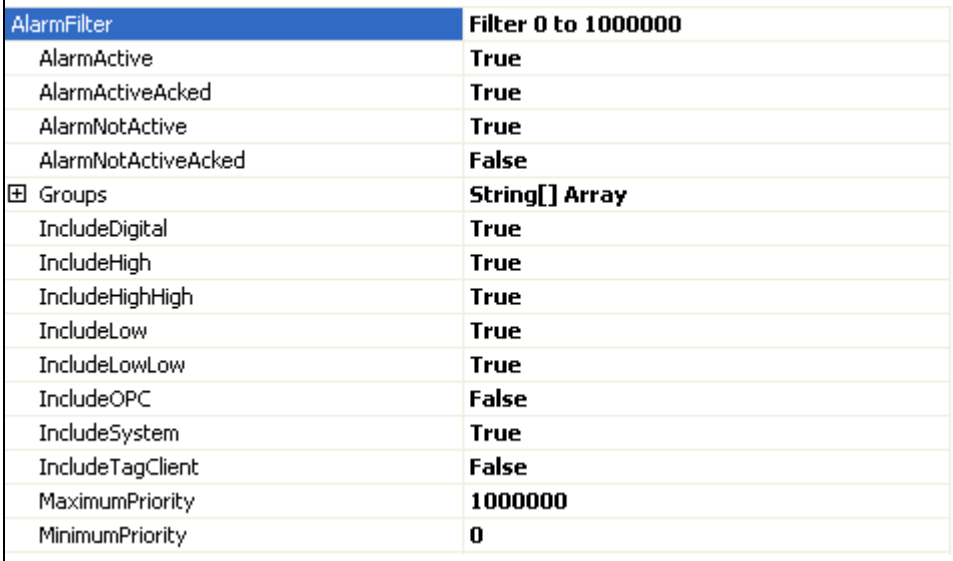
2	Select Windows Application as the project type. 

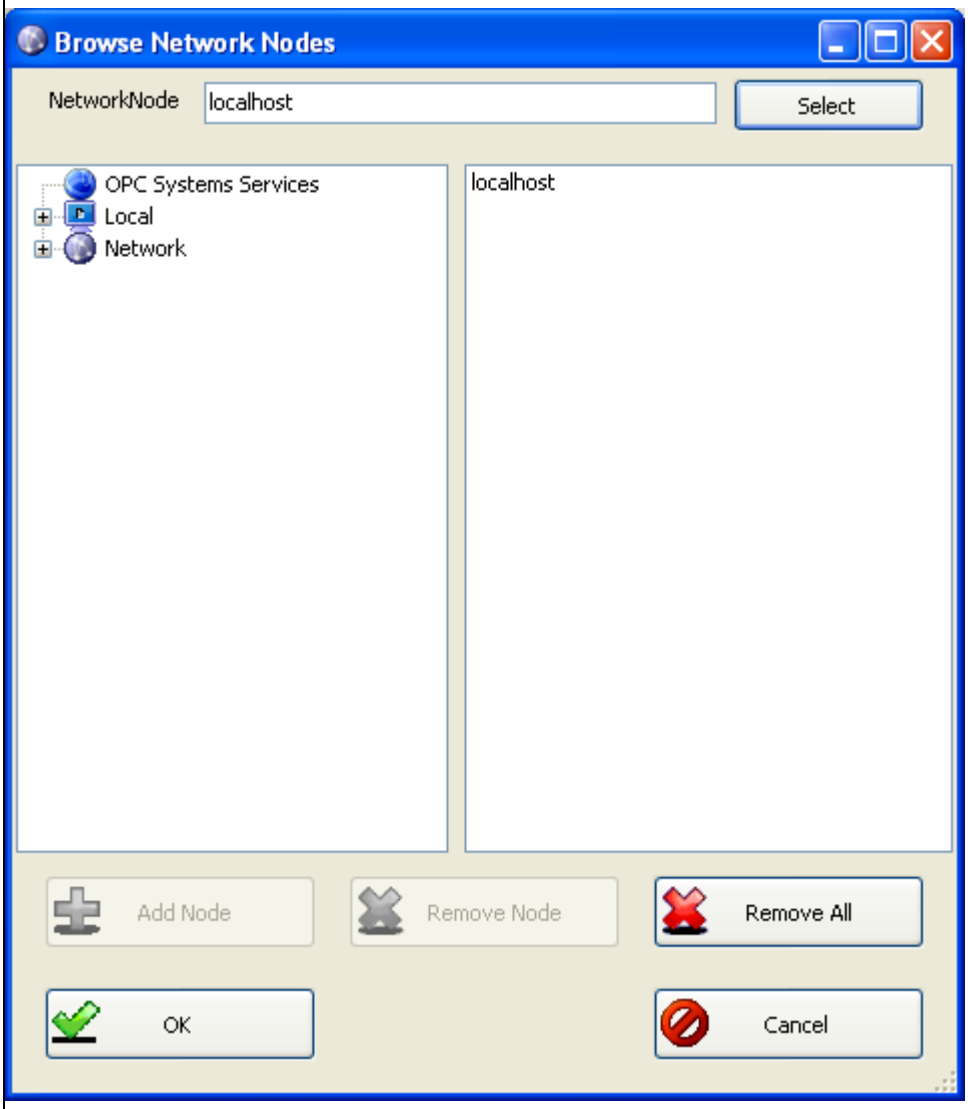
3	From the Toolbox if OPCAlarmControl is not available right click in the Toolbox and select Choose Items. If it is available to step 4.  From the .NET Framework Components select OPCAlarmControl and then select OK.

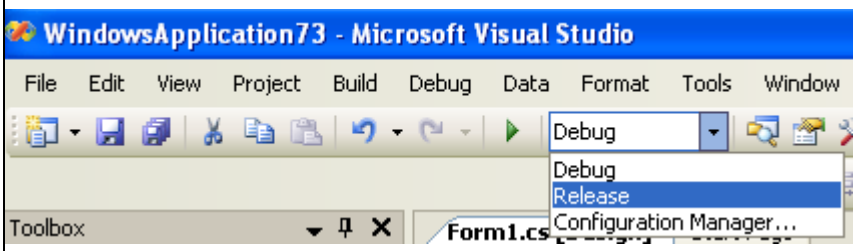
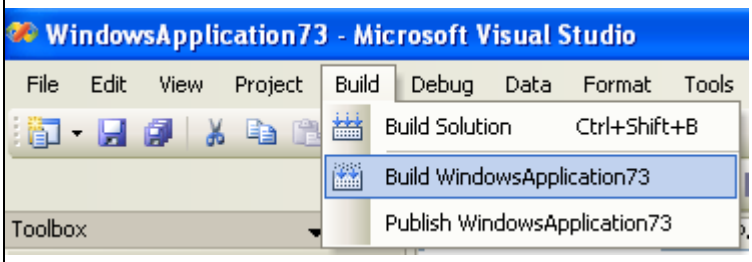


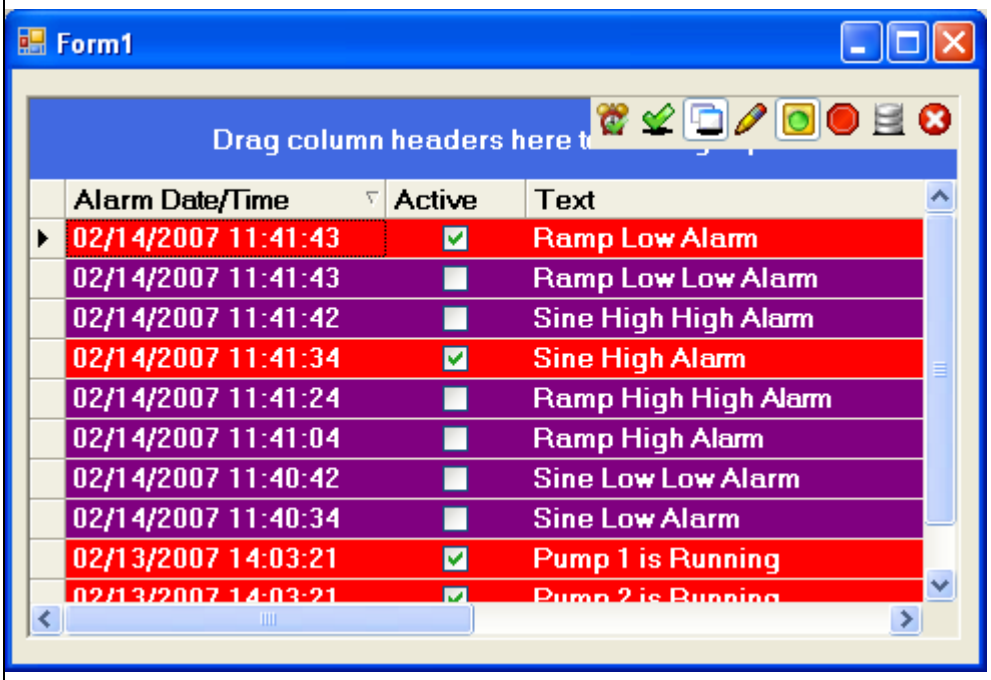
4	Add the OPCAlarmControl component onto the Form.  Resize both the form and alarm window to the desired size.

	
5	Right click on the alarm window and select Properties. 

6	Set the Anchor property to Top, Bottom, Left, Right. 
7	Expand the AlarmFilter property and set the desired filter settings for the alarm window. 
8	Select the AlarmNetworkNodes property and click on the small grey square with the 3 dots at the right. 

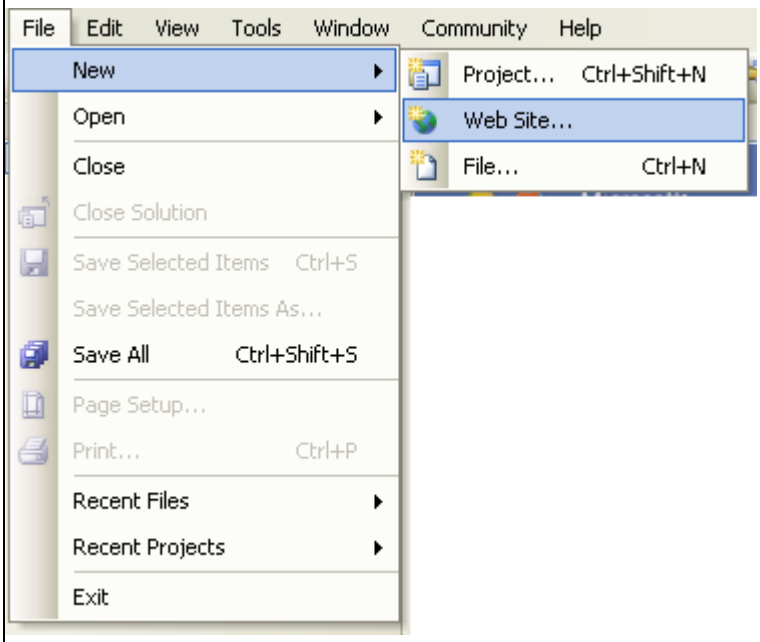
9	Select all network nodes you wish to receive alarms from to this alarm window.  **Note:** If you want the application to be deployed across a network to remote PCs select the Network Node or enter an IP Address in the NetworkNode field and use the Select button to include the network node or IP Address of the OPC Systems Service source. Refer to the VB.NET Example on how to programmatically assign the Alarm Network Nodes.

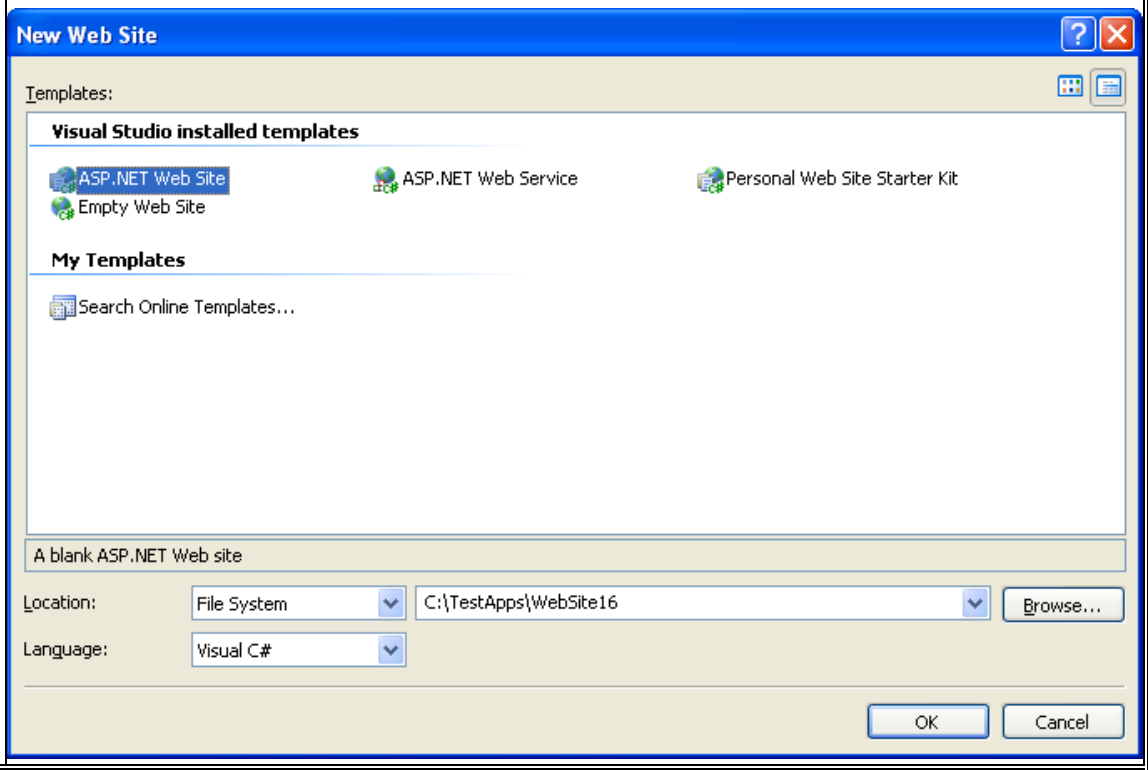
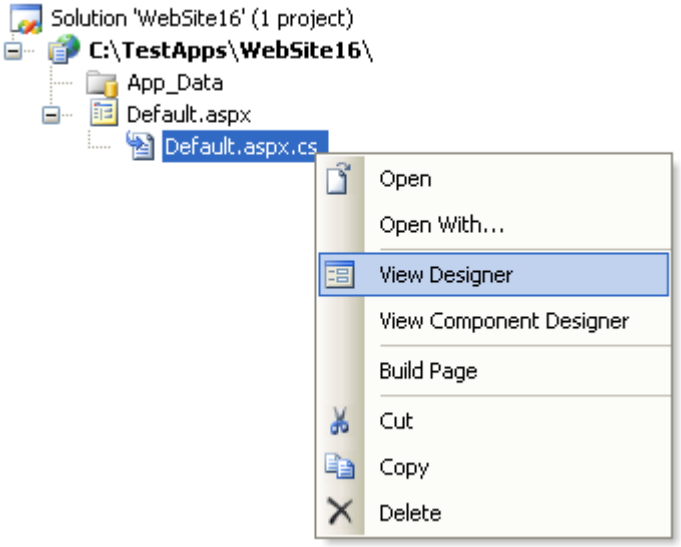
10	If you desire for the operators changes to the alarm window during runtime to remain set the ConfgiurationFile property to a valid file path. Make sure each system the application will run on that the directory path is valid.  **Note:** If you set this property to a file make sure you deploy the file with the application in the directory you specify.. Leave this property blank if you wish to have the default properties set during configuration remain on the application restarting.
11	Set the compile mode on the Visual Studio toolbar to Release. 
12	Select Build from the VS menu and select to Build the application. 

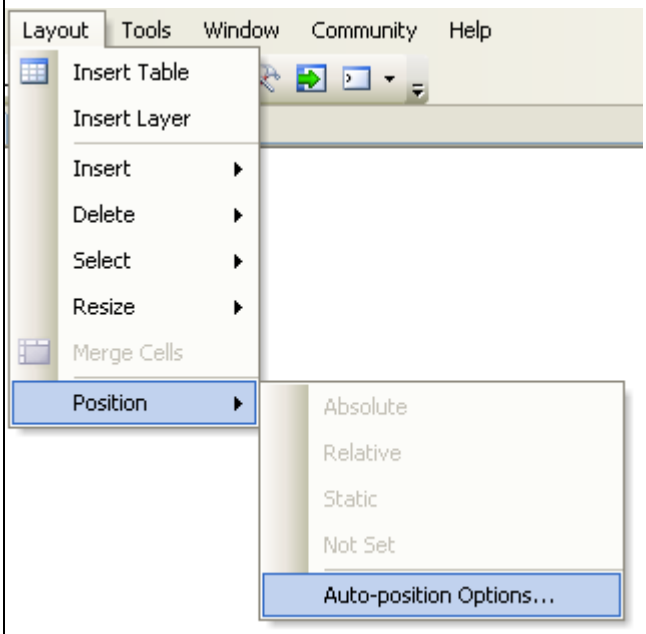
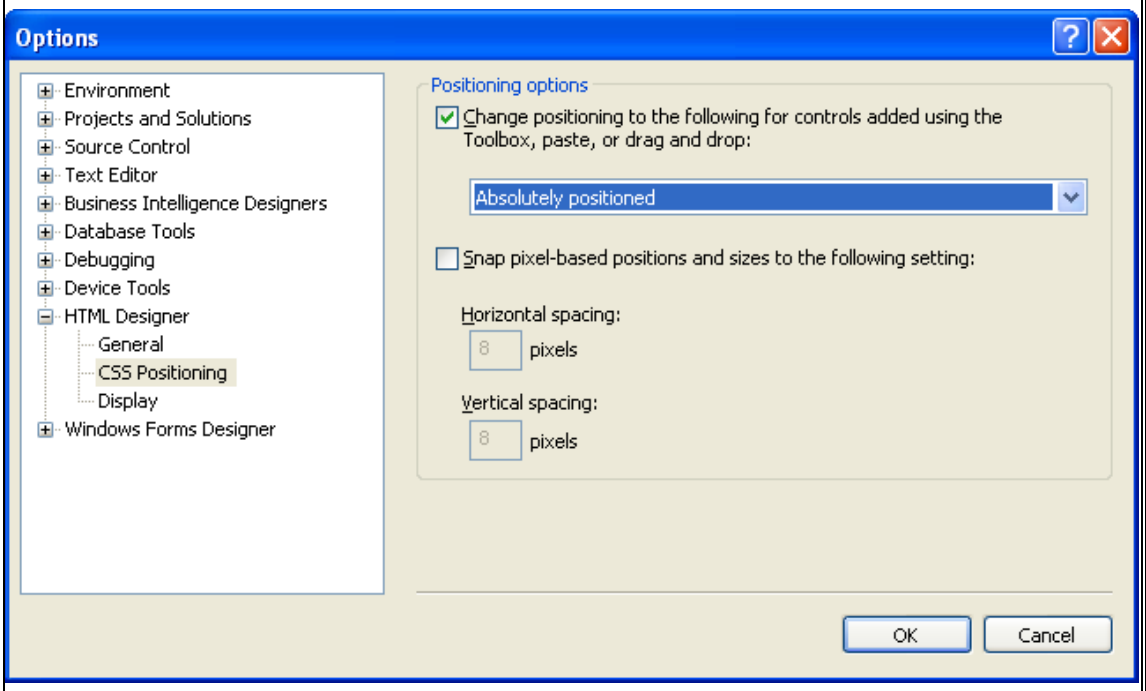
13	Use Windows Explorer to browse for the application located in the bin\Release directory and run the application. 
14	To deploy the application to remote nodes first make sure the AlarmNetworkNodes selection as described in step 9 is set to a Network Node or IP Address. Then simply copy the files in the bin\Release directory to the target systems or follow the Smart Client deployment section in this help file to deploy your application using Click Once Deployment.

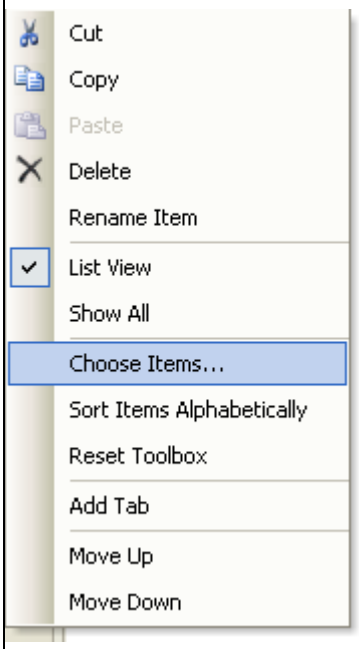
Add Alarm Window to ASP.NET Web Application

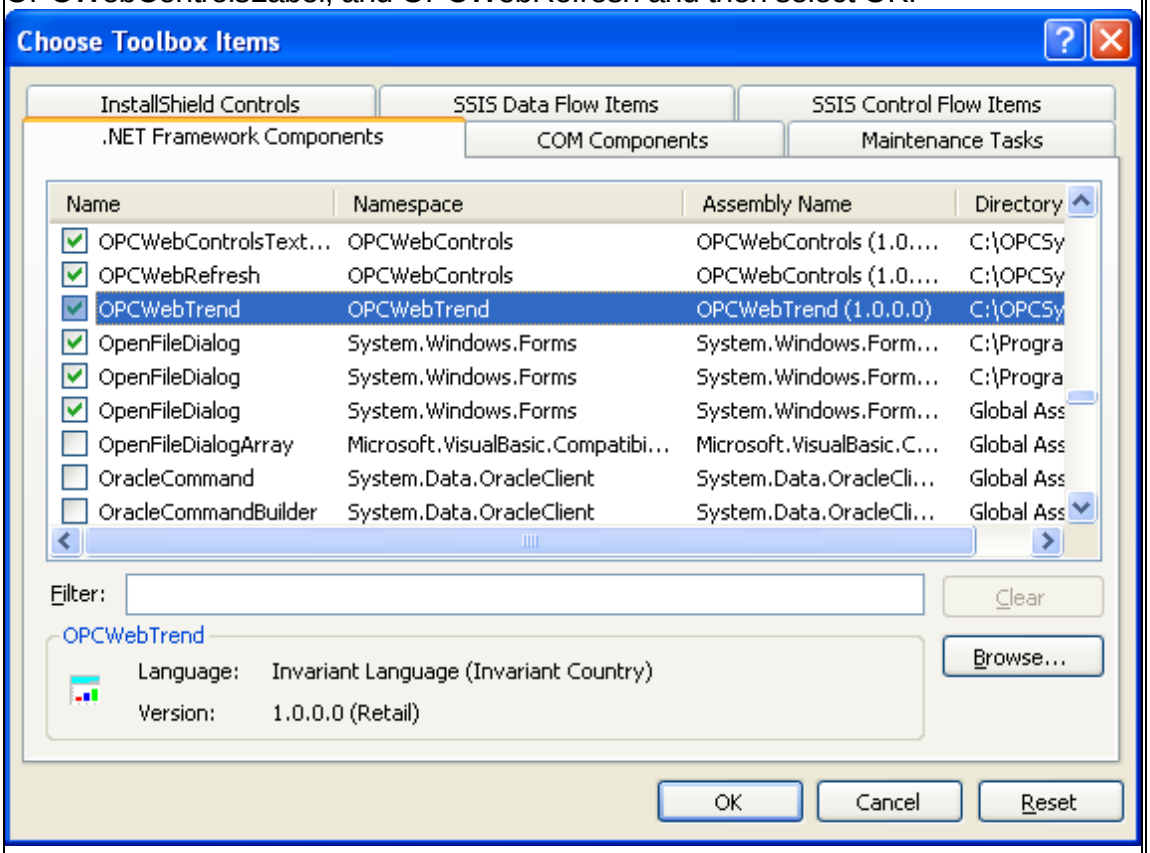
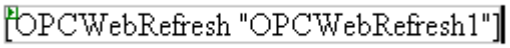
The following steps can be used to add an alarm window to a C#, J#, or Visual Basic.NET web application. All properties are programmatically accessible. The following example demonstrates the trend window with no code required. The trend window can also be used with Visual Studio 2003 web applications if you use the older OPC Systems.NET 1.1 Framework version.

