



TRAINING GUIDE FOR
OPC SYSTEMS.NET

Simple steps to successful
development and deployment.

OPC Systems.NET 入門範例



研杰科技股份有限公司
高雄市左營區民族一路1040之19號
Tel: 07-3452268、02-26550366
<http://www.youngtec.com.tw>

索引目錄

第一章	產品概觀與安裝	4
	OPC Systems Service	4
	Client Components	4
	啓動 Service	4
	Service Logon	6
第二章	Tags	12
	設定 Tags	12
	Calculations	16
	DirectOPC 界面.....	19
	CSV 匯出與匯入	22
	Tags 程式設計界面	22
第三章	Data Logging	23
	設定 Data Logging	23
	從 Visual Studio 應用程式中進行高速資料記錄	30
	Data Logging CSV 匯出和匯入	32
	Data Logging 程式開發界面	32
第四章	Trend	33
	Trending Components	33
	OPC Systems HMI	33
	設定 Trend	34
	在 OPC Systems HMI 中執行 Trend Window	38
	新增 Trend 到 Visual Studio 應用程式	45
	新增 Trend 到 ASP.NET Web 應用程式	51
	新增 Trend ActiveX 控制項到過時的應用程式	60
第五章	Alarming	66
	設定 Alarm Logging	66

	設定 Alarm Notification	70
	Alarm Components	76
	OPC Systems HMI	76
	設定 Alarm Window	77
	執行 Alarm Window	79
	新增 Alarm Window 到 Visual Studio 應用程式	82
	新增 Alarm Window 到 ASP.NET Web 應用程式	89
	新增 Alarm ActiveX 控制項到過時的應用程式	98
第六章	Human Machine Interface	102
	HMI Components	102
	新增 HMI Controls 到 Visual Studio 應用程式	102
	新增 HMI Controls 到 ASP.NET Web 應用程式	107
	新增 Data Control 到 Visual Studio 應用程式	114
第七章	Reports	122
	設定 Reports	122
第八章	設定 Recipes	134
	Recipe CSV 匯出和匯入	142
	Recipe 程式開發界面	143
第九章	設定 Security	144
第十章	預設設定	149
	設定 Options	149
第十一章	Smart Client 開發	151
	Smart Client 範例	151
	Smart Client Click Once 開發	152

第一章 產品概觀與安裝

OPC Systems Service

OPC Systems.NET 使用一個集中式的即時資料庫，作為所有 Client 端程式的資料來源。此即時資料庫執行成 Windows Service，且可以被佈建至多部電腦上，以建立一全企業或是全球性的分散式資料庫。儲存在每個服務中的資訊可包含 Tags、Data Logging Ggroups、Alarm Logging Groups、Alarm Notification Groups、Recipes、Reports、Security Groups、Security Users 和 Systems Options。這些服務可以下列三種方式，從本機或是遠端來進行設定。

1. OPC Systems Configuration Application。
2. CSV 匯入和匯出。
3. 使用 OPC Systems.NET 程式開發界面。

一旦設定檔建立起來之後，可被儲存成二進位檔。使用 Configure-Options 功能，可設定成當服務啟動的時候自動地載入設定檔。

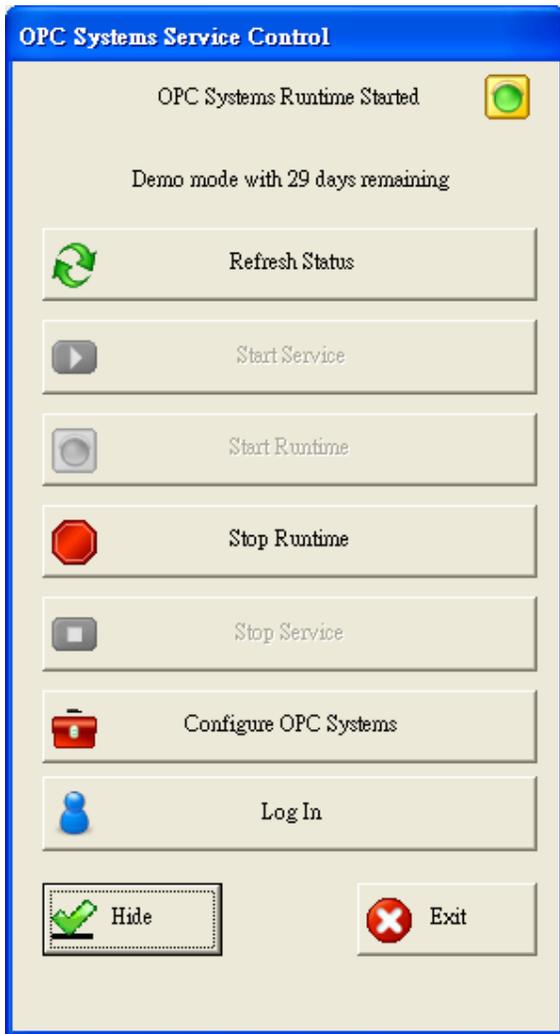
Tags 的資料來源可以是 OPC Servers、OPC Clients、.NET 應用程式，或資料庫，如 SQL Server、Oracle、Access 和 MySQL。你也可以使用 Client Component 的 DirectOPC 介面來直接連接 OPC Server，而不必事先建立 Tags。所有的界面支援 .NET 通訊，所以透過 Internet 的遠端通訊也能夠容易地達成。

Client Components

即時視覺、資料存取、趨勢、警報及歷史資料存取，都可以使用 Client Component 來達成。所有的 Component 使用 .NET 通訊來與 OPC Systems Service 進行溝通。此做法使得它可以透過 Internet 來顯示和控制即時資料，而沒有設定 DCOM 的必要。所有的 .NET Component 都是 100% Managed，所以它們都可以被使用在 Smart Client 和 ASP.NET Web 應用程式，而不必使用到 COM 或 ActiveX 等過時技術。但也提供有趨勢和警報的 ActiveX 來整合過時的產品。

啓動 Service

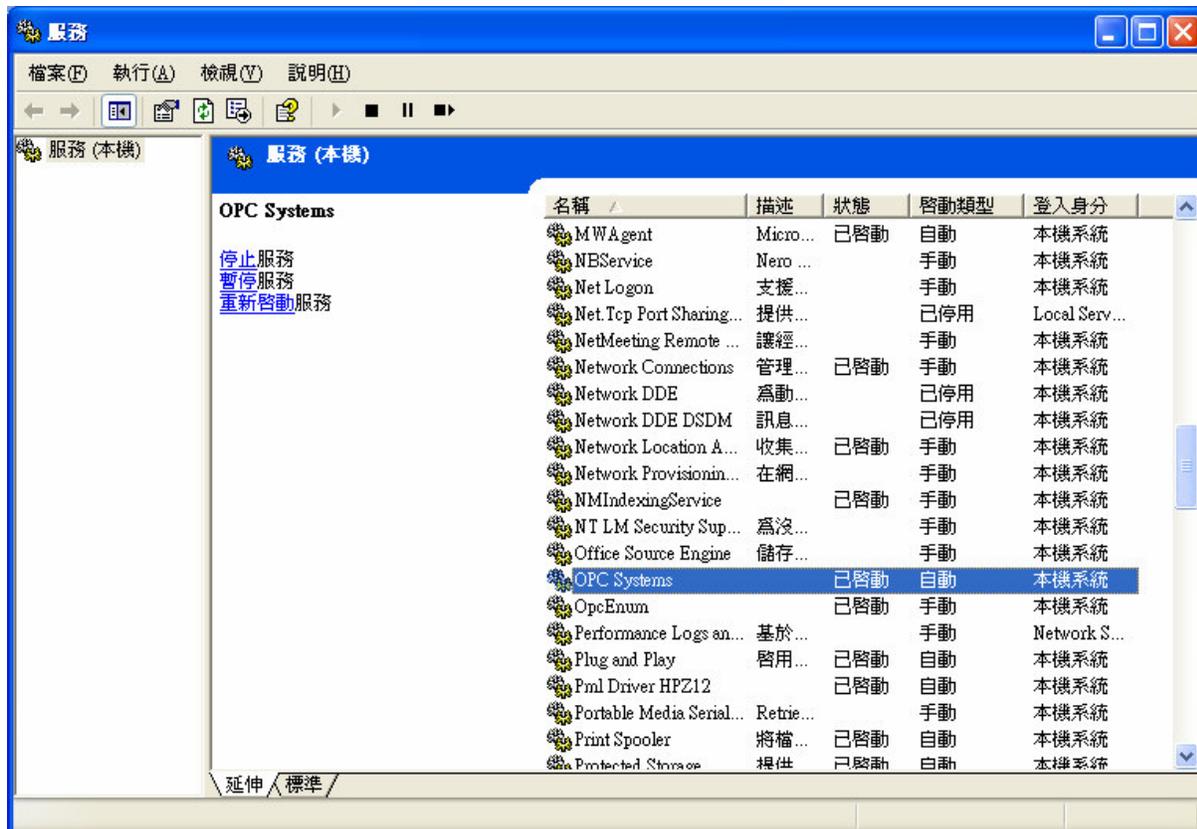
OPC Systems Service Control Manager 會在安裝完畢之後自動出現。使用 OPC Systems Service Control 應用程式可啓動停止 OPC Systems Service。



按下 **Hide** 按鈕可以隱藏 **OPC Systems Service Control Manager** 在右下方的 **System Tray** 中，它會顯示成一個藍色的圓形小圖示。欲再從 **System Tray** 中開啓 **OPC Systems Service Control Manager**，則只要在藍色的圓形小圖示上按滑鼠右鍵，並在快捷功能表上點選 **OPC Systems Service** 即可。欲結束此程式之執行，則請按下 **Exit** 按鈕。

你可以選擇性地在這裡啓動 **Runtime** 或以後再啓動。除了載入 **Tags** 的設定檔，所有其他的修正都可以在 **Runtime** 中進行。你可以在 **Runtime** 中新增、刪除和修改 **Tags**、**Data Logging Groups**、**Alarm Logging Groups**、**Alarm Notification Group**、**Recipes**、**Reports**、**Security Groups** 和 **Security Users**。