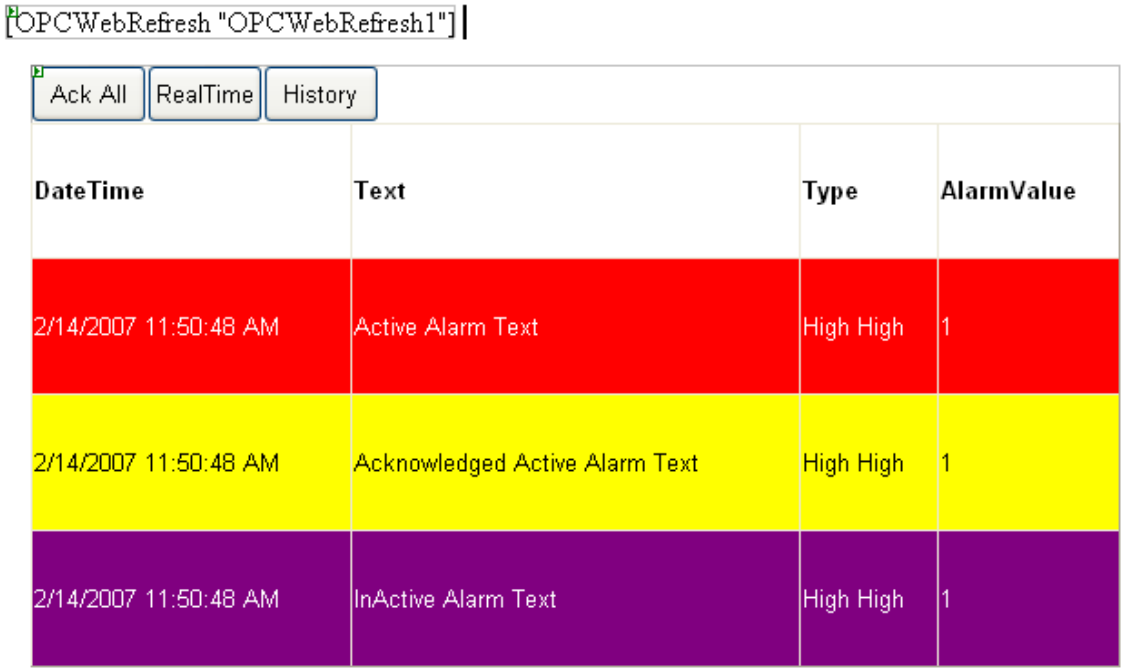
Step	Task
1	Start Visual Studio 2005 or Visual Studio 2008 and select File->New->Web Site to create a new C#, J#, or VB ASP.NET web application. 

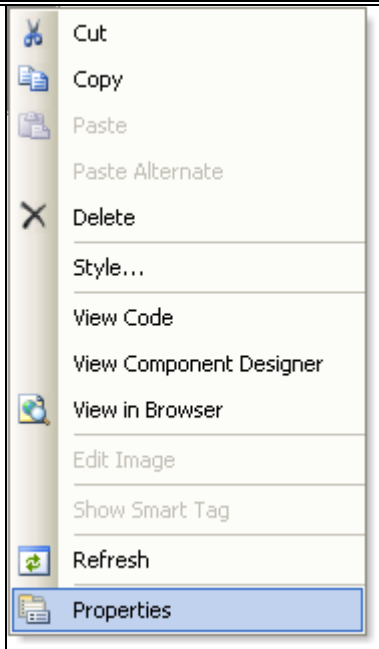
2	Select ASP.NET Web Site and specify your desired Language in the lower right. 
3	Expand the Default.aspx web page in the Solution Explorer and select View Designer. **Note:** If you do not see the Solution Explorer select View->Solution Explorer. 

4	From the VS menu select Layout->Position->Auto-position Options. 
5	Under the HTML Designer set the CSS Positioning to Absolutely positioned. **Note:** You are welcome to use any positioning style you desire, this simply pointed out for new web developers. 

6	From the Toolbox if OPCWebAlarm, OPCWebControlsLabel, and OPCWebRefresh is not available right click in the Toolbox and select Choose Items. If it is available to step 4. 

	From the .NET Framework Components select OPCWebAlarm, OPCWebControlsLabel, and OPCWebRefresh and then select OK. 
7	Add one OPCWebRefresh control to the WebForm.   Add one OPCWebControlsLabel control to the WebForm and then delete it. This is the easiest way to ensure that the web.config file is update for proper AJAX registration.

8	Add the OPCWebAlarm component onto the WebForm.  Resize the alarm window to the desired size. 
9	Right click on the alarm window and select Properties.

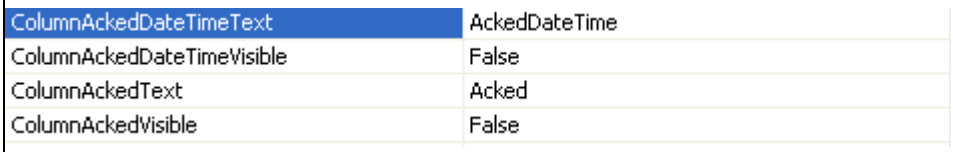
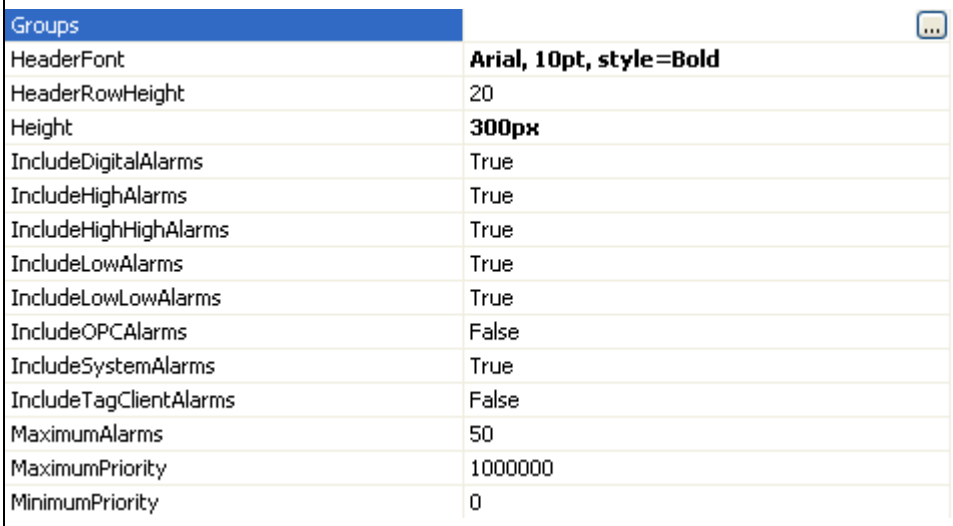
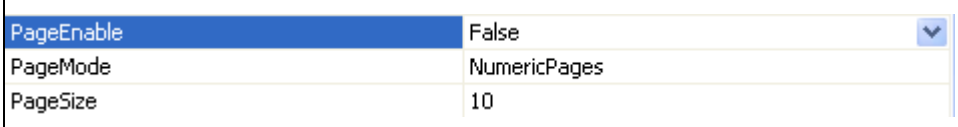


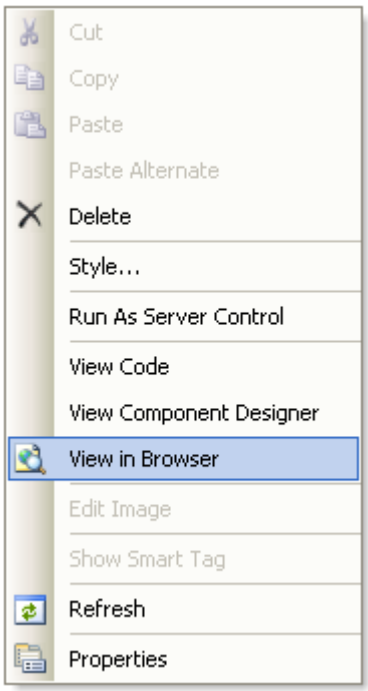
10 Specify the alarm types you desire to view in the alarm window.

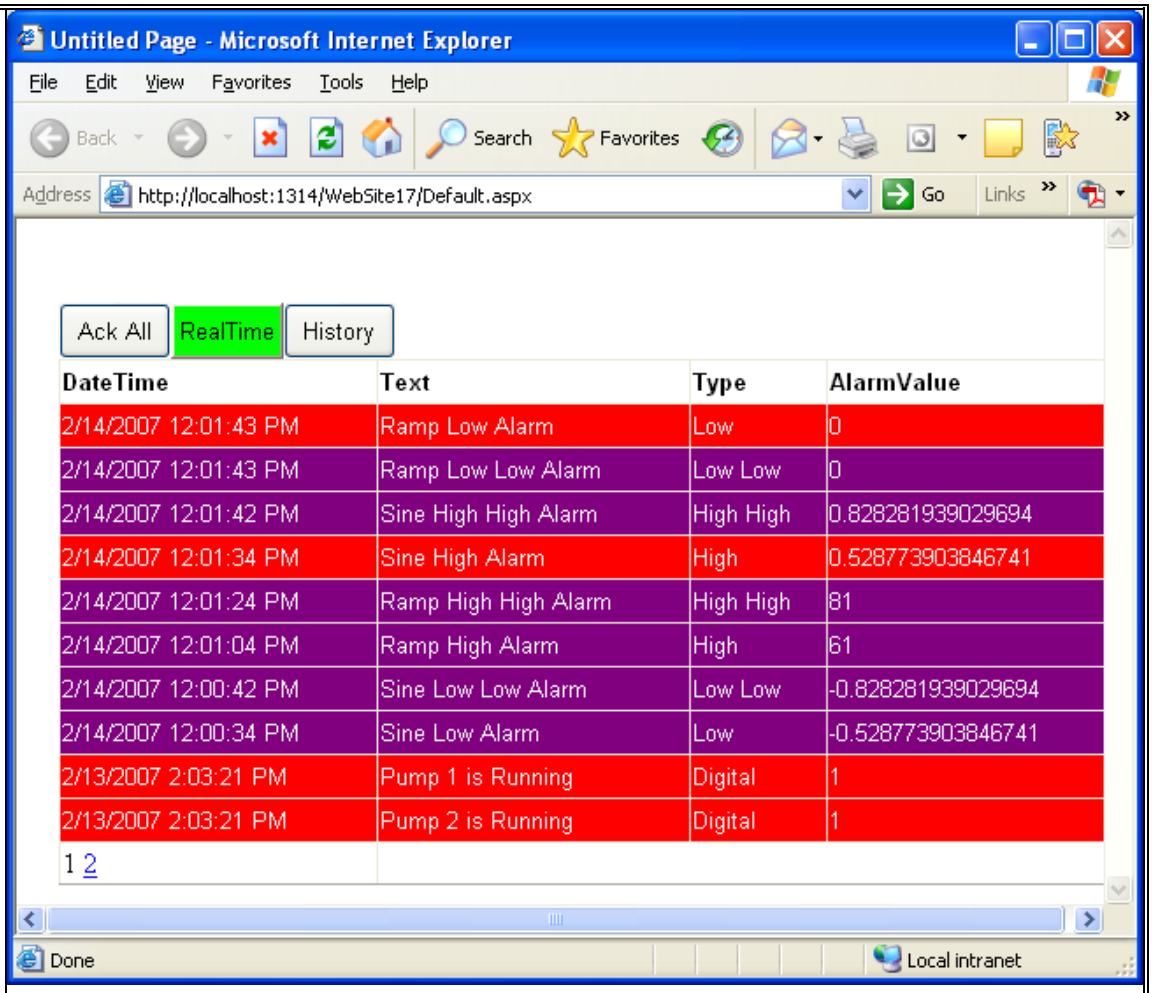
AlarmActive	True
AlarmActiveAcked	True
AlarmActiveAckedBackColor	Yellow
AlarmActiveAckedFont	**Arial, 10pt**
AlarmActiveAckedForeColor	Black
AlarmActiveBackColor	Red
AlarmActiveFont	**Arial, 10pt**
AlarmActiveForeColor	White
AlarmNetworkNodes	localhost
AlarmNotActive	True
AlarmNotActiveAcked	False
AlarmNotActiveAckedBackColor	Lime
AlarmNotActiveAckedFont	**Arial, 10pt**
AlarmNotActiveAckedForeColor	Black
AlarmNotActiveBackColor	Purple
AlarmNotActiveFont	**Arial, 10pt**
AlarmNotActiveForeColor	White

11 Use the AlarmNetworkNodes property to specify all OPC Systems Services you wish to receive alarms from.

AlarmNetworkNodes	localhost

12	Specify what alarm columns you wish to see in the alarm window. 
13	Use the Groups, Include, Max and Min Priority properties to filter what types of alarms will be viewed in the alarm window. 
14	Set the PageEnable property to True if you desire to limit the size of the alarm window. 

15	Right click on the WebForm and select View in Browser. 



19	Set the application for Release mode, modify the web.config file to set the debug property to false, and build your application. Typically the application files are under the Inetpub\wwwroot directory. Refer to the OPC Web Alarm.NET section in the OPC Systems.NET help file for Deployment steps if you encounter any issues.

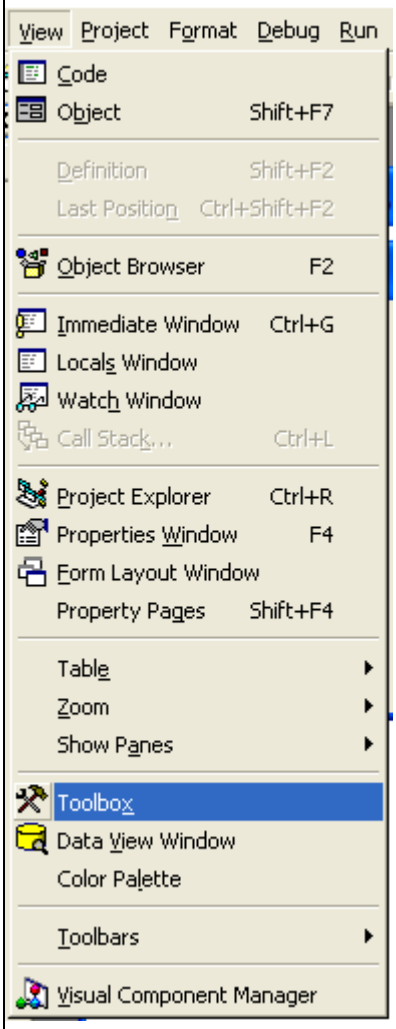
Add Alarm ActiveX to Legacy Application

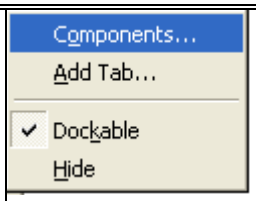
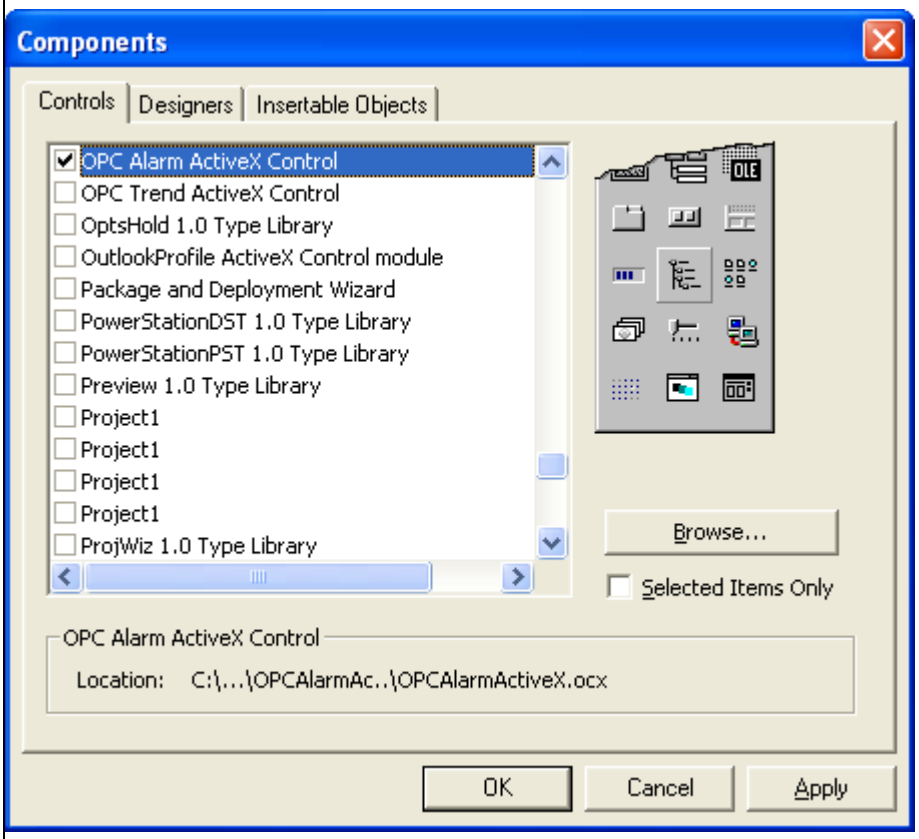
The OPC Alarm ActiveX control can be integrated directly into any ActiveX container, most commonly HMI graphic applications or Visual Studio 6.0.

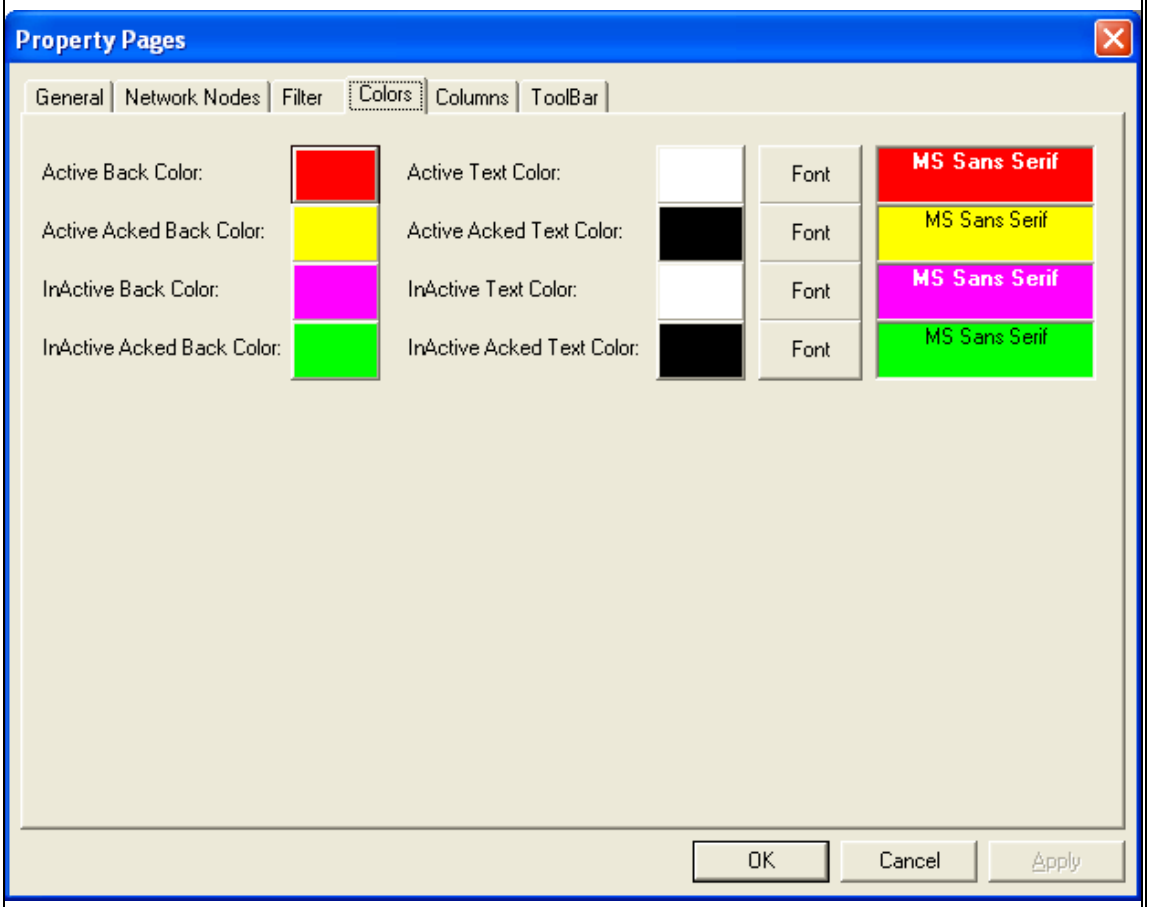
Note: ActiveX controls are not 100% managed and you should use the .NET controls for Visual Studio.NET applications and in more update to date containers that support .NET controls.

Below is an example of using the OPC Alarm ActiveX Control integrated into Visual Basic 6.0 or other ActiveX container of your choice.

Step	Task
1	Start Visual Basic 6.0 with a new or existing Project or other ActiveX container application of your choice. If using other ActiveX container than Visual Studio simply select the appropriate method of the application to add an ActiveX control and select OPCAlarmActiveX.OPCAlarm and skip to Step 6.

2	To add the OPC Alarm.ActiveX Control to your Visual Studio 6.0 development system select View-Toolbox from within Visual Studio. 
3	Right-Click on the Toolbox and select Components.

	
4	Select OPC Alarm ActiveX Control from the list of Controls 
5	From the ToolBox select the OPC Alarm ActiveX Control icon and add the alarm window to any Form.
6	Resize the Alarm Window to any size your desire. Please note that some of the Runtime Properties will not be available for modifying during runtime if the Alarm Window is too small.

7	Select Properties of Alarm Window and set any desired Property.  By default the localhost node is selected and Filters include all alarms except alarms that have been acknowledged and no longer active, OPC communication failures, and Tag Client communications to other OPC Systems Services. You can adjust the Filter as you desire for Priority, Alarm Groups, and Alarm Types. *Note: if you desire to setup the Alarm Window to work on remote application select the Network Nodes to select the appropriate network names of the local or remote OPC Systems Services.