你也可以使用服務管理員來啟動和停止 OPC Systems Service。

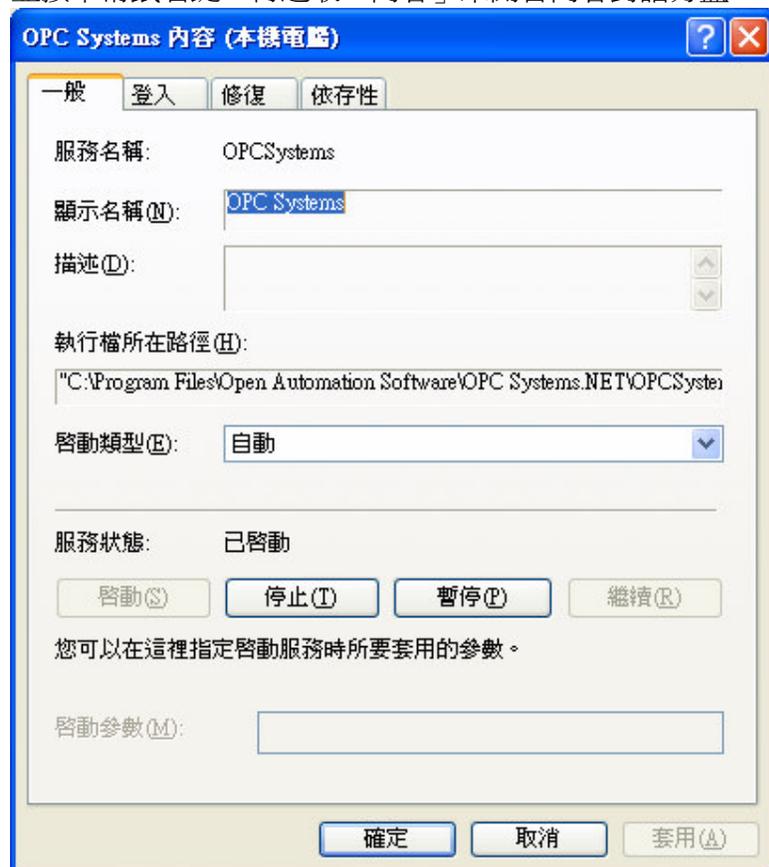


Service Logon

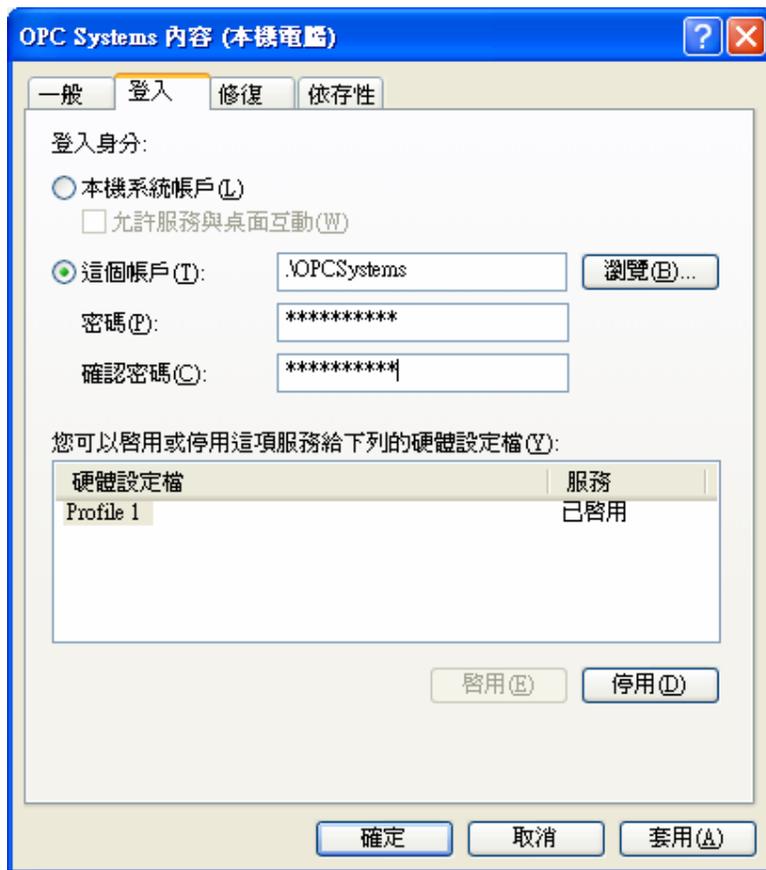
OPC Systems Service 可以被設定成以 LocalSystem 帳號，或是以某一使用者帳號來啟動。如果你從 OPC Server 接收到 bad data quality 的訊息，則有可能是起因於作業系統的帳號權限制。為解決此問題，建議設定成某一使用者帳號來啟動，以方便通過 DCOM 的權限制。

請參照以下做法進行登入帳戶變更作業：

開啓服務管理員，找到 OPC Systems 之後，在其上連按滑鼠左鍵二下，或是點選 OPC Systems 並按下滑鼠右鍵，再選取「內容」來開啓內容對話方盒。

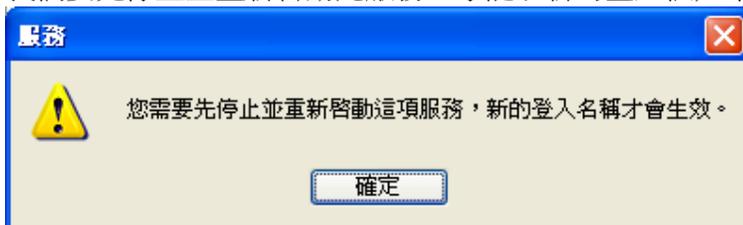


接著選擇「登入」頁籤，將登入身分從「本機系統帳戶」改選成「這個帳戶」，並且按下最右邊的  按鈕，選取一預先建立且具有管理者群組權限的帳號，如本例中是為 OPCSystems。並且必須在「密碼」和「確認密碼」欄位中依序鍵入此帳號之密碼。