8	Select OK at the bottom of the Property Page.
9	Save your application and compile if needed and you are now ready to run the application.

Chapter

6

Chapter 6 - Human Machine Interface

HMI Components

There are 3 different types of HMI components that can be used to display and change real-time data from the tag configuration.

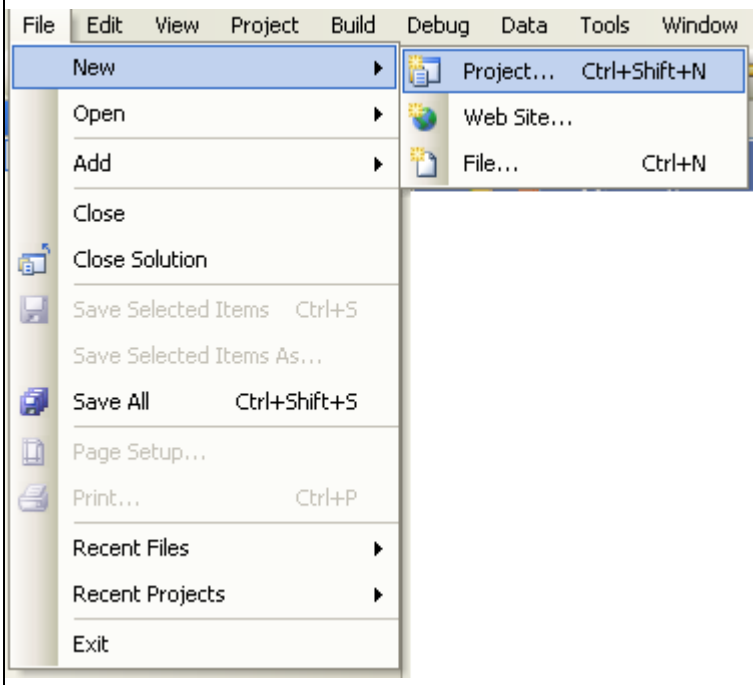
7. OPC Controls.NET 100% managed components for WinForm applications.
8. OPC Web Controls.NET 100% managed ASP.NET components for Web applications.
9. OPC Mobile.NET 100% managed components for pocket PC applications for Windows Mobile 2003 and 5.0.

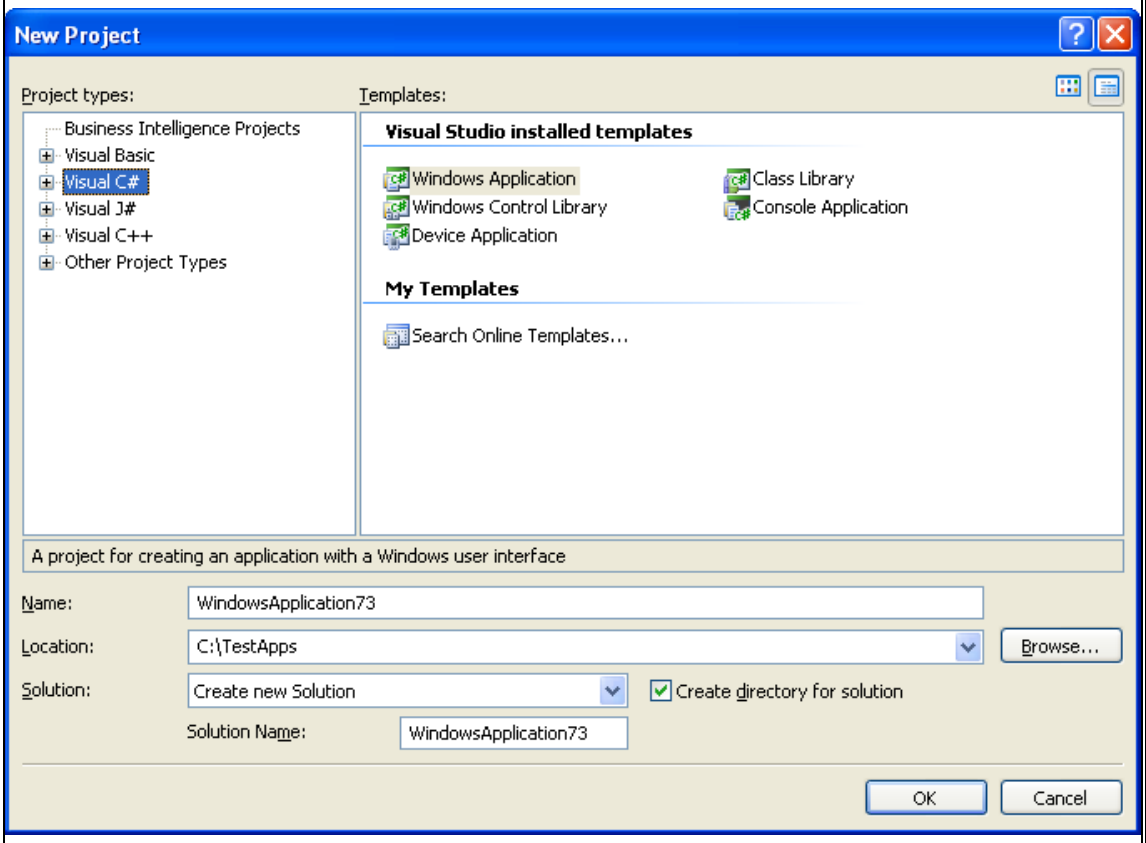
All 3 types of components can be run locally or remotely for live data from any service containing a valid license for the corresponding product.

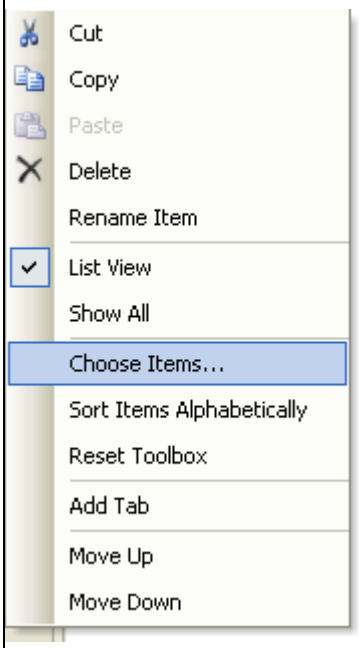
Add HMI Controls to Visual Studio Application

You can download a free version of Visual Basic Express 2008 or C# Express 2008 from www.microsoft.com.

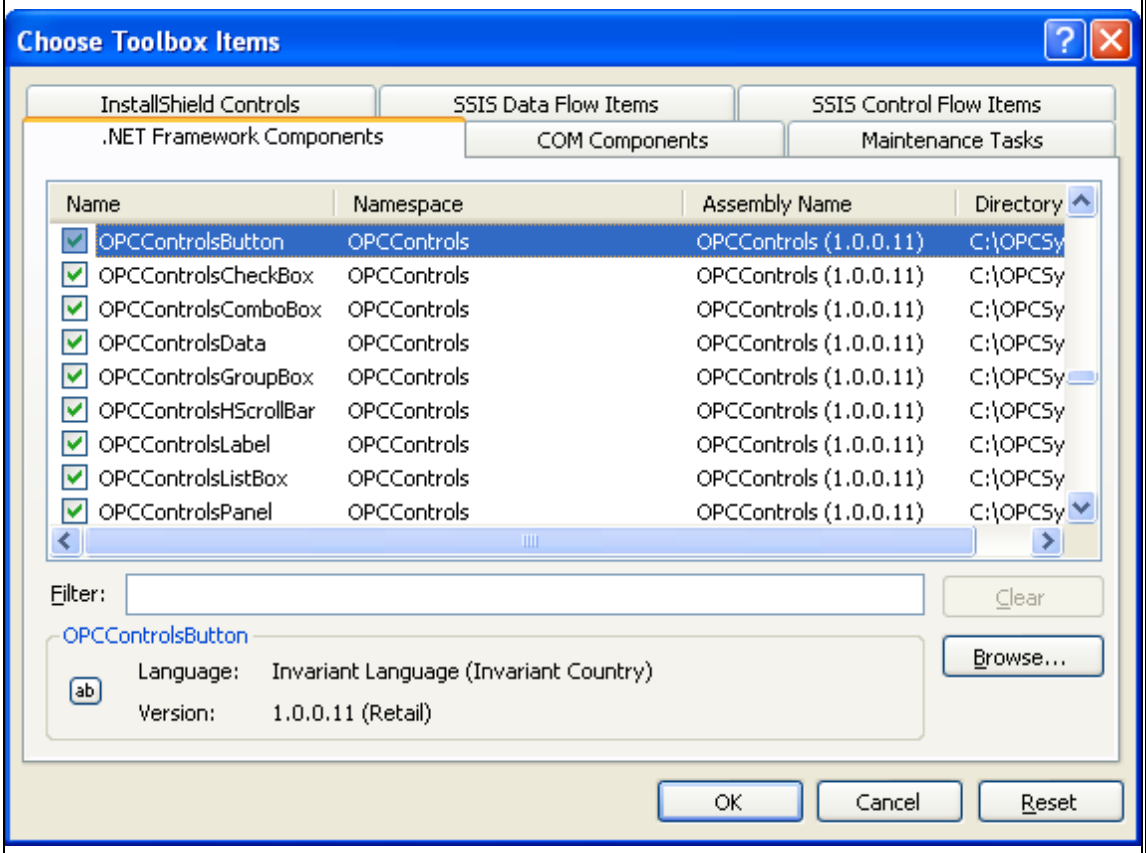
The following steps can be used to add visualization to a C#, C++, or Visual Basic.NET application. Refer to the VB.NET example for programmatic interface of using the OPC Controls components. All properties are programmatically accessible. The following example demonstrates the use of OPC Controls.NET with no code required. The components can also be used with Visual Studio 2003. but you will need to download the older 1.1 Framework version of OPC Systems.NET.

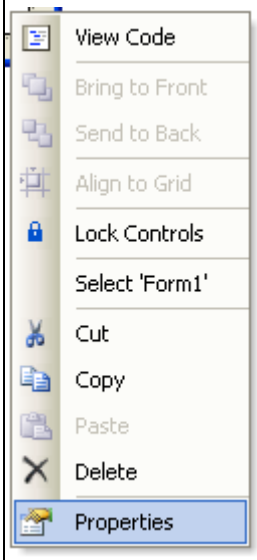
Step	Task
1	Load the default DemoTags Tag configuration if you have replaced your tag configuration with your own tags. Start Visual Studio 2005 or Visual Studio 2008 and select File->New->Project to create a new C#, C++, or VB. 

2	Select Windows Application as the project type. 

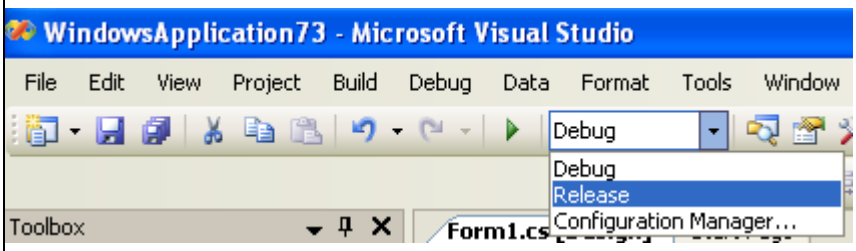
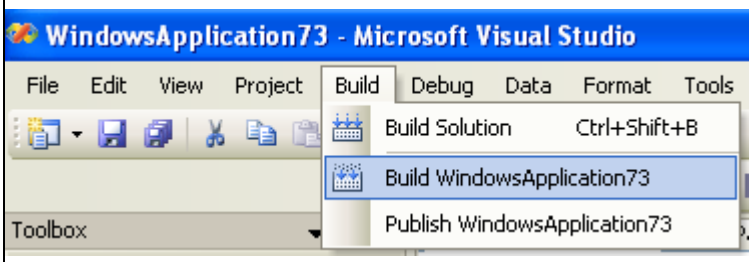
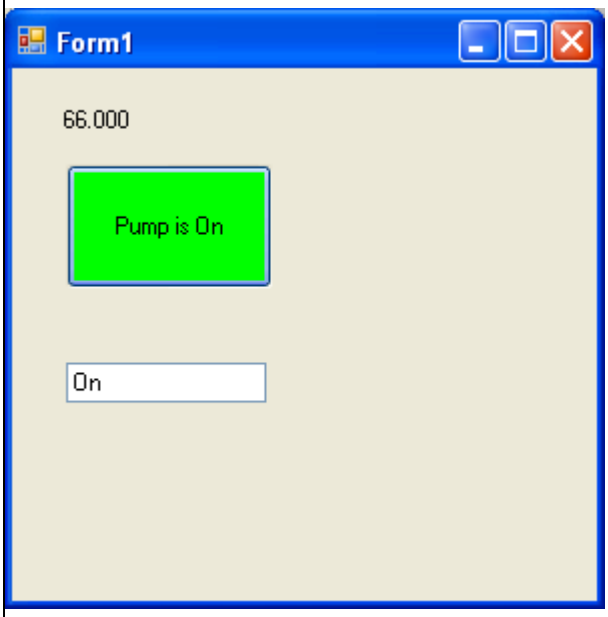
3	From the Toolbox if OPCControls components are not available right click in the Toolbox and select Choose Items. If it is available to step 4. 

From the .NET Framework Components select all of the OPC Controls components and then select OK.



4	Add an OPCControlsLabel component onto the Form.  Right click on the OPCControlsLabel window and select Properties.  Select the TextOPCSystems_Tag property and use the browse button at the right to set the OPC Systems.NET Tag to Ramp.Value.  **Note:** If you wish to run this application on remote PCs make sure to include the Network Node or IP Address of the OPC Systems Service source. **Note:** All Tag names are case sensitive. Ramp.Value is valid, ramp.value is not. **Note:** You can use the DirectOPC interface to connect directly to OPC Server Items is you just need to gain access to the server items without having to create OPC Systems.NET Tags.

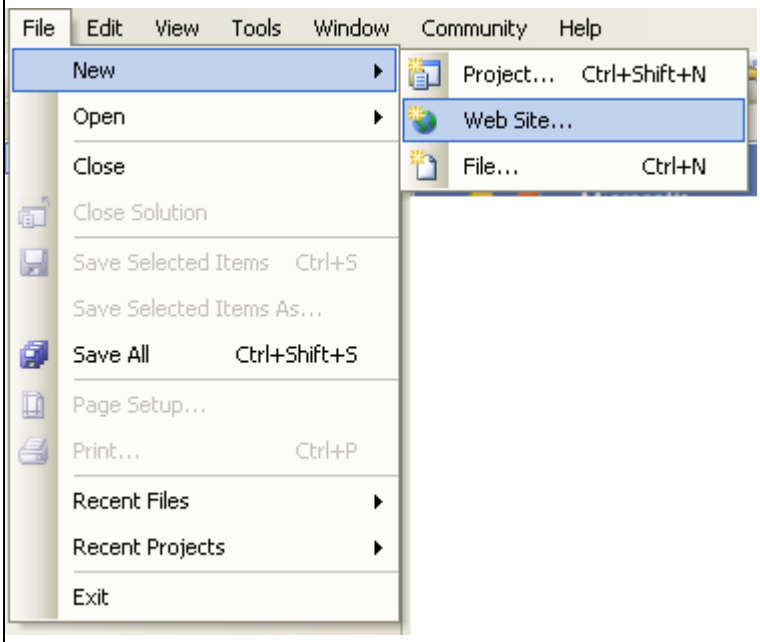
5	Add an OPCControlsButton to the Form.  Set the TextOPCSystems_Tag to Pump.Value. If the Pump Tag does not exist create a Boolean Tag using Configure-Tags with the name Pump. Set the Format fields as defined below.. TextOPCSystems_Tag: **Pump.Value** TextOPCSystemsBadQualityText: ????? TextOPCSystemsFormatBooleanFalse: **Pump is Off** TextOPCSystemsFormatBooleanTrue: **Pump is On** Set the BackColorOPCSystems_Tag to Pump.Value. BackColorOPCSystems_Tag: **Pump.Value** BackColorOPCSystemsBadQualityColor: Yellow BackColorOPCSystemsFalse: Red BackColorOPCSystemsTrue: Lime Set the SetValueOPCSystems_Tag to Pump.Value. and the SetValueOPCSystems property to True. SetValueOPCSystems: **True** SetValueOPCSystems_Tag: **Pump.Value** SetValueOPCSystemsDataType: LogicalDiscrete SetValueOPCSystemsDiscreteValue: ToggleValue
6	Add an OPCControlsTextBox to the Form.  Set the TextOPCSystems_Tag property to Pump.Value. The Format properties for Boolean to Off and On. TextOPCSystems_Tag: **Pump.Value** TextOPCSystemsBadQualityText: ????? TextOPCSystemsDisableUpdateWithFocus: True TextOPCSystemsFormatBooleanFalse: **Off** TextOPCSystemsFormatBooleanTrue: **On** TextOPCSystemsFormatFloat: 0.000 TextOPCSystemsFormatInteger: 0 TextOPCSystemsSendValueOnEnter: True

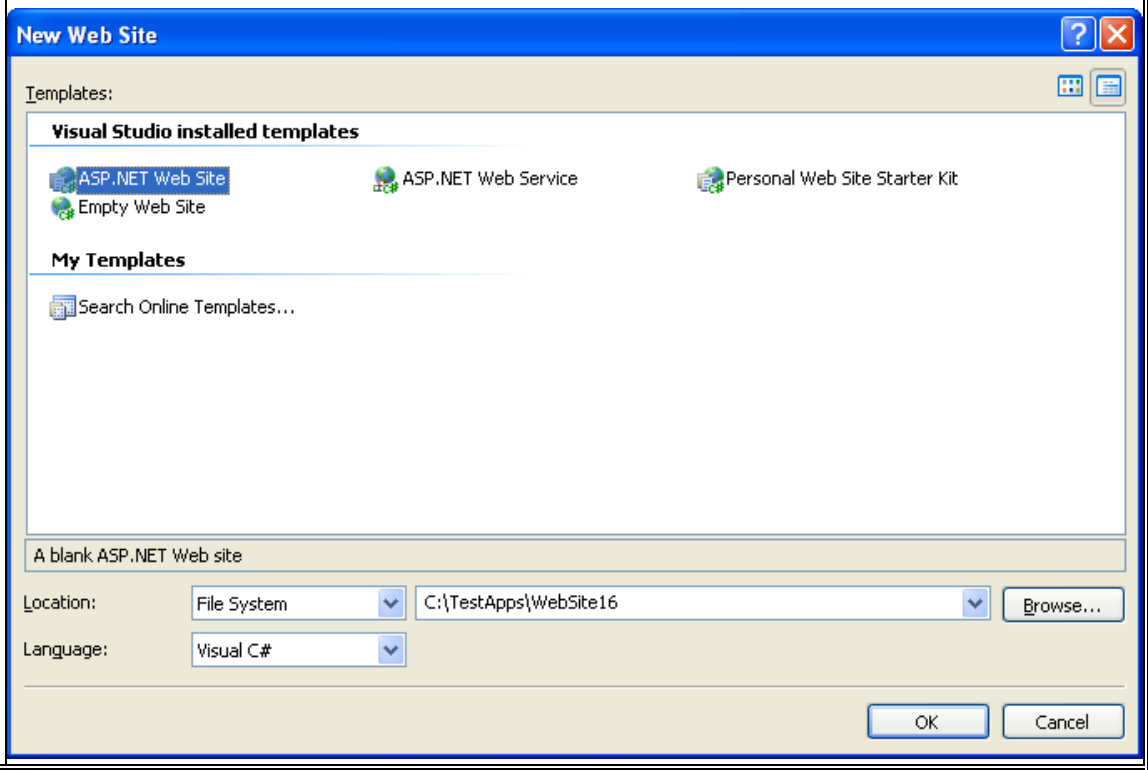
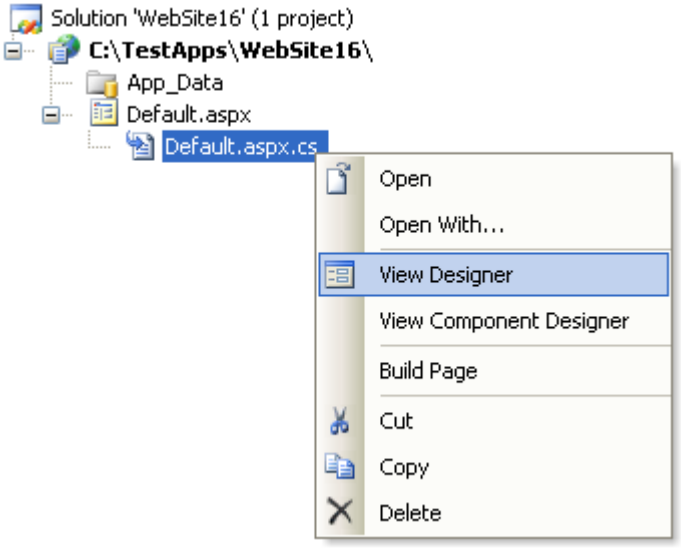
7	Set the compile mode on the Visual Studio toolbar to Release. 
8	Select Build from the VS menu and select to Build the application. 
9	Use Windows Explorer to browse for the application located in the bin\Release directory and run the application. 