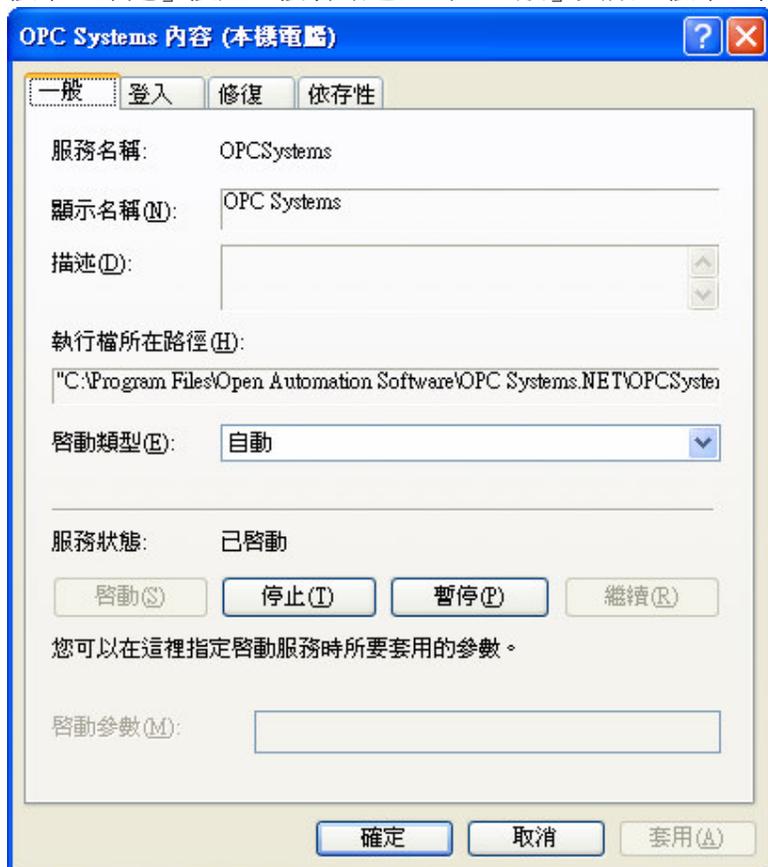




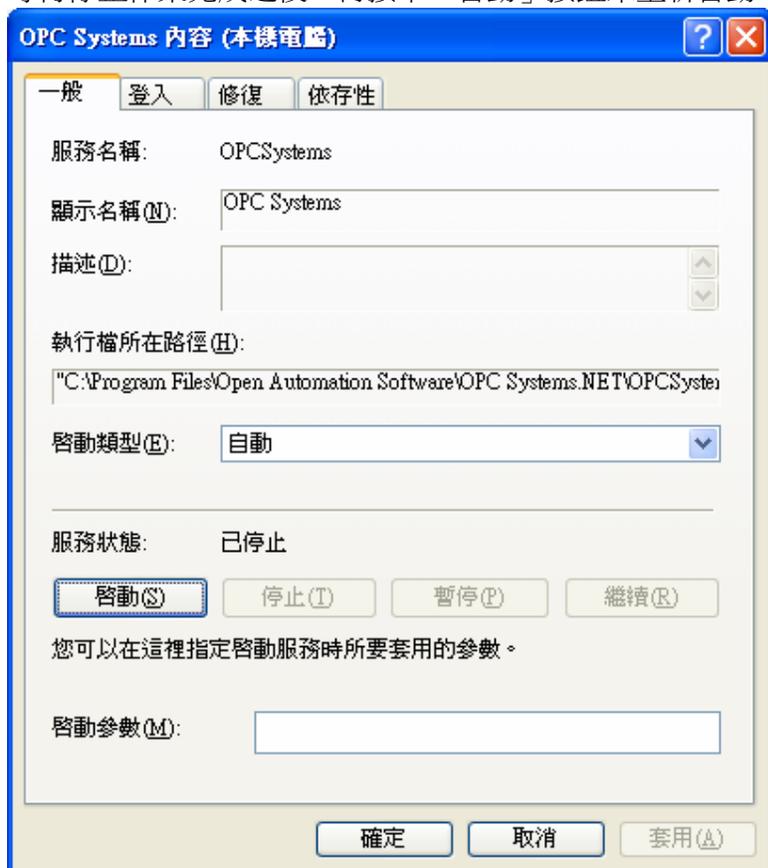
完成後按下此對話方盒右下方的「套用」按鈕，如正確無誤，則會出現以下之對話方盒來提醒我們要先停止並重新啓動此服務，才能以新的登入帳戶來啓動。



按下「確定」按鈕。接著點選回到「一般」頁籤，按下「停止」按鈕。



等待停止作業完成之後，再按下「啟動」按鈕來重新啟動 OPC Systems 服務。



重新啓動成功之後，即可按下「確定」按鈕來完成此設定作業。

第二章 Tags

設定 Tags

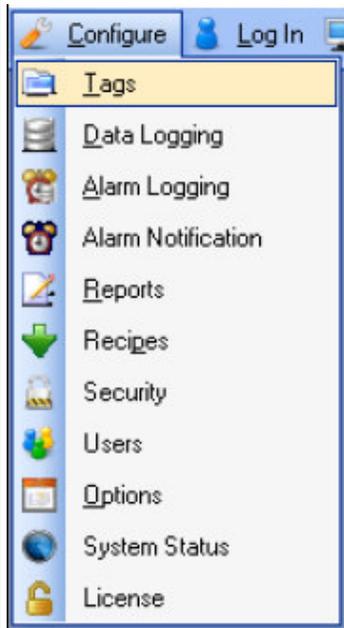
Tags 是用來定義資料來源、Alarm 的高低限、和其它的即時訊號屬性，例如使用「Enable Time On and Counts」來追蹤某一個 Tag 持續 On 多久，和在某一段時間之內變化多少次。Tags 是所有 Client 應用程式的共同資料來源。如欲接受直接來自 Visual Studio 應用程式的資料，使用 Value 為資料來源。其值可以使用 OPC Controls.NET Data Component 來改變。資料來源也可以是 Calculation，其值是本機或遠端 Tags 的運算結果。

不建立 Tag，直接由 Client Component 連接至 OPC Server 的 Items，也是可行的。所以你如果只需要以直接 OPC 為資料來源，而不需其它如 Value 和 Calculation 等，則此設定 Tags 的步驟也可不必進行。但建議至少建立一個 OPC Systems.NET Tag 來測試與 OPC Server 的通訊，以及其它種 Tag 能夠執行的功能。

以下是介紹如何使用 Configure OPC Systems 應用程式來進行增加與定義 Tags 的步驟。也可以 CSV 檔匯出與匯入的方式來進行 Tags 的新增和修改作業。

欲學習如何以 Visual Studio 應用程式來進行 Tags 的新增和修改作業，請參考 Example Code 中的 VB.NET Example - FormConfigureCSV，來了解以何 Method 來進行新增 Tag 和同時定義多個 Tags。

Step	Task
1	啓動 Configure OPC Systems 應用程式。 
2	選擇 Configure → Tags。



3 按下 **Select** 按鈕或點選畫面左方的 **Local** 來選擇本機 OPC Systems Service。



注：

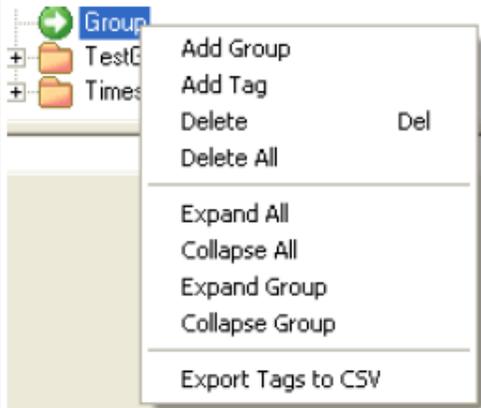
Configure 程式可以用來連接遠端系統。點選 **Network Node** 中的電腦名稱，或是在 **NetworkNode** 輸入方塊中鍵入電腦名稱或是 IP 位址，來連接遠端執行 OPC Systems Service 的電腦。