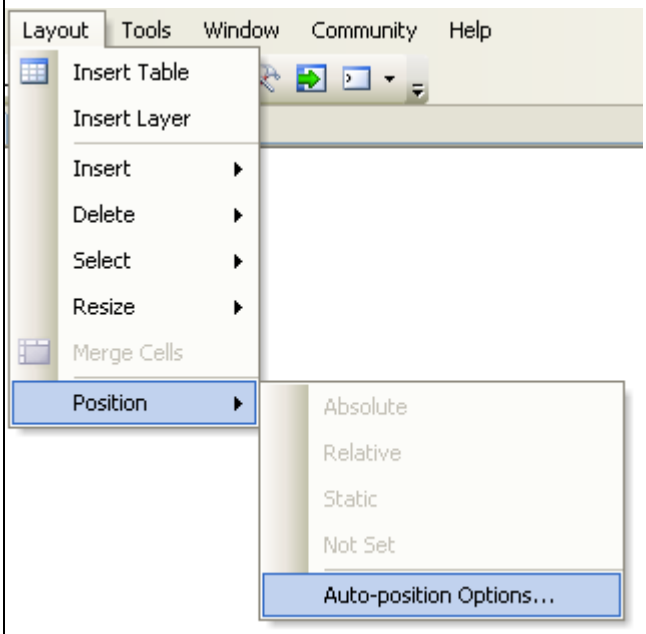
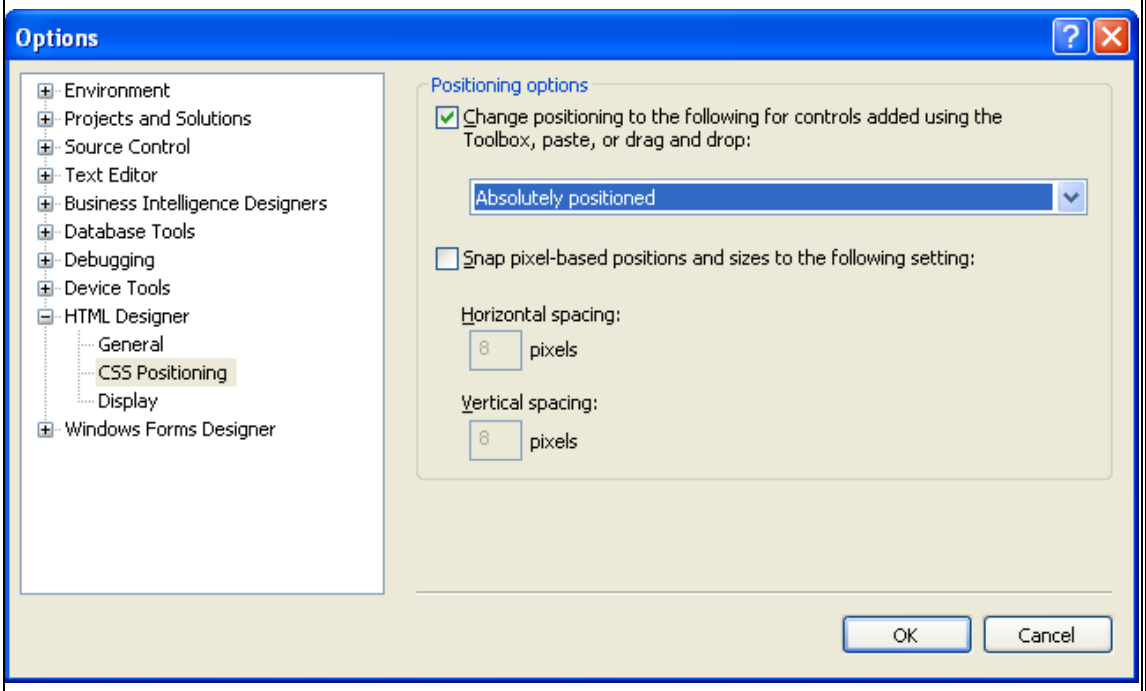
10	To deploy the application to remote nodes first make sure the Tags as described in steps 4, 5, and 6 are set to a Network Node or IP Address. Then simply copy the files in the bin\Release directory to the target systems or follow the Smart Client deployment section in this help file to deploy your application using Click Once Deployment. **Note:** You can also use the OPCControlsNetworkNodes component and assign a network node alias to change all "localhost" tags to the desired remote node. This is done with the AddNetworkNodeAlias method. Refer to the VB.NET Example on the exact syntax of how to use this method. Notice how all OPC Controls data sources for a particular node can be reassigned to a remote node with one simple call.
11	There are many different properties to each control. ImageIndex is available in many of the controls to display different images based on analog or discrete data. The OPC Controls.NET Data component can be used to access data via code with very simple methods. Refer to the Forms FormReadValues and FormWriteValues in the VB.NET Example. Refer to the VB.NET example for demonstration example of all properties of all components.

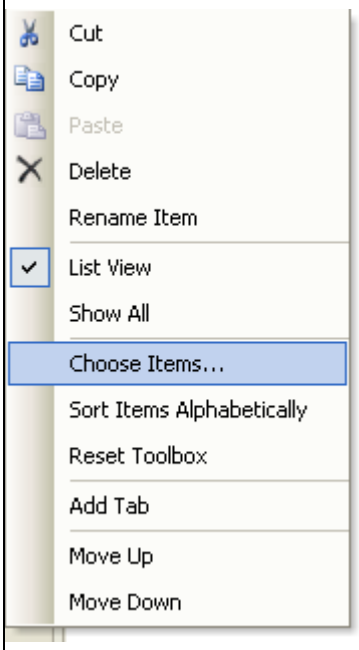
Add HMI Controls to ASP.NET Web Application

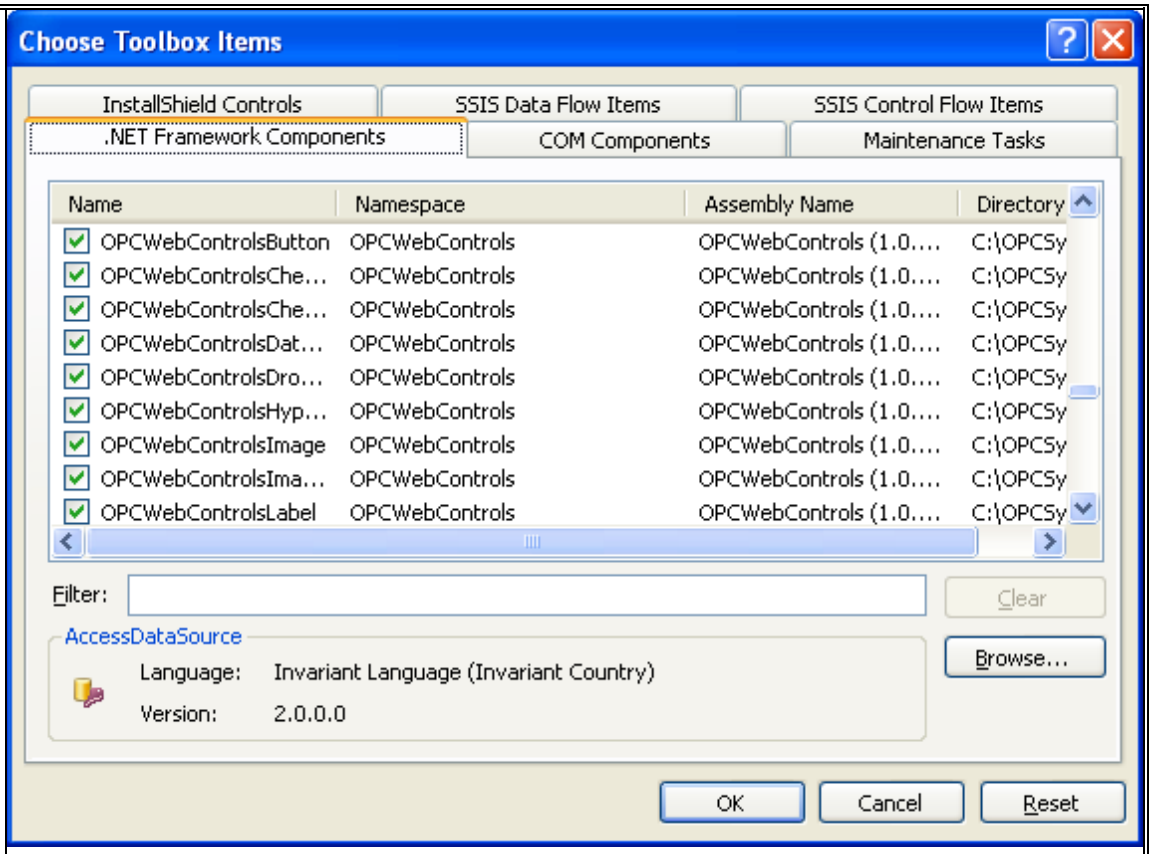
The following steps can be used to add real-time visualization and operator control to a C#, J#, or Visual Basic.NET web application. All properties are programmatically accessible. The following example demonstrates the trend window with no code required. The OPC Web Controls components can also be used with Visual Studio 2003 web applications with the .NET Framework 1.1 version of OPC Systems.NET.

Step	Task
1	Start Visual Studio 2005 or Visual Studio 2008 and select File->New->Web Site to create a new C#, J#, or VB ASP.NET web application. 

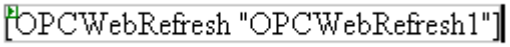
2	Select ASP.NET Web Site and specify your desired Language in the lower right. 
3	Expand the Default.aspx web page in the Solution Explorer and select View Designer. **Note:** If you do not see the Solution Explorer select View->Solution Explorer. 

4	From the VS menu select Layout->Position->Auto-position Options. 
5	Under the HTML Designer set the CSS Positioning to Absolutely positioned. **Note:** You are welcome to use any positioning style you desire, this simply pointed out for new web developers. 

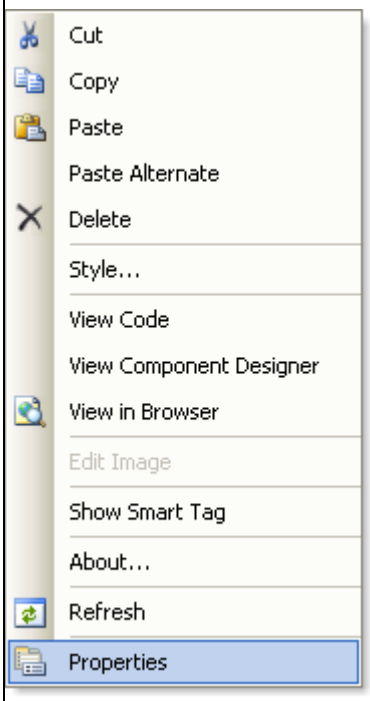
6	From the Toolbox if OPCWebControls and OPCWebRefresh are not available right click in the Toolbox and select Choose Items. If it is available to step 4.  From the .NET Framework Components select all of the OPC Web Controls components and the OPCWebRefresh control and then select OK.

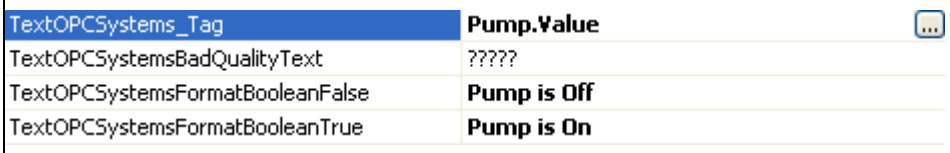
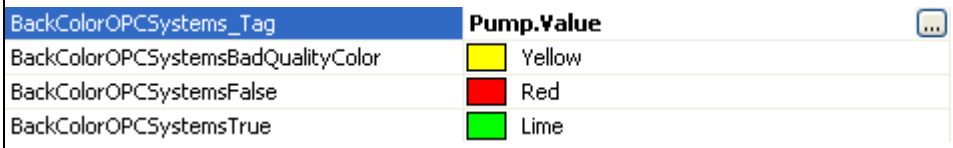
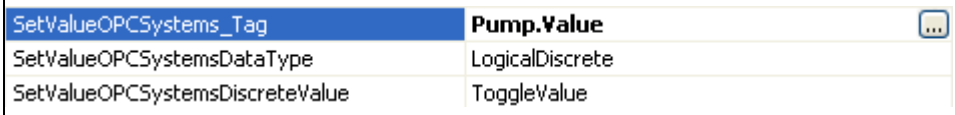
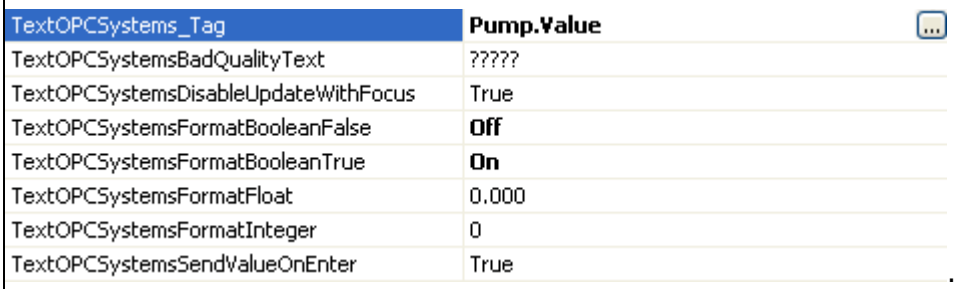


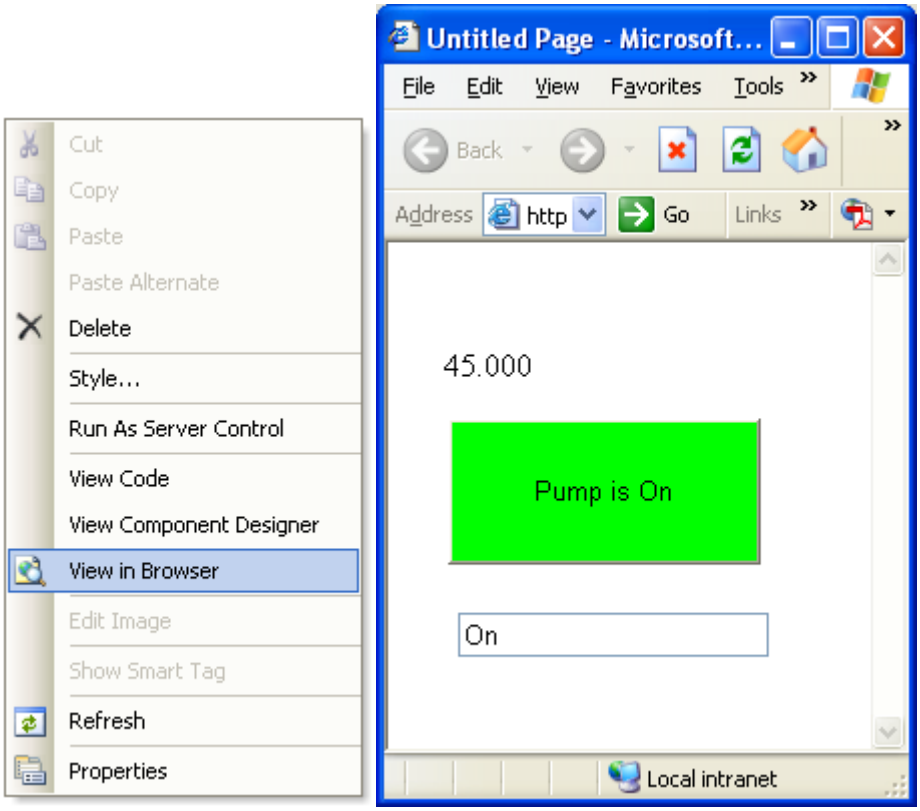
7 Add one OPCWebRefresh control to the WebForm.



For faster updates set the RefreshRate Property of the OPCWebRefresh1 control.

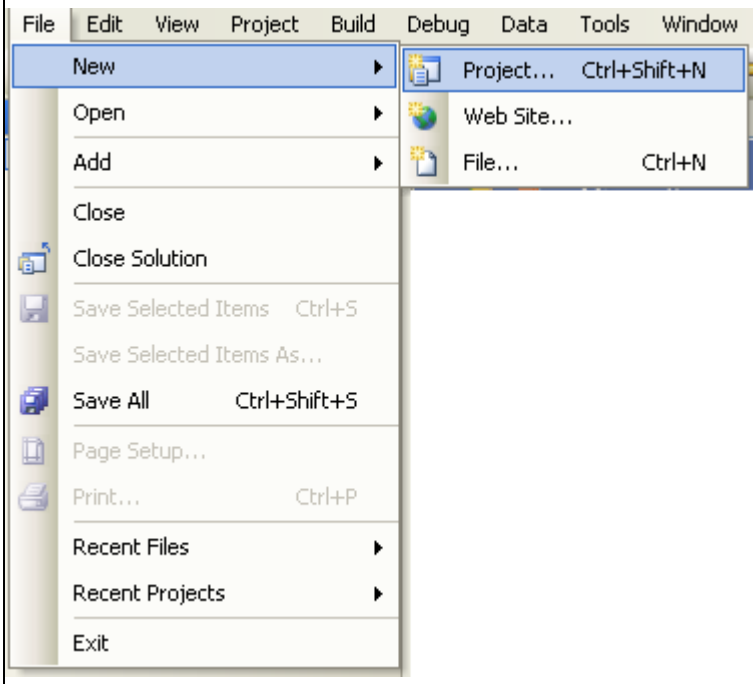
8	Add an OPCWebControlsLabel onto the WebForm.  Right click on the OPCWebControlsLabel window and select Properties.  Select the TextOPCSystems_Tag property and use the browse button at the right to set the OPC Systems.NET Tag to Ramp.Value.  **Note:** All Tag names are case sensitive. Ramp.Value is valid, ramp.value is not. **Note:** If you want the web application to be deployed across a network to remote PCs select the Network Node or enter an IP Address in the NetworkNode field and use the Select button to include the network node or IP Address of the OPC Systems Service source. If the web application will be running on the local system and simply need access to the web application through remote browsers use the Local service. **Note:** You can use the DirectOPC interface to connect directly to OPC Server Items is you just need to gain access to the server items without having to create OPC Systems.NET Tags.

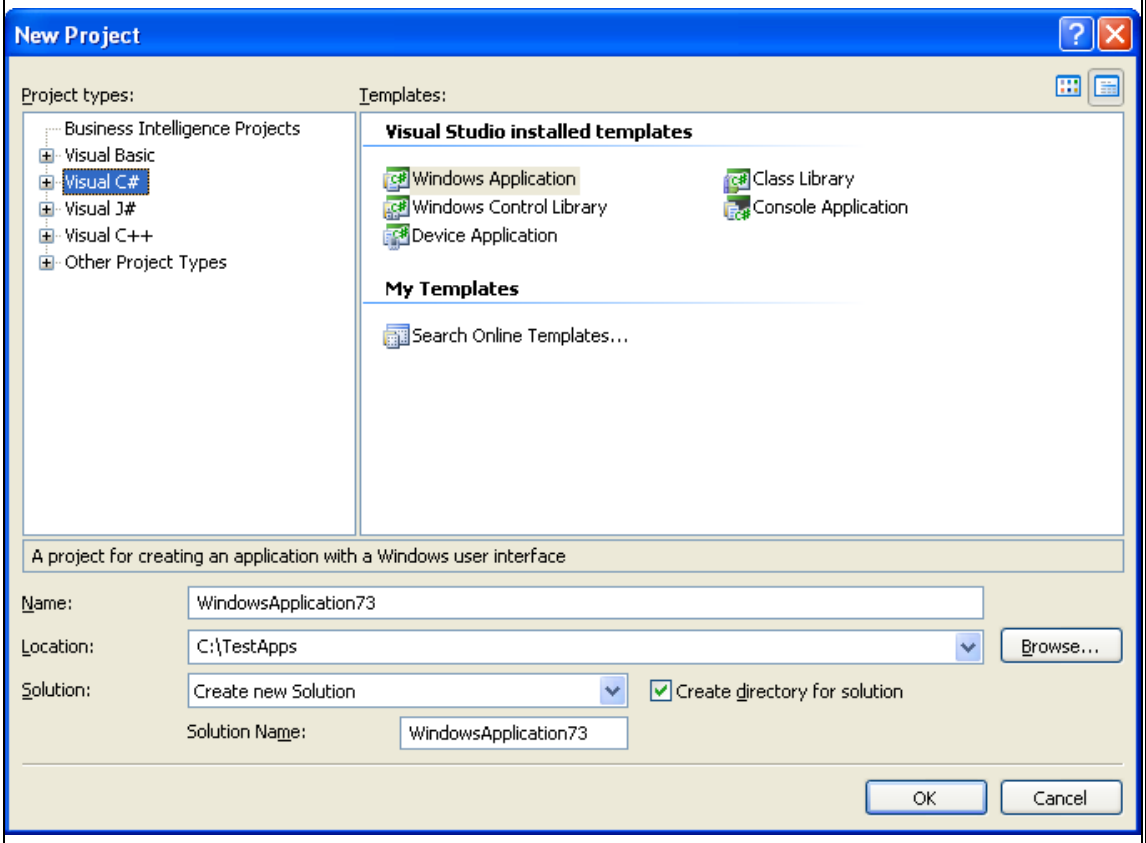
9	Add an OPCControlsButton to the Form.  Set the TextOPCSystems_Tag to Pump.Value. If the Pump Tag does not exist create a Boolean Tag using Configure-Tags with the name Pump. Set the Format fields as defined below..  Set the BackColorOPCSystems_Tag to Pump.Value.  Set the SetValueOPCSystems_Tag to Pump.Value. 
10	Add an OPCWebControlsTextBox to the Form.  Set the TextOPCSystems_Tag property to Pump.Value. The Format properties for Boolean to Off and On. 

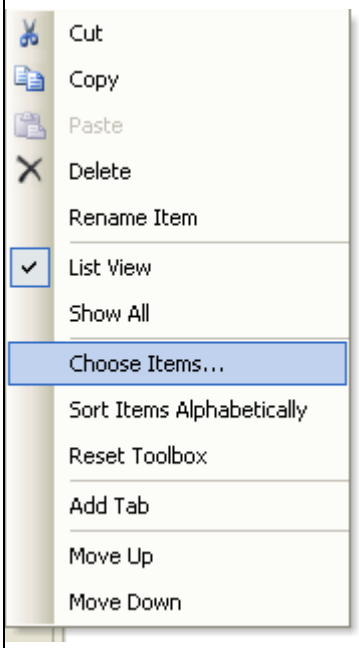
11	Right click on the WebForm and select View in Browser. 
12	Set the application for Release mode and build your application. Typically the application files are under the Inetpub\wwwroot directory. Refer to the OPC Web Controls.NET section in the OPC Systems.NET help file for a list of all controls and their properties.

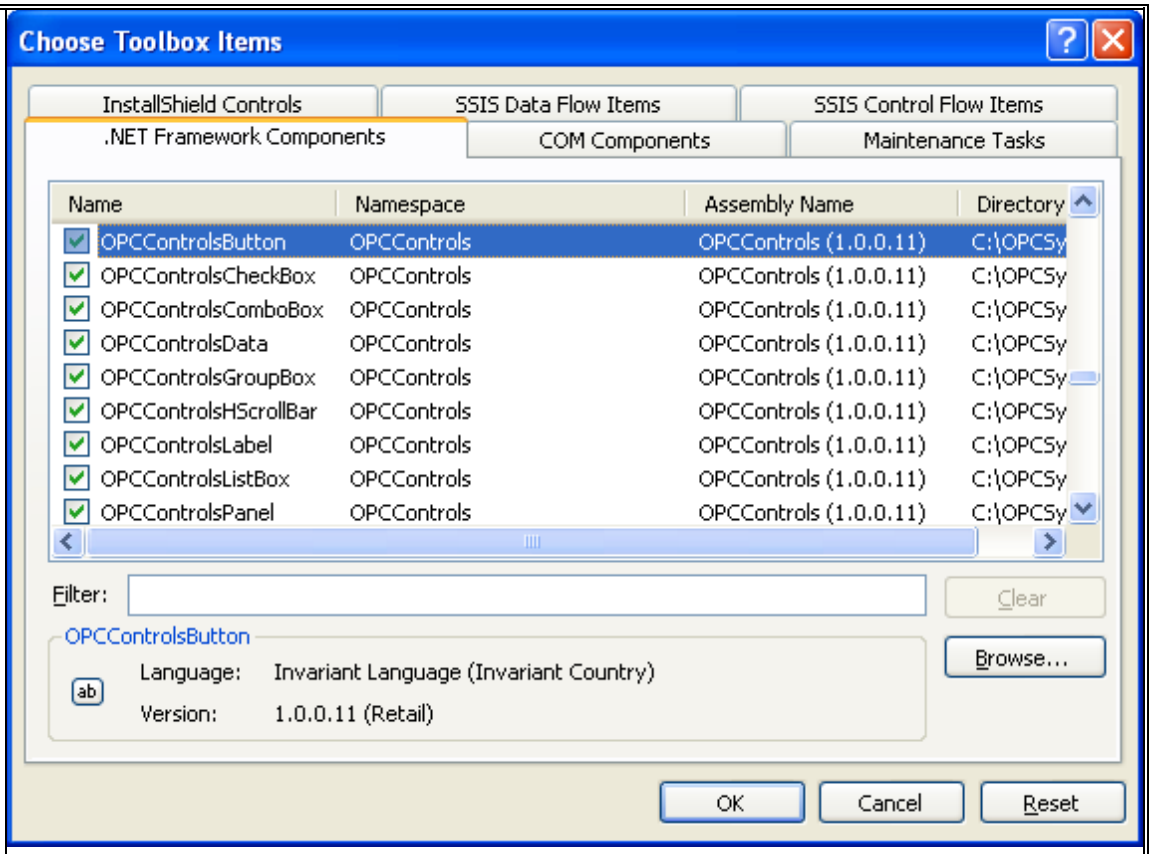
Add Data Control to Visual Studio Application

The following steps can be used to add real-time read and write data functions to a C#, C++, or Visual Basic.NET WinForm application. The OPC Controls.NET data component can also be used in ASP.NET Web applications. Refer to the Read Values and Write Values Forms in the VB.NET example for programmatic interface of using the OPC Controls Data component. All properties are programmatically accessible. The component can also be used with Visual Studio 2003 with the older 1.1 Framework version of OPC Systems.NET.

Step	Task
1	Start Visual Studio 2005 or Visual Studio 2008 and select File->New->Project to create a new C#, C++, or VB. 

2	Select Windows Application as the project type. 

3	From the Toolbox if OPCControls Data component is not available right click in the Toolbox and select Choose Items. If it is available to step 4.  From the .NET Framework Components select the OPC Controls Data component and then select OK.



4 Add an OPCControlsData component to the Form.

5	Add a Button component onto the Form.
	
	 
	
	Change the Name of the button to buttonAddTags..
	
	 
	
	Change the Text property of the button to Add Tags.
	
	 
	
	Double click on the button and enter the following code.
	
	For VB:
	
	`Private Sub ButtonAddTags_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonAddTags.Click`
	`  Dim tagsToAdd(1) As String`
	`  tagsToAdd(0) = "Ramp.Value"`
	`  tagsToAdd(1) = "Pump.Value"`
	`  OpcControlsData1.AddTags(tagsToAdd)`
	`End Sub`
	
	For C#:.
	
	`private void buttonAddTags_Click(object sender, EventArgs e)`
	`{`
	`  string[] tagsToAdd = new string[2];`
	`  tagsToAdd[0] = "Ramp.Value";`
	`  tagsToAdd[1] = "Pump.Value";`
	`  opcControlsData1.AddTags(tagsToAdd);`
	`}`
	
	Note: If you wish to run this application on remote PCs make sure to include the Network Node or IP Address of the OPC Systems Service source. To connect to remote tags simply include a network node name or IP address as follows.
	
	"\\192.168.0.1\Ramp.Value"
	
	You can also use the OPCControlsNetworkNodes component to alias one network node for another. Refer to the VB.NET Example for a working demonstration.
	
	Note: All Tag names are case sensitive. Ramp.Value is valid, ramp.value is not.

Note: You can additionally use DirectOPC items with the OPC Controls.NET Data component to connect directly to OPC Server Items without having to create OPC Systems.NET Tags. To see the full syntax of a DirectOPC Item use an OPC Control Label and browse an OPC Systems.NET Tag for the Text_OPCSystems_Tag property and select DirectOPC.

The following is an example of a DirectOPC Item.

[DirectOPC:KEPware.KEPServerEx.V4\Channel_1.Device_1.Tag_1:DirectOPCUpdateRate:1:DirectOPCDataType:Double Float:DirectOPCAccessPath:]

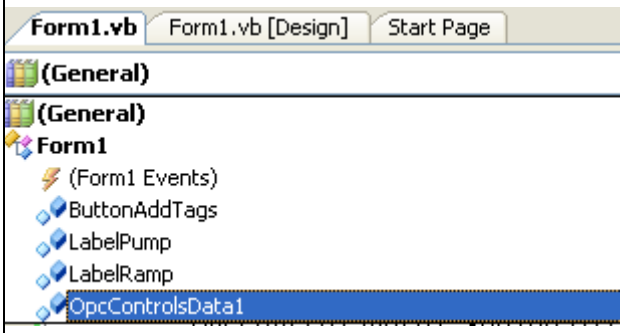
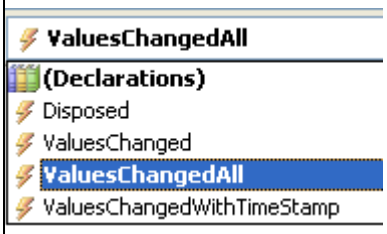
OPC Server: KEPware.KEPServerEx.V4

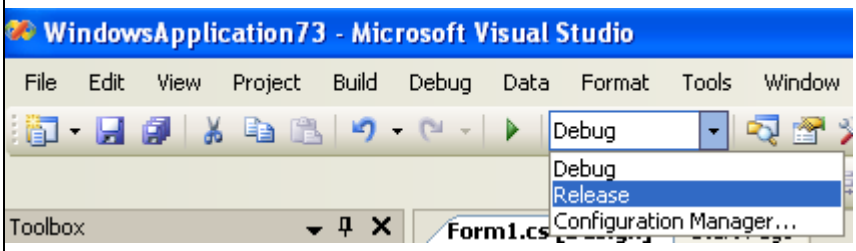
OPC Item: Channel_1.Device_1.Tag_1

OPC Update Rate: 1

Data Type: Double Float

All client components can use this alternative syntax for connecting directly to OPC Servers. This provides a direct pass through for OPC Items on local and remote systems.

6	For Visual Basic view the code of Form1.
	
	Select OpcControlsData1 from the object list.
	
	 
	
	Select ValuesChangedAll from the event list.
	
	 
	
	Enter the following code or your own code to do what every you desire with the data. Make sure to first check the Quality as True before using the Value as a valid value.
	
	`Private Sub OpcControlsData1_ValuesChangedAll(ByVal Tags() As String, ByVal Values() As Object, ByVal Qualities() As Boolean, ByVal TimeStamps() As Date) Handles OpcControlsData1.ValuesChangedAll`
	`Dim ValueString As String`
	`Dim Index As Int32`
	`For Index = 0 To Tags.GetLength(0) - 1`
	`If Qualities(Index) Then`
	`ValueString = "Tag: " + Tags(Index) + " is " _`
	`+ Values(Index).ToString + " at " _`
	`+ TimeStamps(Index).ToString("HH:mm:ss.fff")`
	`Else`
	`ValueString = "Tag: " + Tags(Index) + " is bad"`
	`End If`
	`Next`
	`End Sub`
	
	All values are queued in the order they are received from the Service and you may receive multiple values for the same tag in the same event.

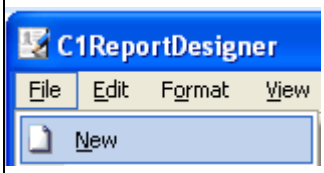
7	Run the application in Debug or Release mode. 
8	To deploy the application to remote nodes first make sure the Tags as described in step 5 are set to a Network Node or IP Address. You can also use the OPCControlsNetworkNodes component to alias the "localhost" node for another IP Address or node name. Registered domain names are also allowed. Then simply copy the files in the bin\Release directory to the target systems or follow the Smart Client deployment section in this help file to deploy your application using Click Once Deployment.
9	Refer to the VB.NET example for demonstration example of all properties of all components. The Read Values Form is a good resource for programmatic interface for reading values and working with both a Queue and a Hashtable. The Write Values Form is a good resource on how to programmatically write values.

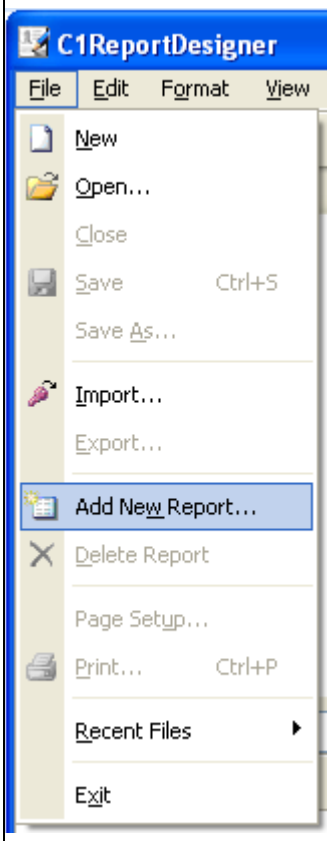
Chapter 7

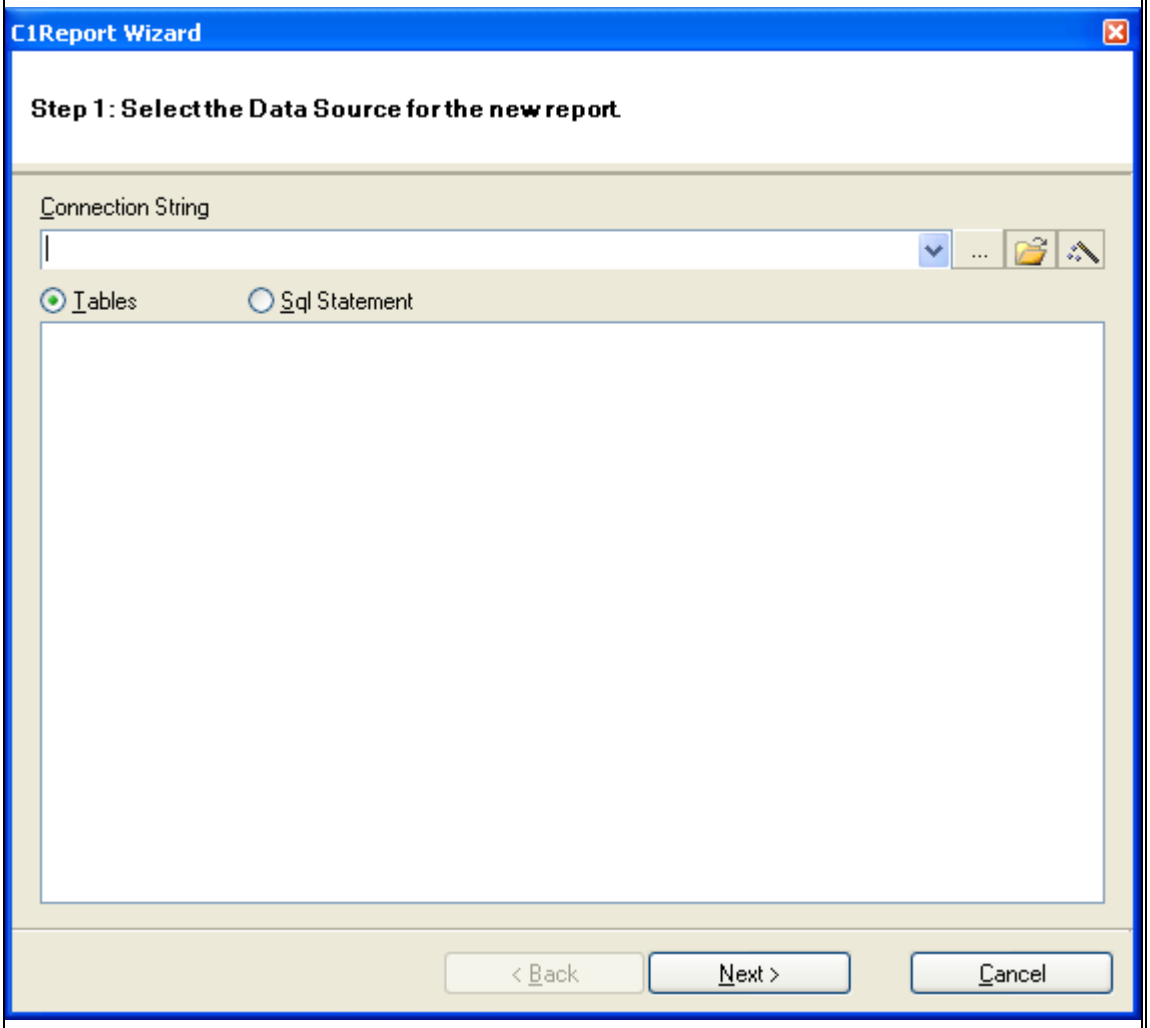
Chapter 7 - Reports

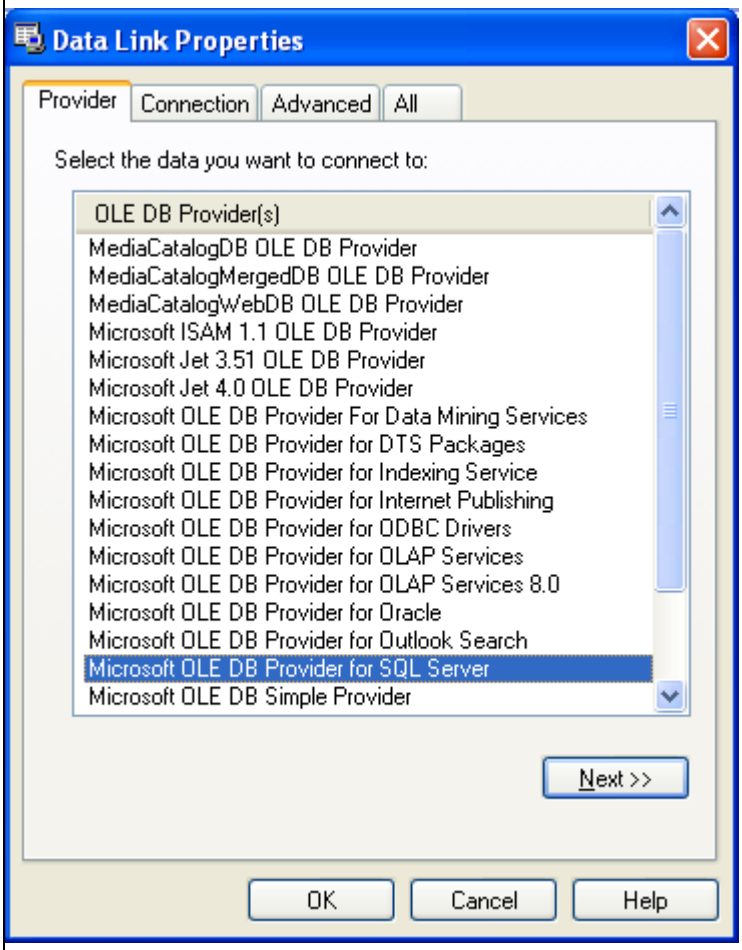
Configure Reports

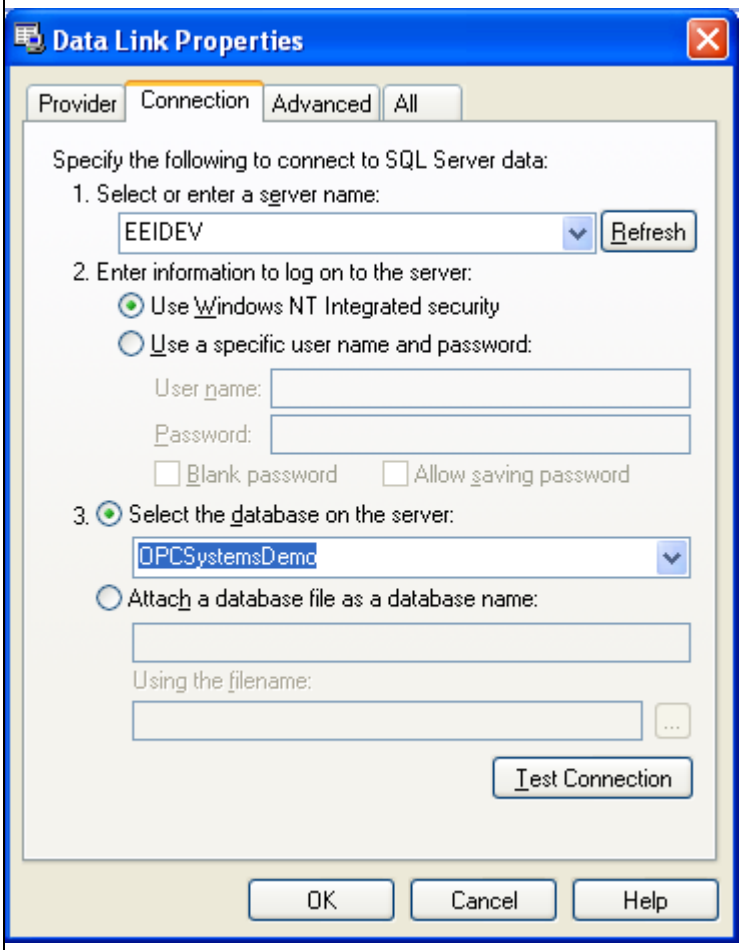
Reports can be created to generate PDF, HTML, Word, Excel, Text, and printer reports automatically with Configure->Reports or manually with the Report Viewer.

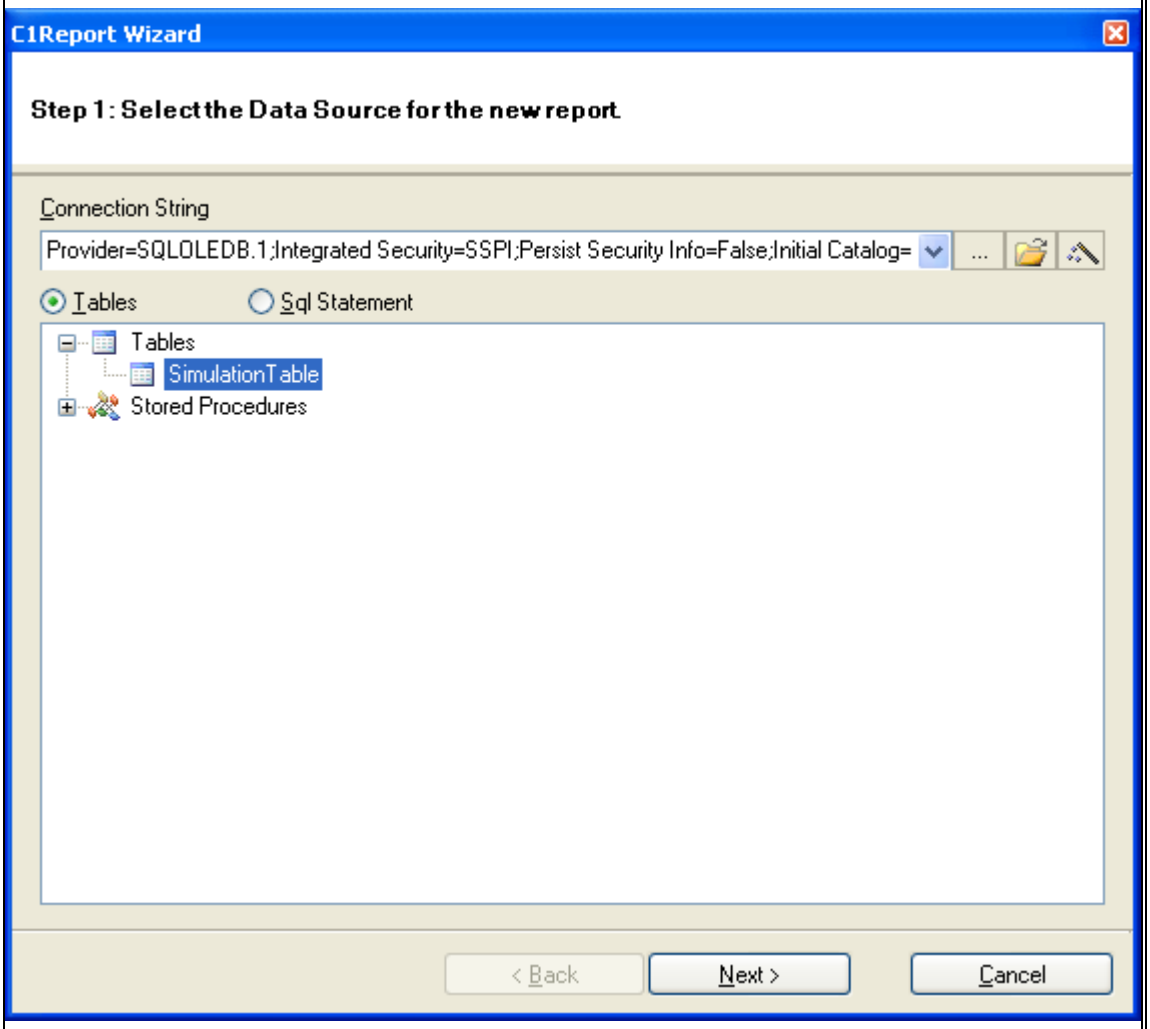
Step	Task
1	First complete the Configure Data Logging example and put the OPC Systems Service in Runtime mode to log data.
2	Start the Report Designer application from Component One.
3	Select File-New to clear the current Report Designer workspace. 