4 在 **Local OPC Systems Service** 上按滑鼠右鍵並選擇 **Add Tag**。



注：

你可以依照需要增加多層次群組，並在其中加入 Tags。其做法是在 **Root** 層新增一群組，再點選該新增出之群組並按滑鼠右鍵，選取 **Add Group** 或 **Add Tag**。



5 在 Add Tag dialog box 中鍵入 Tag name，並按下 OK 按鈕。



6 以 Sine 為 Tag name，重覆步驟 4 和 5。

7 以 Random 為 Tag name，重覆步驟 4 和 5。

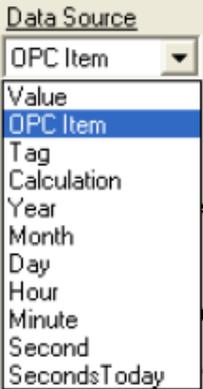
8 選擇 Ramp Tag。



所有的 Tag 屬性會出現在下方窗格。



9 在 Value 的部份，選擇 OPC Item 為資料來源。

	
10	<p>使用下方窗格右方的 OPC Browse 按鈕來瀏覽 OPC Servers 。</p> 
11	<p>在 Browse OPC Servers 對話方塊中展開 Local 及之下的 EEI.OPCSimulator 並選擇 SimDevice 。</p> 
12	<p>從 OPC Items 中選取 Ramp 並按下 OK 來輸入 OPC Item: EEI.OPCSimulator\SimDevice.Ramp 。</p>
13	<p>勾選 Trend Point 選項。使得該 Tag 的值可被 Trend 功能使用。</p> 
14	<p>輸入 Ramp 到 Description 欄位。</p>
15	<p>按下 High High 按鈕並設 Value 欄位為 80 。</p>
16	<p>按下 High 按鈕並設 Value 欄位為 60 。</p>
17	<p>按下 Low 按鈕並設 Value 欄位為 40 。</p>
18	<p>按下 Low 按鈕並設 Value 欄位為 20 。</p>
19	<p>按下右下方的 Apply Changes 按鈕。</p> 
20	<p>點選 Random 和按下 Value 按鈕。</p>  <p>接著以 Random 取代 Ramp，重覆步驟 8 到 14 和 19。OPC Item 為 EEI.OPCSimulator\SimDevice.Random 。</p>

21 點選 Sine 並按下 Value 按鈕。

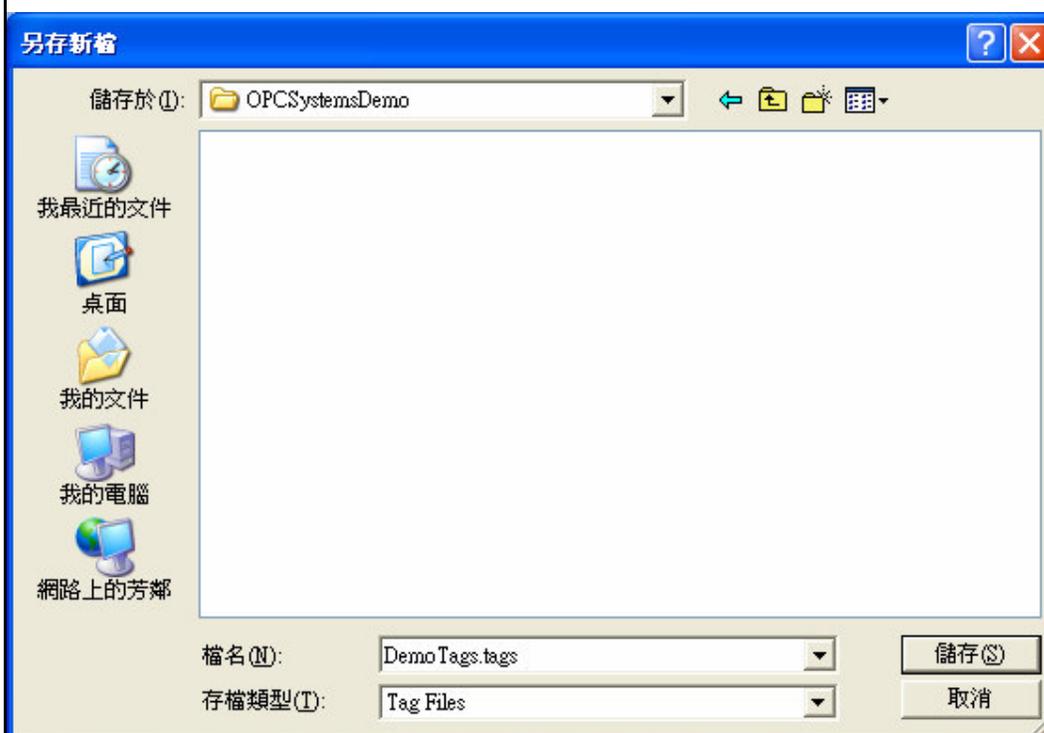


以 Sine 取代 Ramp，重覆步驟 9 到 19。使用 0.9 為 High High 值，0.8 為 High 值，0.2 為 Low 值，和 0.1 為 Low Low 值。OPC Item 為 EEI.OPCSimulator\SimDevice.Sine。

22 按下上方工具列的  Save 按鈕來儲存剛才之設定。

23 在 C:\ 之下建立 OPCSystemsDemo 目錄。

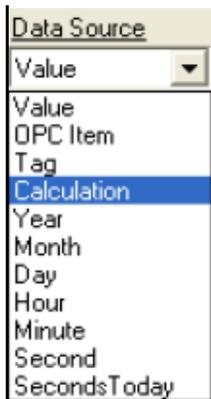
儲存檔案 DemoTags.tags 在 C:\OPCSystemsDemo 目錄之下。



Calculations

你可以定義 Tag 來自動處理既有本機和遠端 Tag 資料來源之數學運算。

Step	Task
1	啟動 Configure-Tags 來建立一個名稱為 Total 的新 Tag。
2	設定這個 Tag – Total 的資料來源為 Calculation。