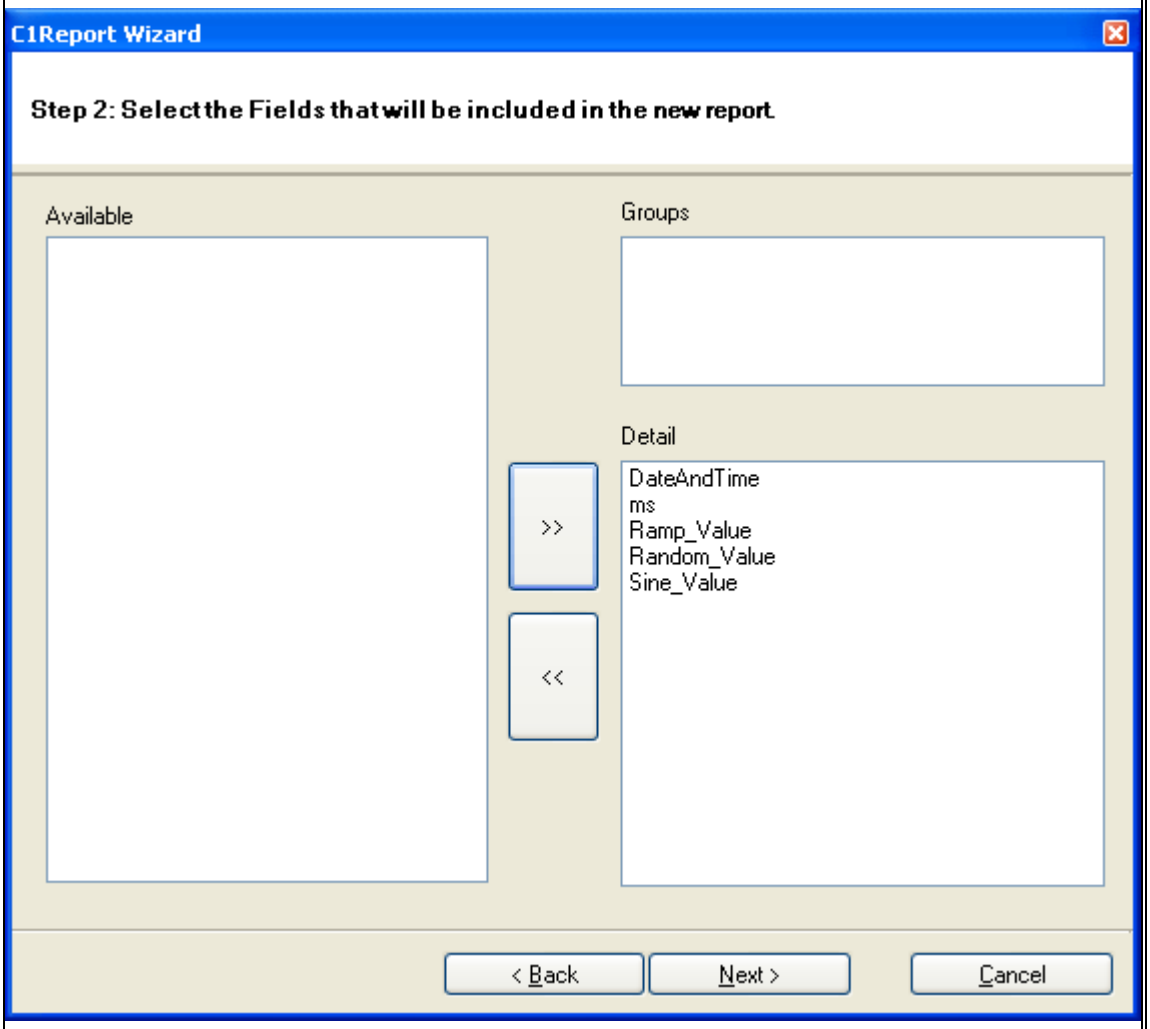
4	Select File-Add New Report 

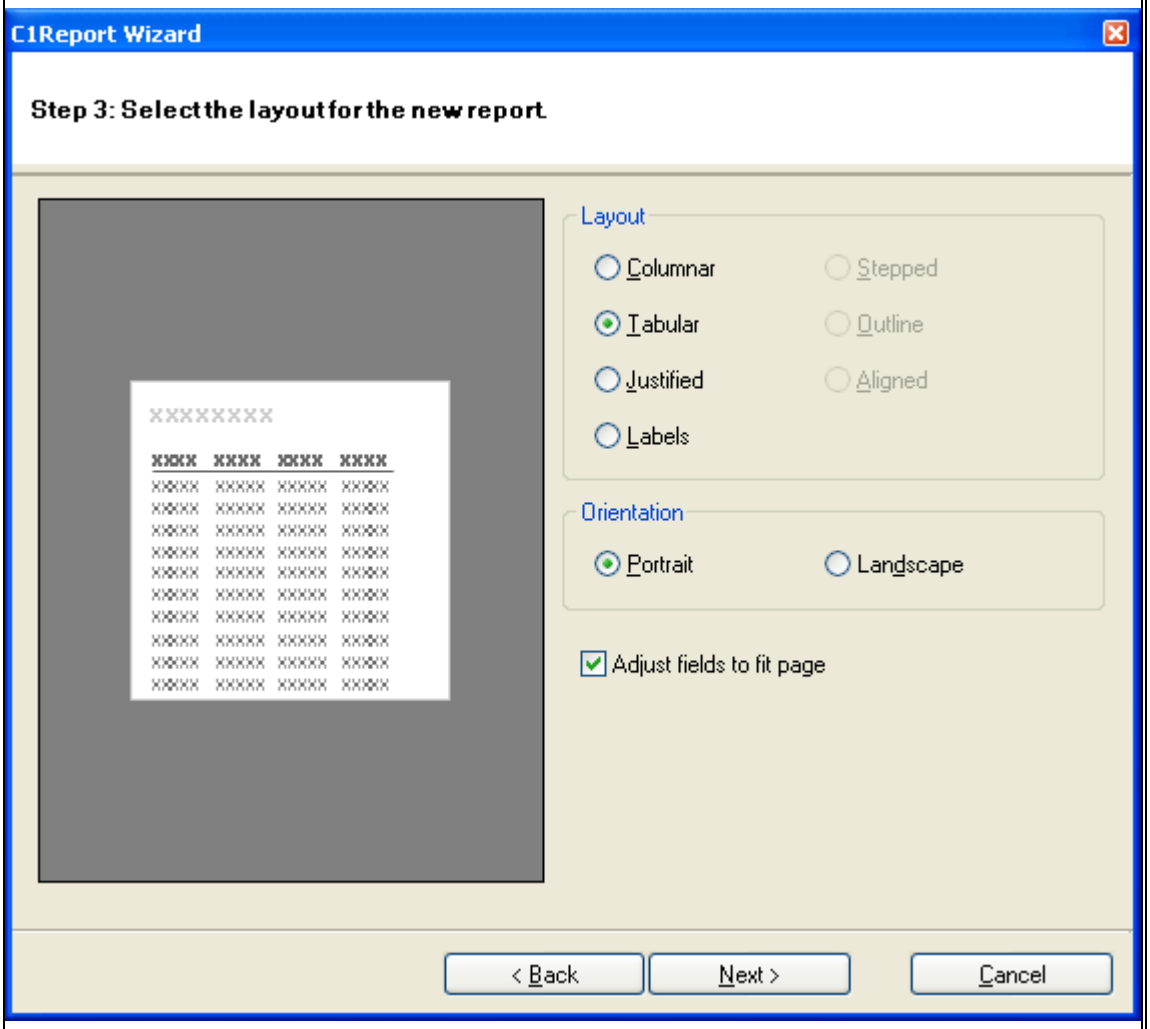
5	... Use the Build Connection String button to specify a data source. 

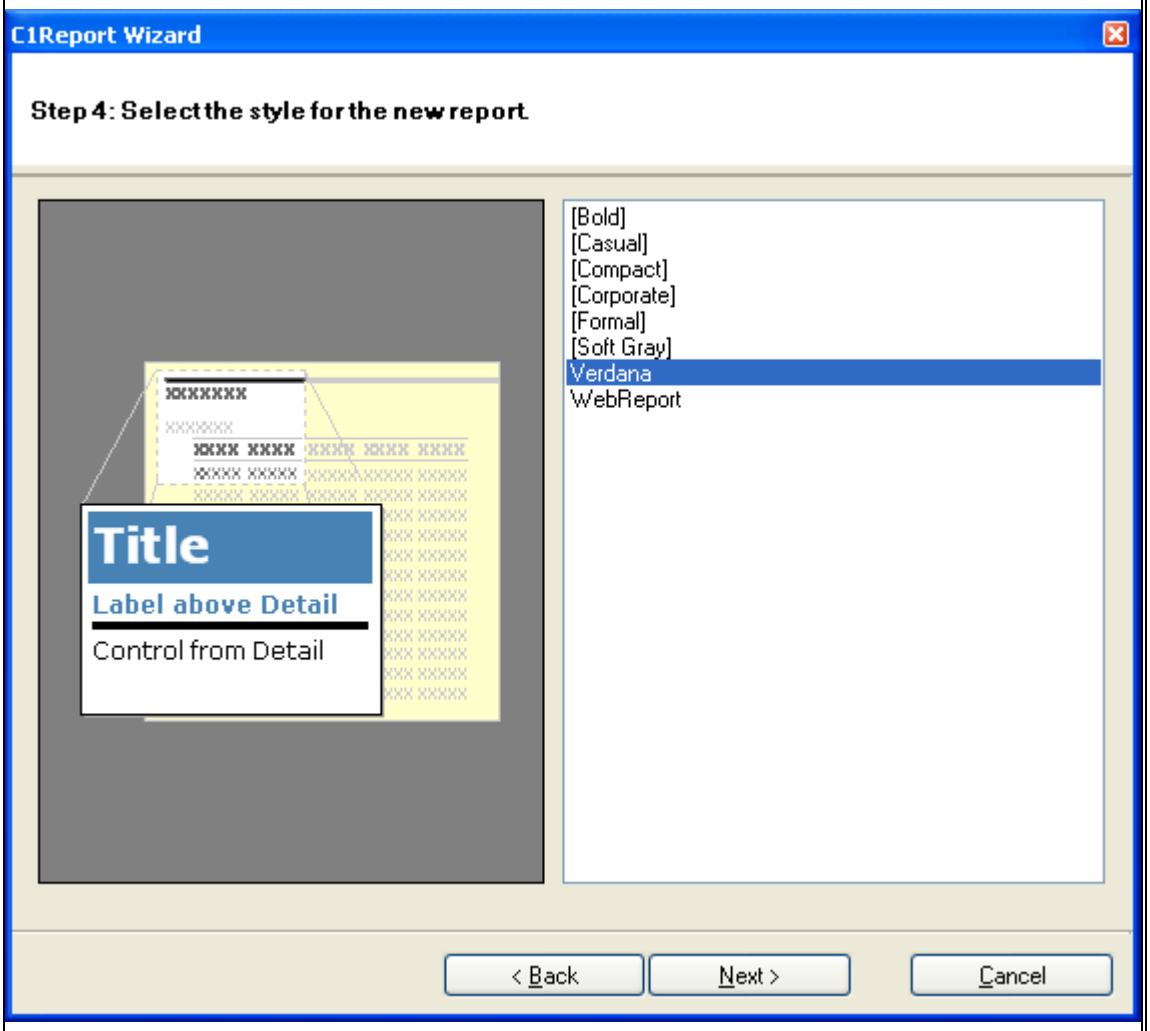
6	Specify the correct Provider for your database. 

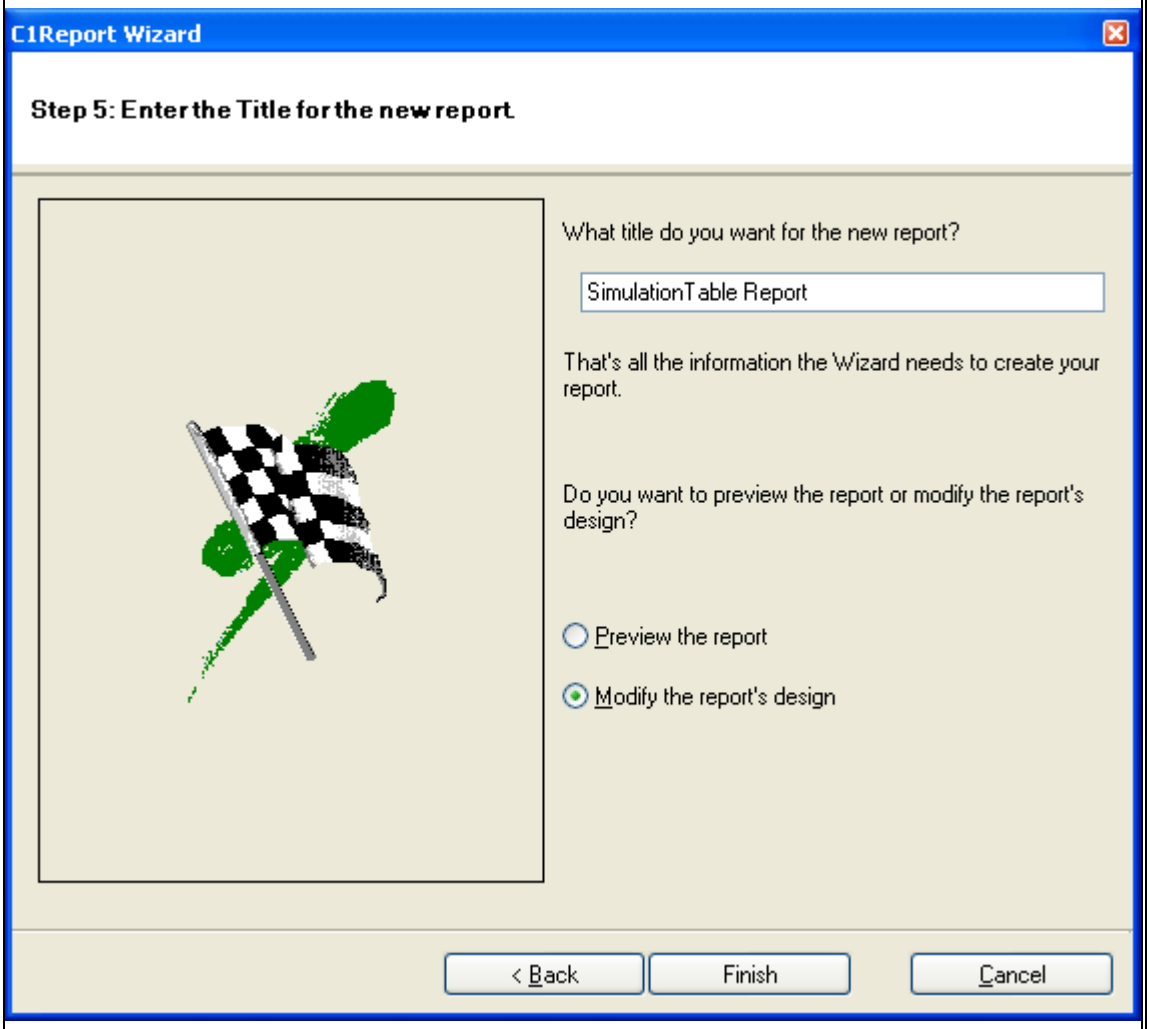
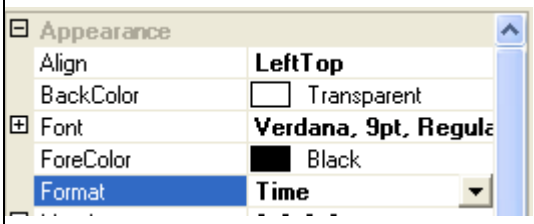
7	Select the proper Server with Windows NT Integrated security if using SQL Server Desktop. Select OPCSystemsDemo database. 
8	Use the Test Connection button to verify proper database connection and select OK 

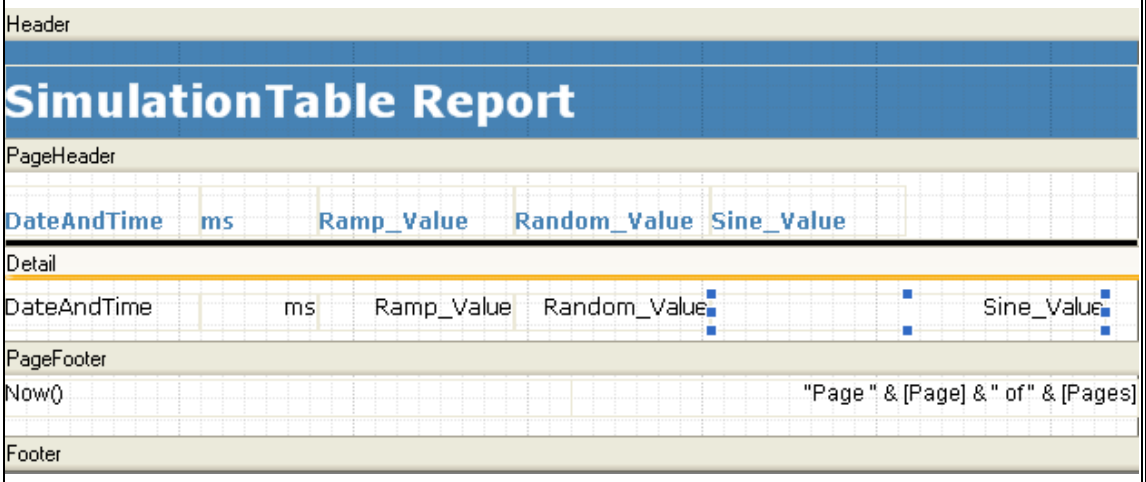
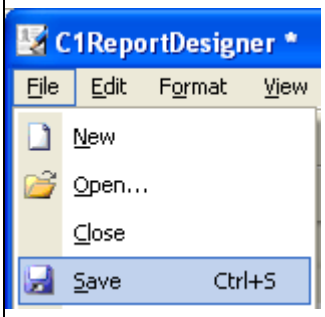
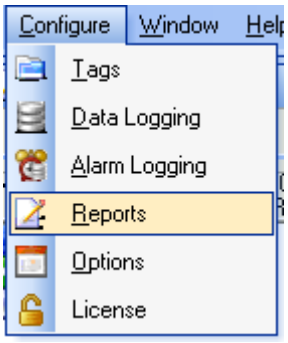
9	Select SimulationTable from Tables and then Next 

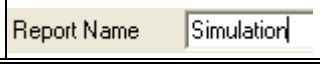
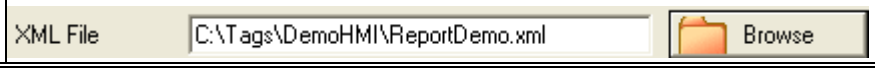
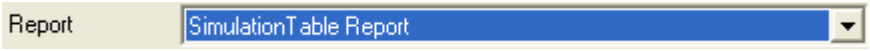
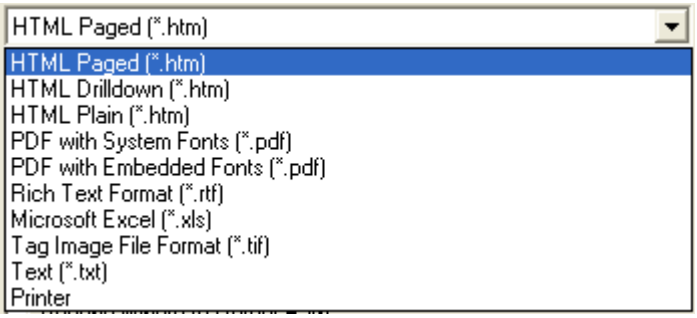
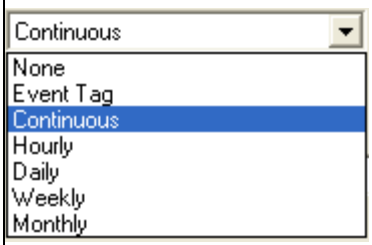
10	Move all Available fields into the Detail window. 

11	Select the default of Tabular report layout, then Next. 

12	Select the desired template style. The report colors, fonts, and images can be altered from these defaults. 

13	Select to Modify the reports' design. 
14	Select the DateAndTime field in the Detail section of the report and using the Properties window in the lower left set the to set the Format Property to Time. 

15	Stretch the size of the Sine_Value field in the Detail window to a larger width. 
16	Select File-Save and save the file with the name ReportDemo in the directory of your choice.  This creates the XML file that is selected in the Configure Reports step.
17	Start Configure OPC Systems application.
18	Select Configure-Reports. 

19	Select the Local OPC Systems Service. 
20	Enter the Report Name of Simulation in the field in the upper right. 
21	Browse for the XML you just created with the Report Desinger. 
22	Select the Report you created in the Report Designer using the Report list box. 
23	Select the desired Output Type. Choose a type that you have a application to view the file with. 
24	Specify the Output Path for the file. You do not need to put an extension on the file.  Specify the
25	Set the Execution Type to Continuous. 

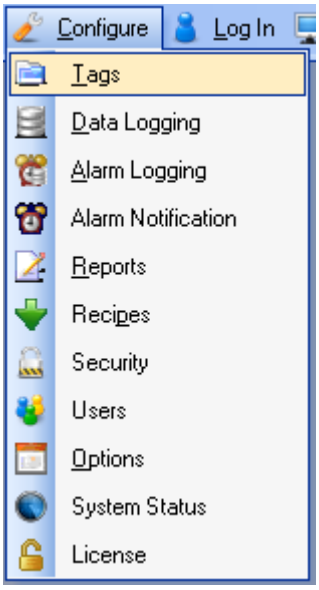
26	Set the Execution Rate to 60 Seconds.
27	Specify the Filter as None for now. With large amounts of data this would be important to filter the query to only the data of interest for the Report.
28	Select the Add button to add the Report. 
29	Select the Save button on the toolbar at the top. 
30	Save the file DemoReport.Reports in the directory C:\OPCSystemsDemo. Use Configure-Options to automatically load this configuration file when the OPC Systems Service starts. When you start the Service in Runtime mode reports will automatically be generated every minute. You can also define an output type of PDF and set the option to E-Mail the report when it is generated.

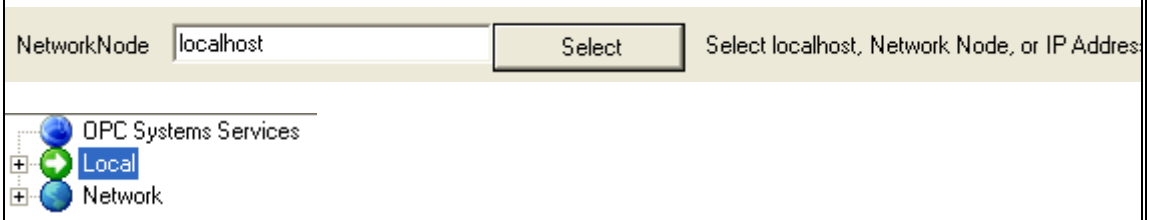
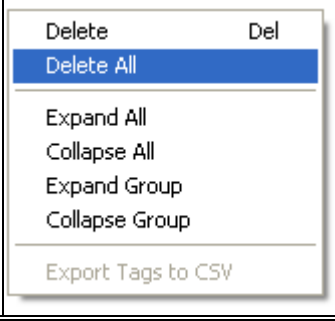
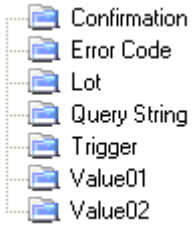
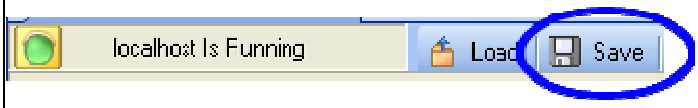
Chapter

8

Chapter 8 – Configure Recipes

Recipes are used to transfer values from a database to OPC Systems.NET Tags. If the Tags Data Source is set to OPC Item the values will also be written to the OPC Item the Tag is defined to. The most common implementation is to have a variable in a PLC trigger the recipe execution. The PLC will then wait for either the Confirmation or Error Code to be written to the PLC to indicate if the recipe execution has completed. If in Error the PLC can then retry the Trigger.

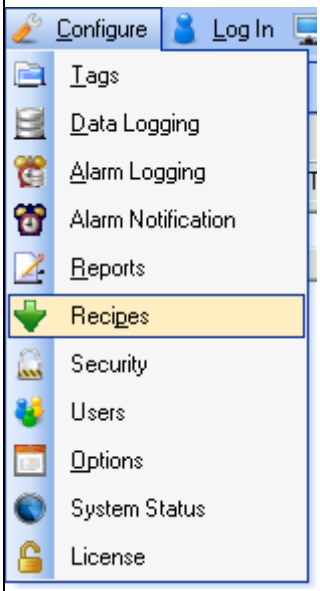
Step	Task
1	Start Configure OPC Systems application. 
2	Select Configure-Tags to add some Tags to use with the Recipe execution. 