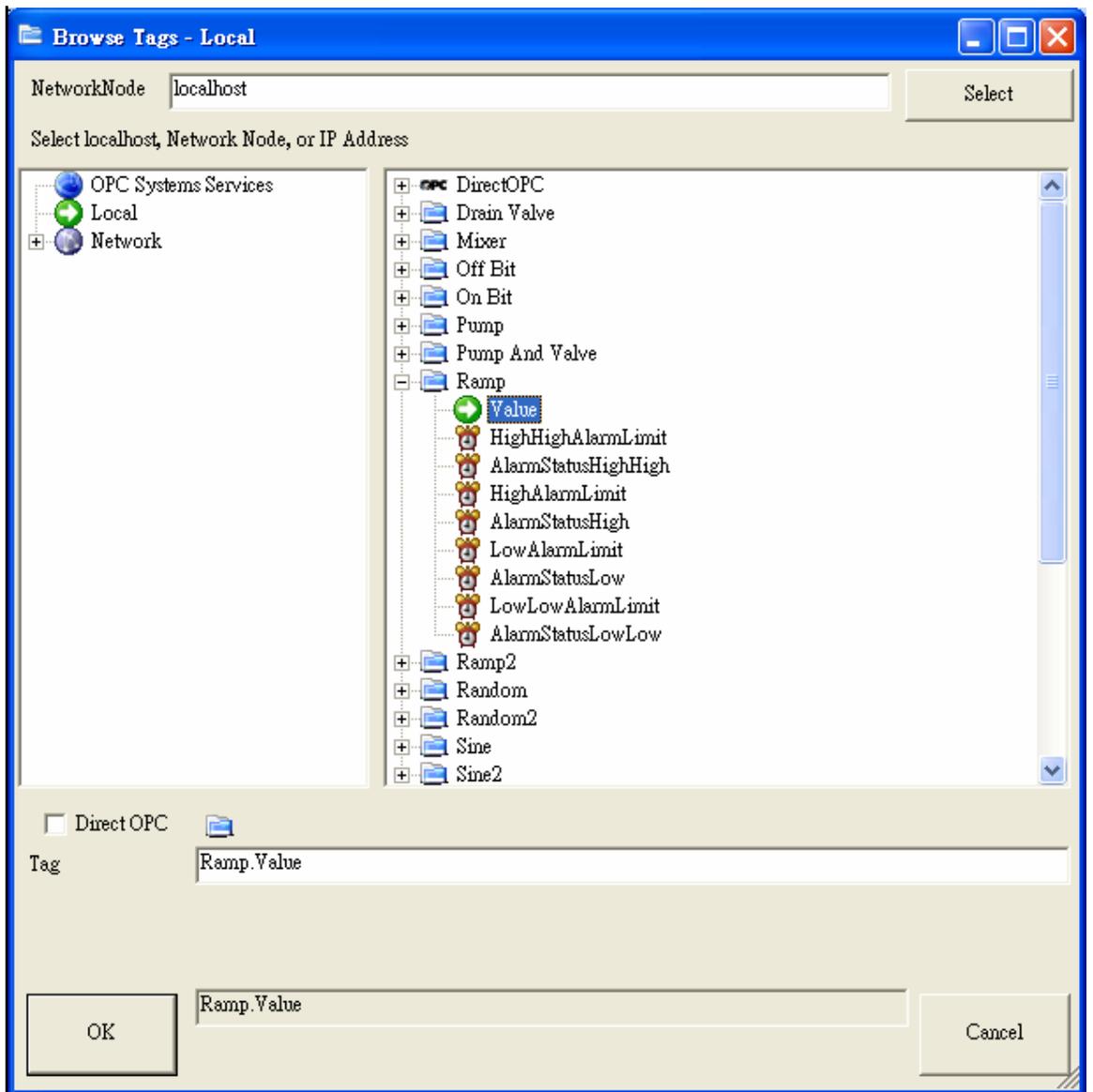


3 使用位於右方的 **Edit** 按鈕來開啓 **Calculation Edit** 對話方盒。

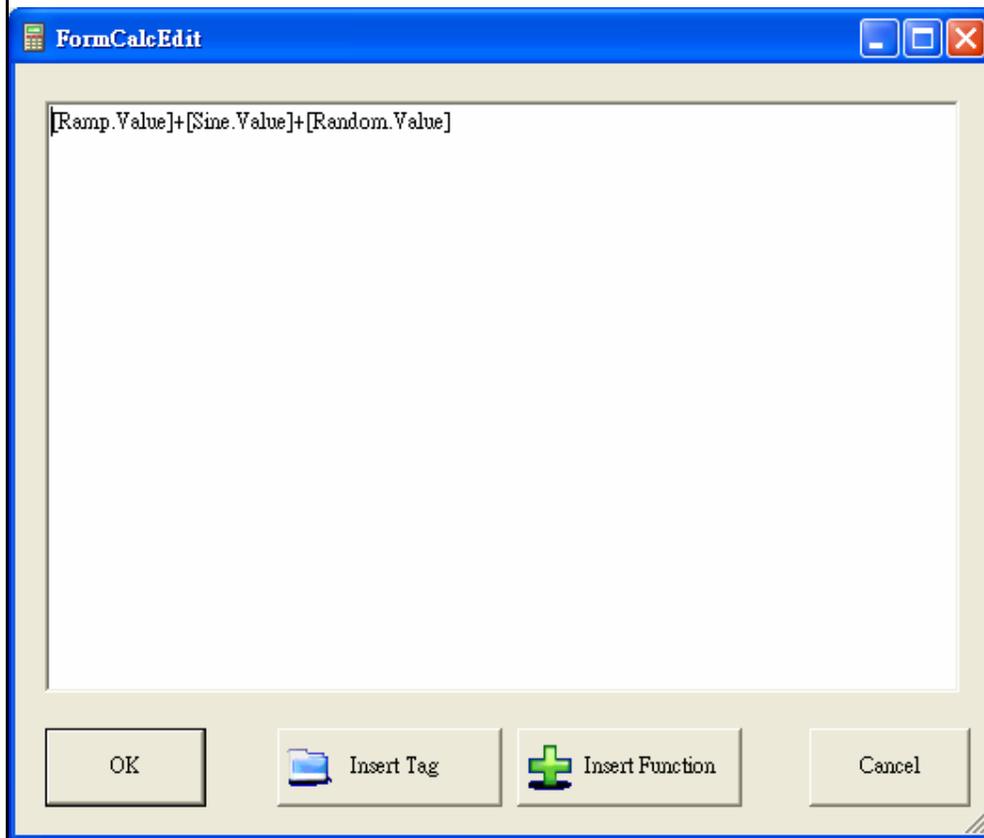


按下 **Insert Tag** 按鈕並選取 **Ramp.Value**。





在[Ramp.Value]後面加入+號，再加上 Sine.Value。在[Sine.Value]後面加入+號，再加上 Random.Value。此方程式最終為[Ramp.Value]+ [Sine.Value]+[Random.Value]。