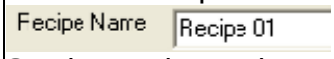
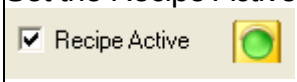
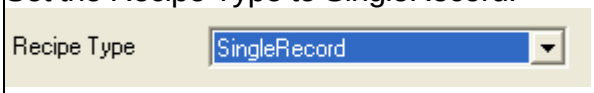
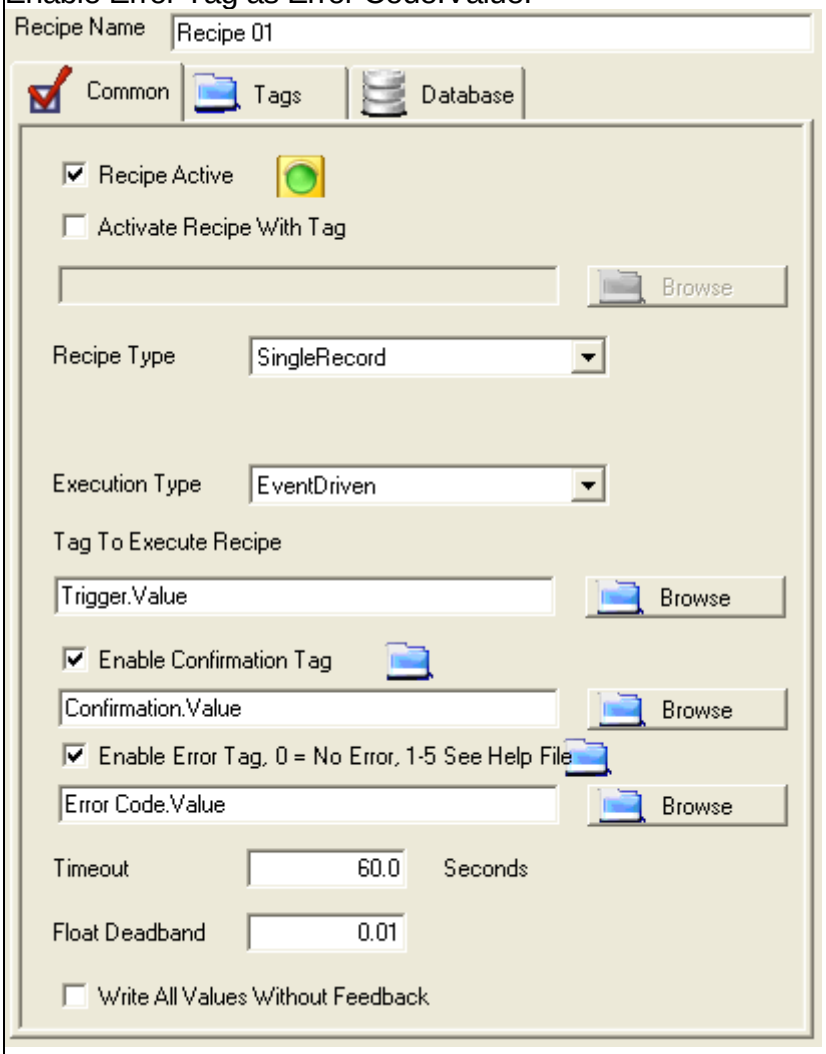
3	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left. 
4	If you are working on an existing Tag configuration you want to keep save it now. Right Click in the Tag Window and select Delete All to clear all Tags. 
5	Add the following Tags with the specific Data Types Trigger as a Boolean Value01 as an Integer Value02 as an Integer Lot as a String with the Value A Confirmation as a Boolean Error Code as an Integer Query String as a String with Data Source of Calculation "WHERE Lot = " & "'" & [Lot.Value] & "'" 
6	Select the Save button on the toolbar at the top and use the file name Recipe Tags. 

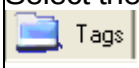
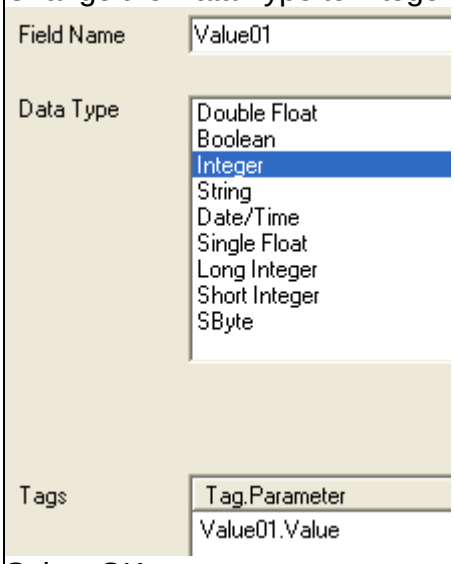
7	Use Microsoft SQL Server, Oracle, Access, or mySQL to create the following Table. Create a Database called OPCSystemsRecipes Create a Table called RecipeValues with the following structure. Enter the following values to the Table RecipeValues.
8	With the Configure OPC Systems application select Configure-Recipes.
9	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left.

Column Name	Data Type	Allow Nulls
Lot	char(10)	☑
Value01	int	☑
Value02	int	☑

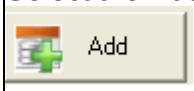
Lot	Value01	Value02
A	101	102
B	201	202
C	301	302



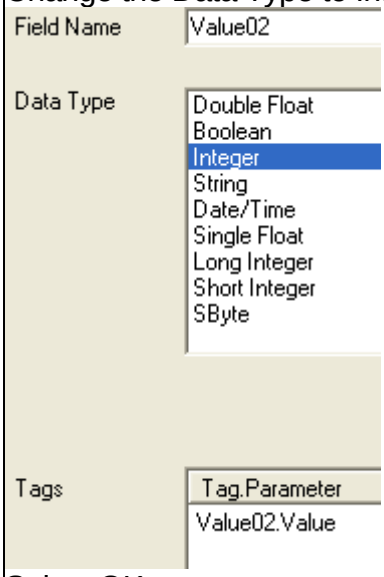
10	Enter the Recipe Name Recipe 01.  Set the Recipe Active.  Set the Recipe Type to SingleRecord.  Set the Tag to Execute Recipe as Trigger.Value. Enable Confirmation Tag as Confirmation.Value. Enable Error Tag as Error Code.Value. 

11	Select the Tags Tab  With the Tags properties you can assign the database table fields to the OPC Systems.NET Tags to write to when the recipe executes. If the Data Source of the OPC Systems.NET Tag is defined as an OPC Item it will automatically write the value to the OPC Item. You could also optionally use the DirectOPC selection to connect directly to OPC Items on OPC Servers instead of OPC Systems.NET Tags. Select the Add Field button.  Select the Tag and Parameter Value01.Value. Change the Field Name to Value01. Change the Data Type to Integer.  Select OK.

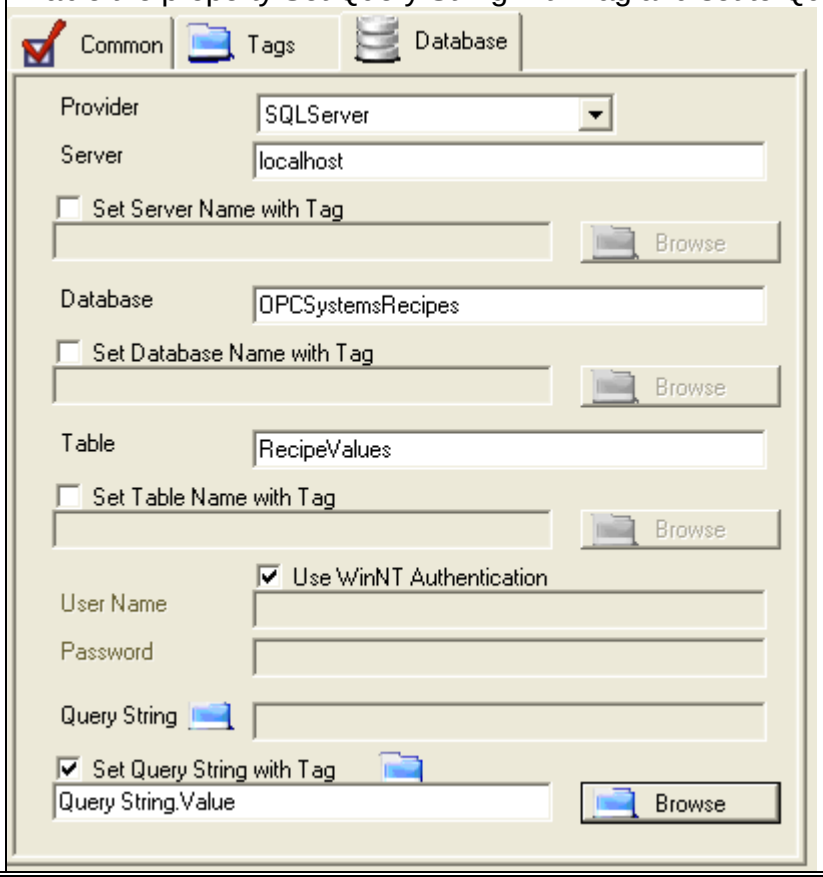
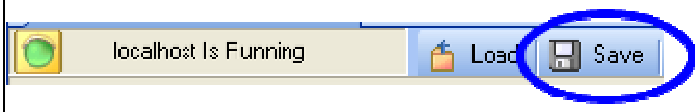
Select the Add Field button.



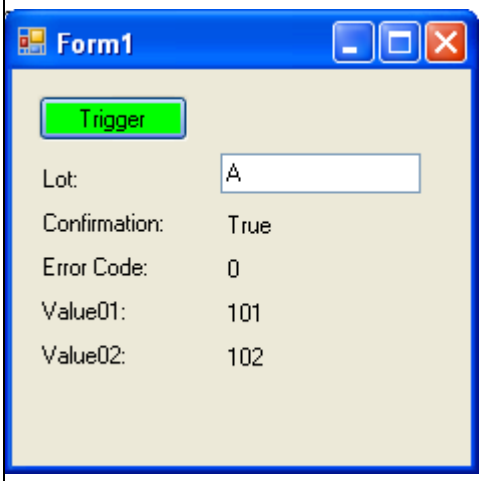
Select the Tag and Parameter Value02.Value.
Change the Field Name to Value02.
Change the Data Type to Integer.



Select OK.

12	Select the Database Tab. Define the proper database connection. Enable the property Set Query String with Tag and set to Query String.Value. 
13	Select the Add button in the lower left to add the Recipe. 
14	Select the Save button on the toolbar at the top.  Save the file RecipeExample.Recipes in the directory C:\OPCSystemsDemo.

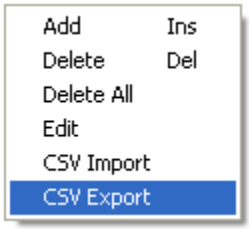
15	At this point you are ready to execute the recipe which you can do with Configure-Tags to set the Lot to A, B, or C and set the Trigger from False to True. You can then see that the Confirmation Tag will be set to True and Value01 and Value02 update if the Recipe is successful or the Error Code Tag will be a positive number if there is an error. Review the Recipe Execution Log found in the Error Log directly if there is an error. Optionally create a Standard Windows application with the following on a Form to recipe execution in a more interactive execution. If you are unfamiliar with the OPC Controls.NET product you may want to follow the steps in this Training Guide on Add HMI Controls to Visual Studio. OPCControlsButton Text = Trigger SetValueOPCSystems_Tag = True SetValueOPCSystems_Tag = Trigger.Value BackColorOPCSystems_Tag = Trigger.Value OPCControlsTextBox TextOPCSystems_Tag = Lot.Value OPCControlLabel TextOPCSystems_Tag = Confirmation.Value OPCControlLabel TextOPCSystems_Tag = Error Code.Value OPCControlLabel TextOPCSystems_Tag = Value01.Value OPCControlLabel TextOPCSystems_Tag = Value02.Value

	Run the Recipe Example application and enter the Lot as A, B, or C and click on the Trigger Button to see the values being set. 
16	When you execute a Recipe with Error Logging enabled under Configure-Options a log will be generated for the recipe execution in the same directory as the Error Log. A Queued Recipe Type will delete the returned record with the recipe execution is successful. A MultipleRecord Recipe Type has only 2 fields, one for the Tag Name to set and one for the Value to write to the Tag. Using this type you can define what Tags to write to in the database. When a recipe is executed it will only write to Tags where the desired value from the database is different than the current value of the Tag. For floating point values use the Float Deadband property under the Common tab to define what is the allowable range to determine if a value is the same or different. The Timeout property is the amount of time the recipe execution will wait for all values to be returned from the Tags. If one or more of the values does not read back from the OPC Server if defined as an OPC Item Data Source the recipe execution will end with an Error Code of Timeout. The property Write All Values Without Feedback will not wait for the values to be returned and will write all database values to all defined Tags. This is preferred for a faster execution when using the Execution Type Continuous.

Recipe CSV Export and Import

All Recipes can be exported to a CSV file along with all recipe field names to individual sub files by right clicking on the Local service and select Export CSV.

You can also export just the individual fields for a recipe group by right clicking in the Tags field list.



Recipe Programmatic Interface

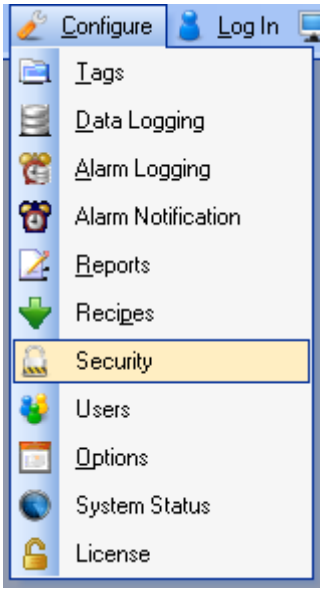
Use the OPC Systems component in your Visual Studio application to programmatic modify Recipe groups. Refer to the Configure Recipes Form and the Configure CSV Form in the VB.NET example for an example and how to add and modify recipe groups. Refer to the OPC Systems Component help file for all of the method syntax.

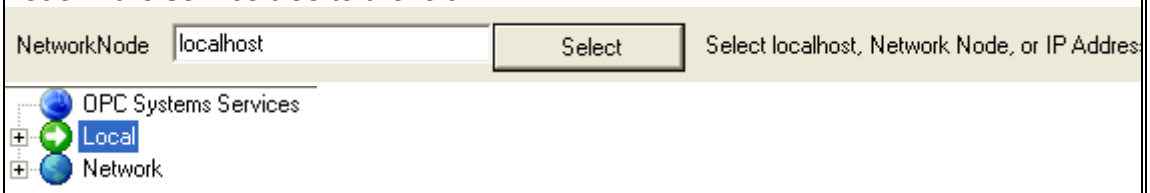
Chapter

9

Chapter 9 – Configure Security

Security can be implemented to restrict modifying configuration parameters and limit real-time and historical data access.

Step	Task
1	Start Configure OPC Systems application. 
2	Select Configure-Security. 

3	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left. 
4	The Default Security Group contains all of the security default settings if a remote client application is not logged and is connected to the service. The Enable All Features selection will enable all features for security access. You can add multiple Security Groups, each one having its own security settings. You then define Security Users with Configure-Users to assign a Security Group to a User. You cannot define the Default Security Group to a User.
5	The Common properties are general features that are common to the entire OPC Systems Service that can be restricted. The Tags properties are the different types of methods that can be performed on the real-time Tag database. The Read Tags properties provide the feature to disable or enable real-time data access to Tags. You can either Disable All Tags From Reading and then Enable specific Tags or leave all Tags for reading , but just Disable specific Tags. The Read Tags properties affect real-time data access from HMI controls like OPC Controls.NET and OPC Web Controls.NET, but also restrict real-time Trending and Data Logging. The Write Tags properties limit client applications from writing to Tags. The Trends properties are for a few specific name retrieval functions. The Trend RealTime properties help to limit access for real-time trending. The Trend History properties allow restriction of history replay from clients.

 The Data Log properties limit data logging configuration parameters from access and modification. To limit the actual data being logged use the Read Tags properties.

 The Alarms properties limit obtaining the Alarm Group Names defined in the Service.

 The Alarm Ack properties limit the ability to acknowledge alarms based on alarm priority and Alarm Groups.

 The Alarm RealTime properties limit the ability to access the current alarms based on alarm priority and Alarm Groups. This also restricts Alarm Logging.

 The Alarm History properties limit the ability to access historical alarms from a database based on alarm priority and Alarm Groups.

 The Alarm Log properties limit alarm logging configuration parameters from access and modification. To limit the actual alarms being logged use the Alarm RealTime properties.

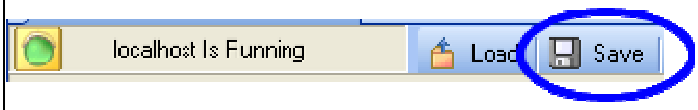
 The Alarm Notification properties limit alarm notification configuration parameters from access and modification.

 The Reports properties limit report configuration parameters from access and modification.

 The Recipes properties limit the recipe configuration parameters from access and modification.

 The Security properties limit the security configuration parameters from access and modification.

 Warning Make certain to first add a Security Group to all access and modification to the security configuration and add a User assigned to the new

	Security Group before disabling access and modification of Security in the Default Group.  The Options 1, 2, and 3 properties are to limit access of the parameters found under Configure-Options.
6	 Once you have defined a new Security Group select Configure-Users to define User Names, Passwords, and Security Groups defined to each user.
7	In order to save your Security and User configuration select the Save button on the toolbar at the top and use your desired file name.  Set the Default Security Configuration to load under Configure-Options.
8	Under Configure-Options note the 2 following properties. Security User Name for Service Security Password for Service These properties will allow the OPC Systems Service to run under this defined user account to give remote client access privileges for Read Tags for Calculations and Data Logging, and for Realtime Alarms for Alarm Logging. This is important if the remote OPC Systems Services have Security restrictions for these features.

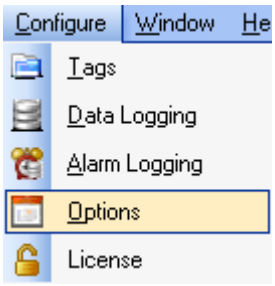
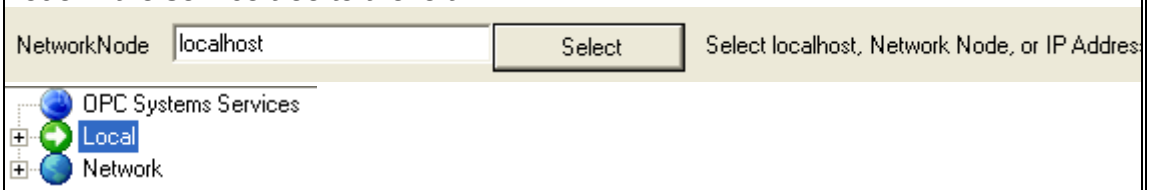
9	 Refer to the VB.NET Example on how to programmatically login to multiple client components in a Visual Studio Application. You can also use the LogIn icon on the Trend and Alarm Controls individually. This code is in the FormMain code of the VB.NET Example. The OPC Contrrols.NET LogIn and LogOff methods are found in the OPCControlsLogIn control that will control user access for all OPC Controls.NET components used in a Visual Studio application. The ShowUserLogIn method displays a Log In dialog for security access log in. You can also obtain the Current User with the CurrentUser function. The OPC Trend.NET component has LogIn and LogOff methods to use for programmatic interface, but also a LogIn icon that can be displayed on the toolbar. The OPC Alarm.NET component has LogIn and LogOff methods to use for programmatic interface, but also a LogIn icon that can be displayed on the toolbar. The OPC Web Controls.NET OPCWebRefresh contains the LogIn and LogOff methods for controlling for a specific web page. The OPC Web Trend control has LogIn and LogOff methods. The OPC Web Alarm control has LogIn and LogOff methods. The OPC Systems Component used for programmatic interface of all configurations has LogIn and LogOff methods.
10	The Configure OPC Systems application has a Log In selection on the main menu in order to grant access privileges to a service with security protection. It is important to know that features like CSV Import and Export may be restricted based on the security policy of the OPC Systems Service it is connected to.

Chapter 10

Chapter 10 - Default Settings

Configure Options

The OPC Systems.NET Service will automatically load the default configuration files specified under Configure Options. There is also a few adjustable parameters for the Service including enabling an OPC Server Watchdog and to re-load realtime trending and alarm data. The file that is generated with Configure-Options is OPCSystems.options in the installation directory of OPC Systems.NET (C:\Program Files\Open Automation Software\OPC Systems.NET\).

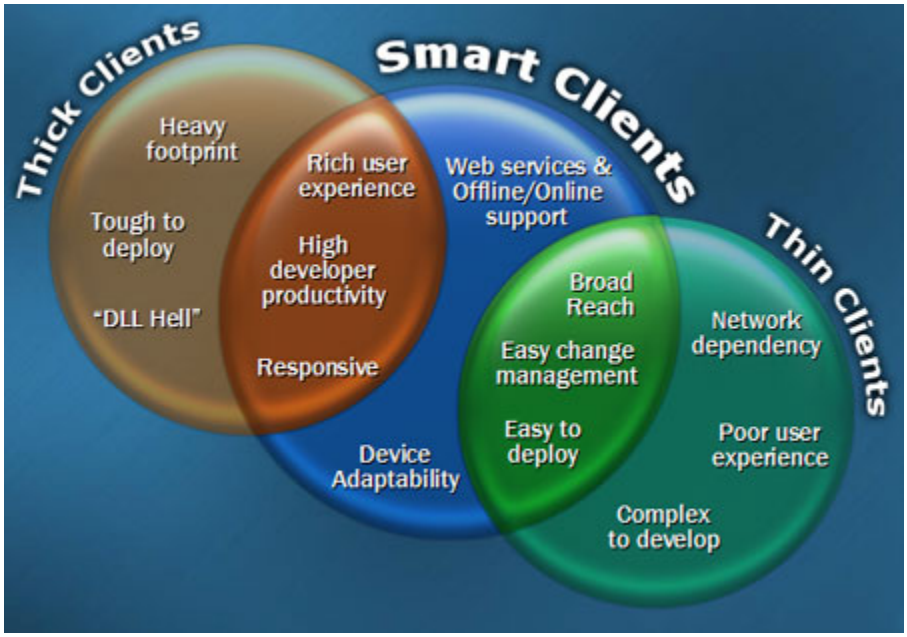
Step	Task
1	Start Configure OPC Systems application.
2	Select Configure-Options. 
3	Select the Local OPC Systems Service by selecting the Select button or the Local node in the service tree to the left. 
4	Check the Load Default Tag Configuration On Service Start. Browse for the file C:\OPCSystemsDemo\DemoTags.tags.