按下 OK 按鈕，再按下 **Apply Changes** 按鈕，此時將可看到 Total 的值為該三個 Tag 值的加總。

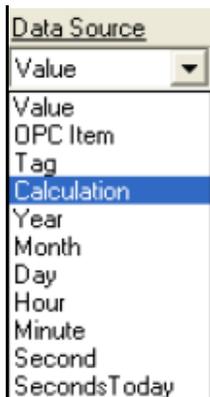
注：

若是 Calculation 中任一個參考到的 Tag 之值為 bad，則此 Calculation Tag 之值同樣是為 bad。

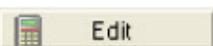
DirectOPC 界面

所有使用 OPC Systems.NET Tags 為資料來源的 Client Component，都可以透過 DirectOPC 界面直接使用從 OPC Server 來的值。

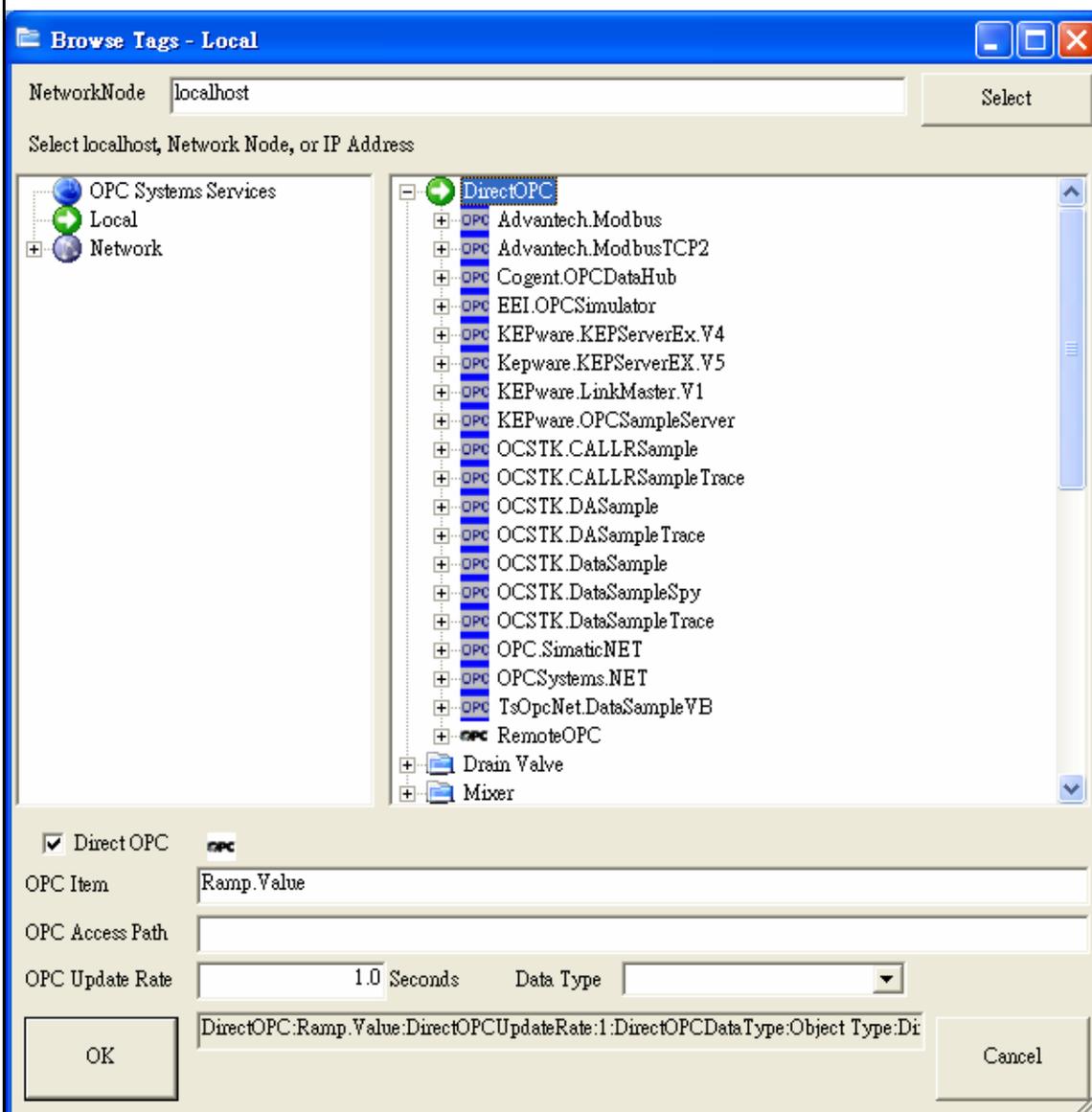
Step	Task
1	啟動 Configure-Tags 來建立一個名稱為 CalcDirectOPC 的新 Tag。
2	設定這個 Tag – CalcDirectOPC 的資料來源為 Calculation 。