	The Error Log path defaults to the \Log directory under the installation directory (C:\Program Files\Open Automation Software\OPC Systems.NET\). You can assign this to any directory you like. Enable Transaction Logging only when you need to see the raw value being received by all OPC Server Items with Good Qualilty. All Bad Quality OPC Items will show up in the Error Log. It is recommend to set an OPC Server Watchdog Rate of 60 seconds if you are connecting to local or remote OPC Servers. If you are performing data logging from a remote OPC Systems Service or to a remote database engine it is recommend to enable Store Data Logging Buffering to Disk on Network Loss or Database Engine Failure. This will then buffer data to disk leaving RAM free when data cannot be delivered due to a temporary network loss. When using this feature you can limit the maximum number of time the buffering occurs to not overrun the hard disk for long periods of failure. If you have used the Time On and Counts feature of any Tag you will want to specify the File to Store Times and Counts in order to retain the numbers if your system restarts.
9	Select the Apply Changes button to save the changes for the OPC Systems Service you are connected to. 

Chapter

11

Chapter 11 - Smart Client Deployment



Applications developed with 100% managed components can be executed on remote systems without installing the application itself on the remote systems. This makes application updates extremely simple and provides a better user experience than web applications provide. The .NET Framework provides built in security to limit remote application execution without proper authorization.

Visual Studio 2005 and 2008 make Smart Client deployment very easy with a deployment wizard for the Project you wish to deploy. This wizard allows you to set version information that the remote systems will use to determine if an update is available. This deployment method requires the client systems to have version 2.0 of the .NET Framework if you are using Visual Studio 2005 and version 3.5 of the .NET Framework if you are using Visual Studio 2008.

Components that include COM or legacy ActiveX controls are not Smart Client compatible.

All .NET components of OPC Systems.NET are Smart Client compatible.

There are a few simple steps to deploy an application as a Smart Client.

The following examples walk you through a complete Smart Client execution.

Example Smart Client

The Example application installed with OPC Systems.NET is a Smart Client compatible application demonstrating remote connectivity to any OPC Systems.NET Service. The application includes OPC Controls.NET, OPC Trend.NET, OPC Alarm.NET, and the OPC Systems configuration component.

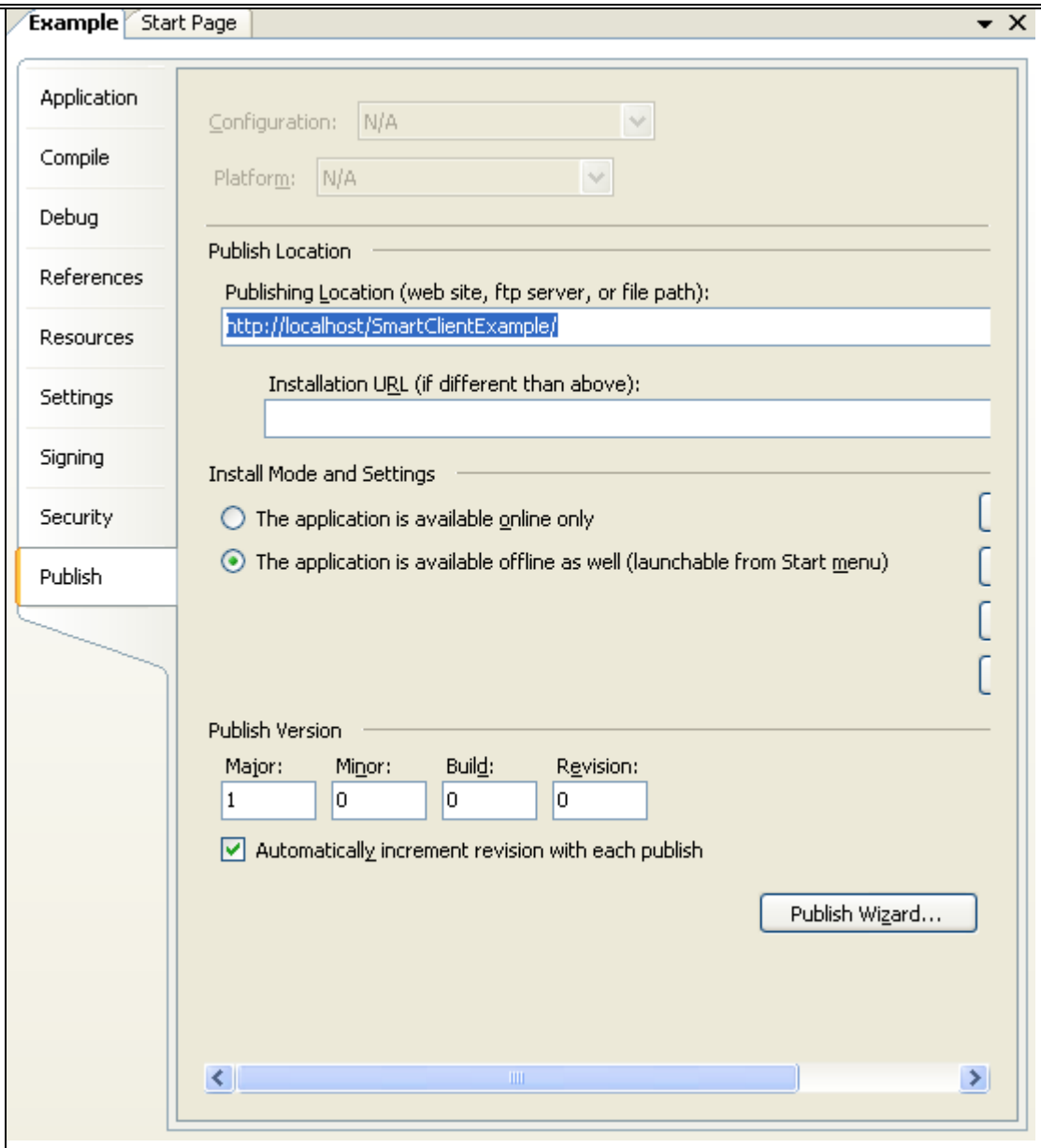
When building your own Smart Client application make sure to include either the Network Node name or IP Address of the OPC Systems Services you wish to connect to. This would include the OPC Controls Tag properties, the OPC Trend pen Tag names, the OPC Alarm window network nodes, and any routines using the OPC Systems programmatic interface.

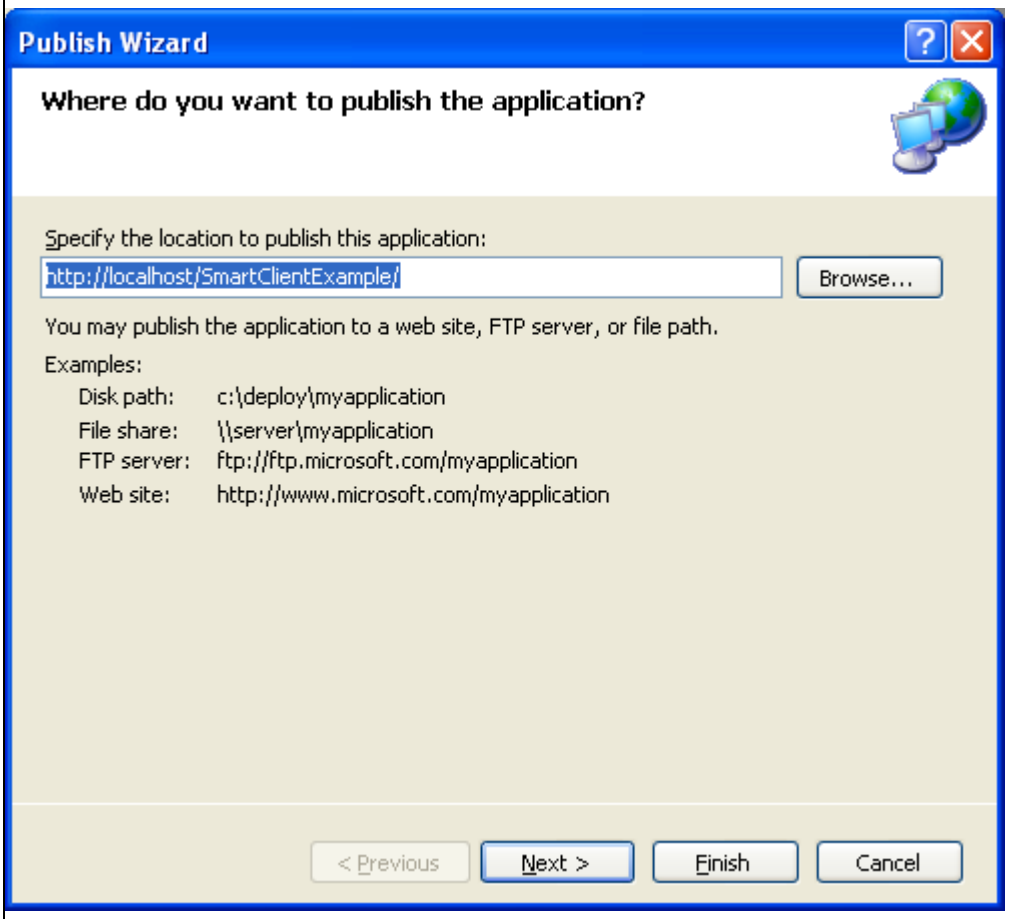
OPC Systems.NET uses TCP port number 58723, so be sure to add your applications you deploy to any firewalls on your system with this TCP port number as an exception.

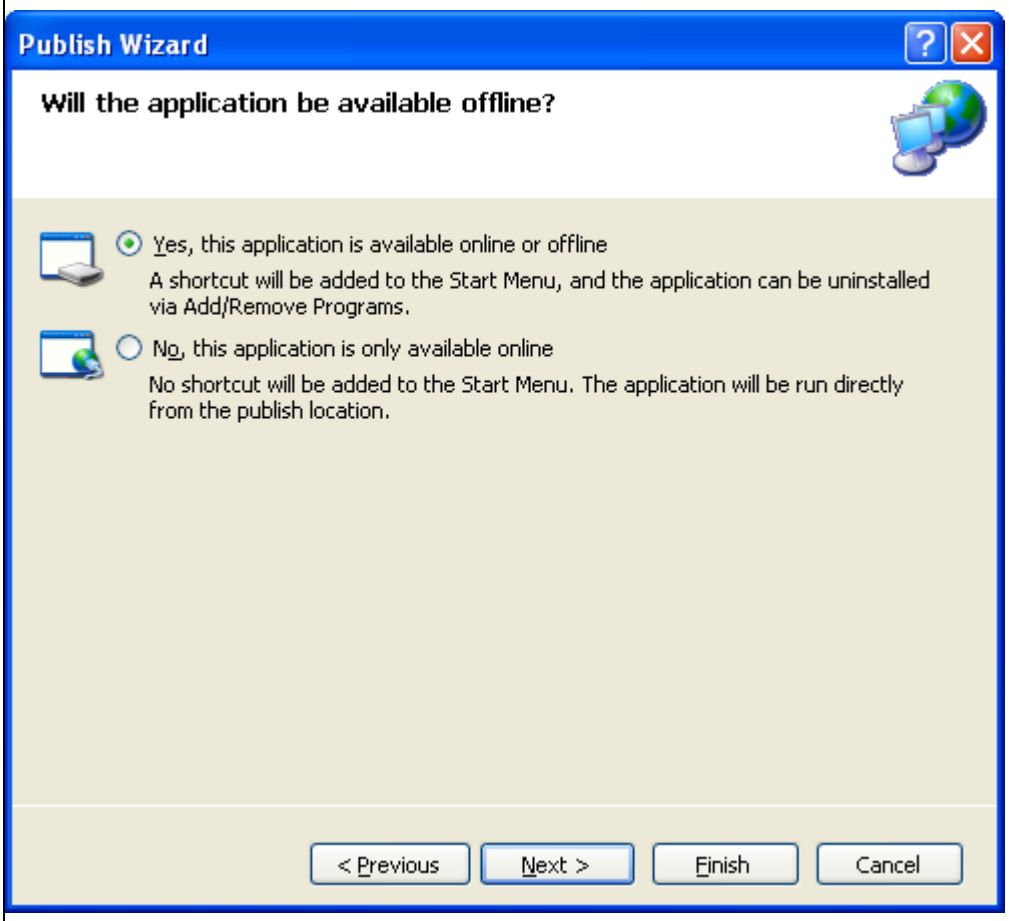
Smart Client Click Once Deployment

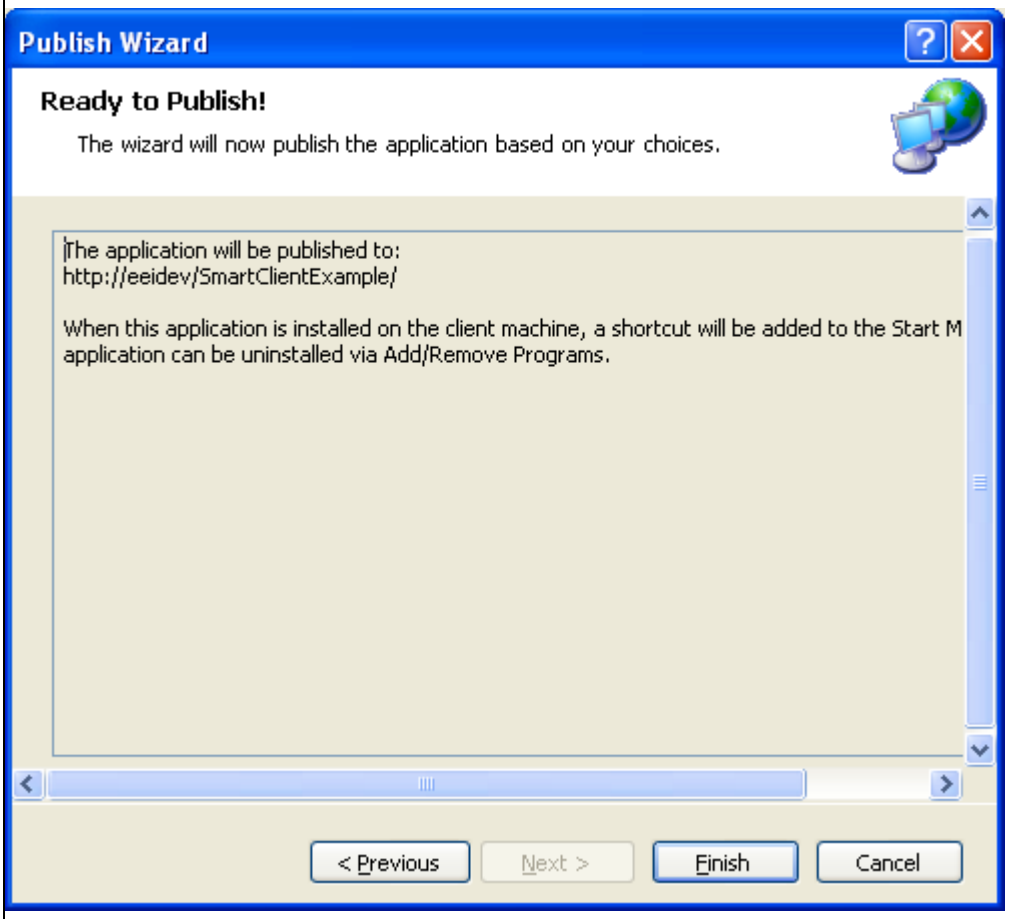
To deploy applications with Visual Studio 2005 with the .NET Framework 2.0 is extremely simple.

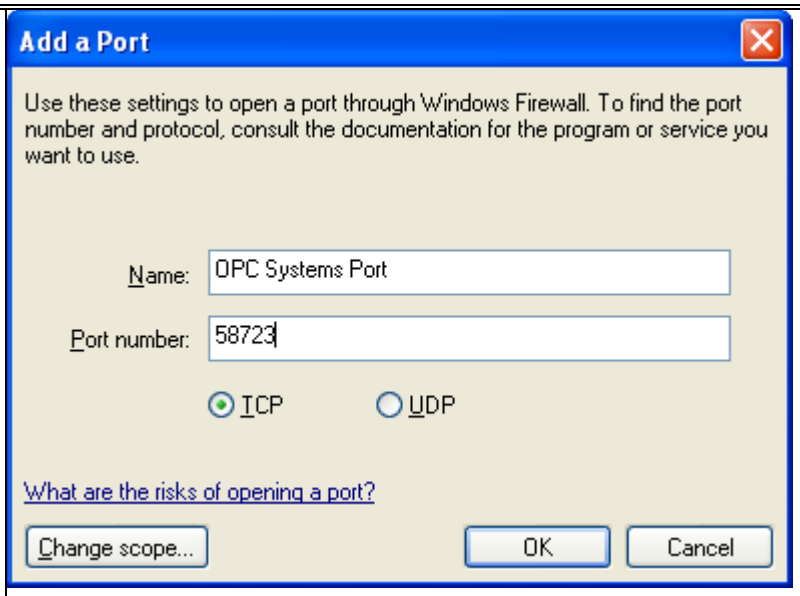
Step	Task
1	Make sure IIS is installed and running on the system you wish to deploy to.
2	In Visual Studio 2005 or 2008 select the Properties of the project you wish to deploy. Next select the Publish tab to view the Click Once deployment section.

	
3	Set the Publish Location to the local or remote system that is running IIS.
4	Set the initial Publish Version and either leave the automatic increment or disable the feature and set it manually each time you want to make an update to the application.

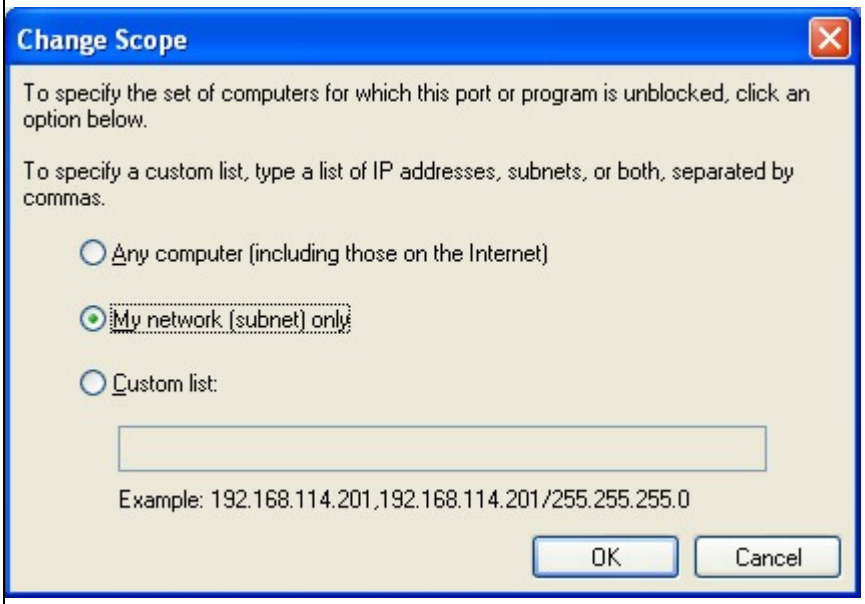
5	Select the Publish Wizard button to begin the steps of deploying the application. 

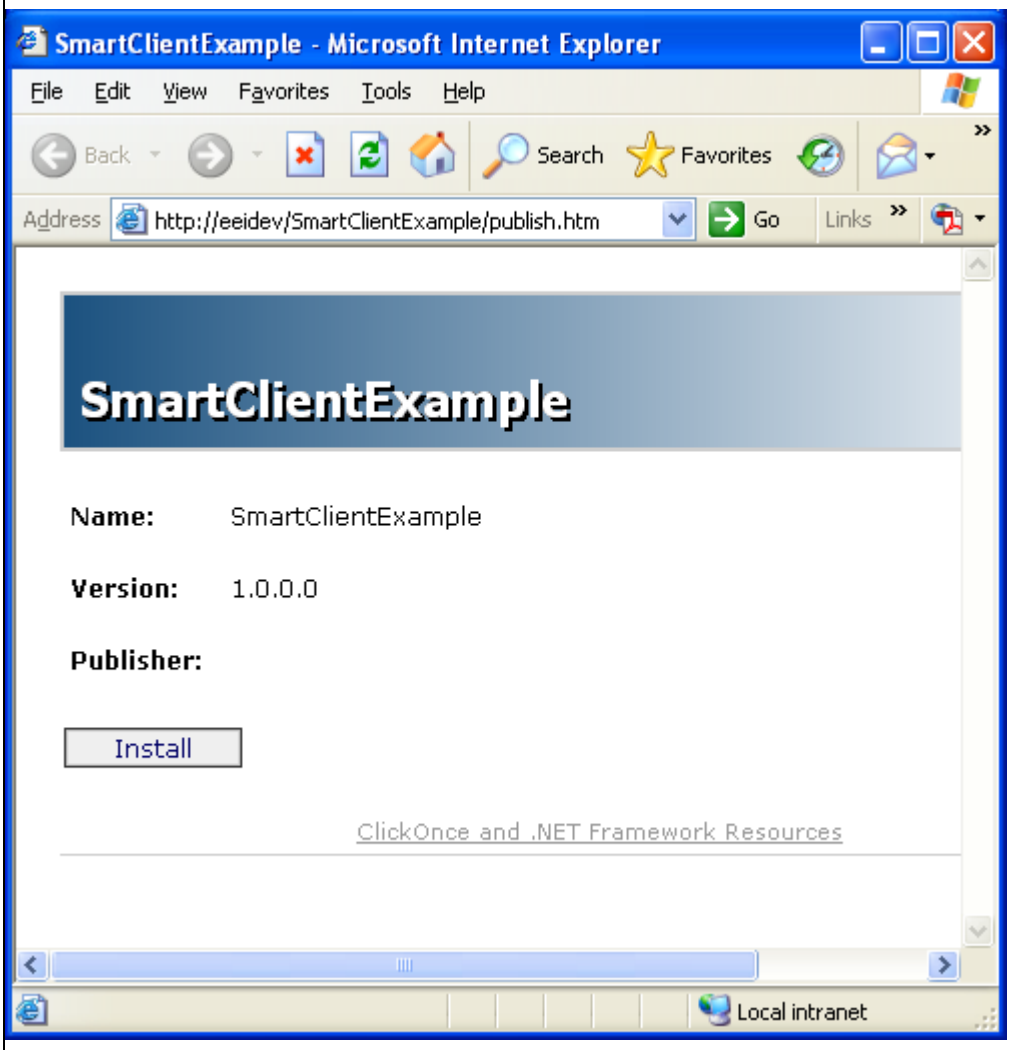
6	Determine if the application will be installed remotely to run on each system even if the server is not reachable or if the application will only be available with the server is online. 

7	Confirm that the deployment virtual directory is correct and select Finish. 
8	Open port 58723 on all systems that will be communicating together that have a firewall enabled. On each OPC Systems Service system that you wish to connect to and each remote system you will be running the application from add TCP port number 58723 if a firewall is enabled. For Windows XP SP2 Select Control Panel-Windows Firewall. Select the Add Port button. Enter a Name and the Port Number 58723.



Use the Change scope button to limit the IP range to the systems that will be communicating together.



9	From any remote client system simply open url of the deployment server publish page using a web browser and select the Install button.  With Smart Client deployment you can now easily deploy updates to one location and make updates just to the one deployment server and all remote users automatically receive the updates on the next time they run the system. When the application is installed for offline/online deployment each time it runs it connects to the deployment server to see if there is a newer version available. The user is prompted to install the update or ignore and continue to run the application.