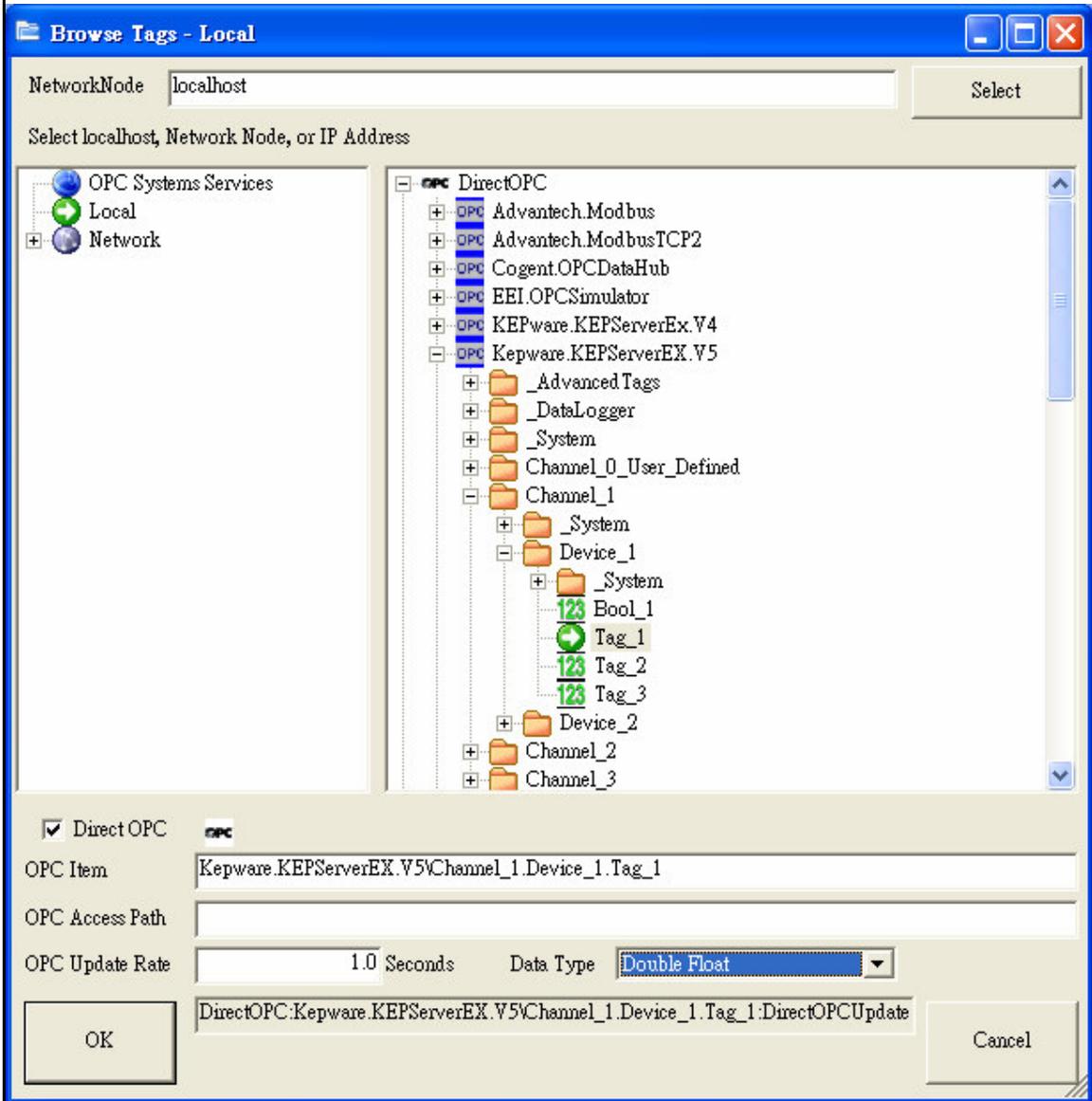
3 使用位於右方的 **Edit** 按鈕來開啓 **Calculation Edit** 對話方盒。



按下 **Insert Tag** 按鈕並點選 **DirectOPC** 旁的加號(+)來展開並瀏覽該部電腦上的 **OPC Servers**。



瀏覽並找到你欲連接的 OPC Server Item，並設定 Update Rate 和資料型態，如下圖所示。



按下OK按鈕之後，可看到其最終的Tag名稱爲

[DirectOPC:Kepware.KEPSEServerEX.V5\Channel_1.Device_1.Tag_1:DirectOPCUpdate Rate:1:DirectOPCDataType:Double Float:DirectOPCAccessPath:]。

OPC Server : Kepware.KEPSEServerEX.V5

OPC Item : Channel_1.Device_1.Tag_1

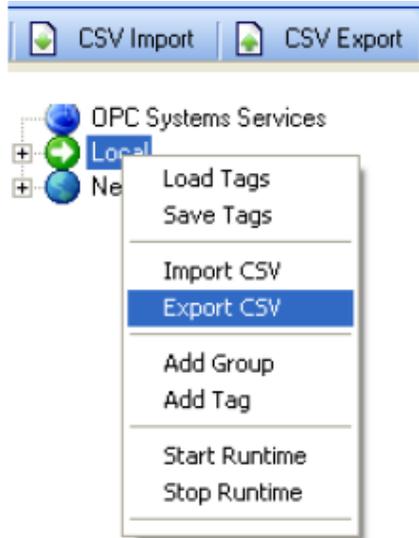
OPC UpdateRate :1

Data Type : Double Float

所有 Client Component 都可以使用此語法來直接連接 OPC Server。

CSV 匯出與匯入

Tags 可以被匯出成 CSV 檔，利用 Excel 來編輯。點選 Local Service 或是欲管理且具權限之電腦名稱，然後按下滑鼠右鍵並選擇 Export CSV，或是按下上方工具列之 Export CSV 按鈕。



以 Microsoft Excel 開啓儲存起來之後的 CSV 檔並編輯。其內只有 Tag 欄位是必要的，其它欄位是可有可無的。如果某個屬性欄位被刪除，則新的 Tag 會被建立，且具有預設值屬性會自動產生。

完成 Microsoft Excel 的編輯並存檔之後，相對的使用 Import CSV 來將編輯過的 CSV 檔匯入，然後以 Configure-Tags 的第 22 步驟來進行儲存設定檔動作。

注：

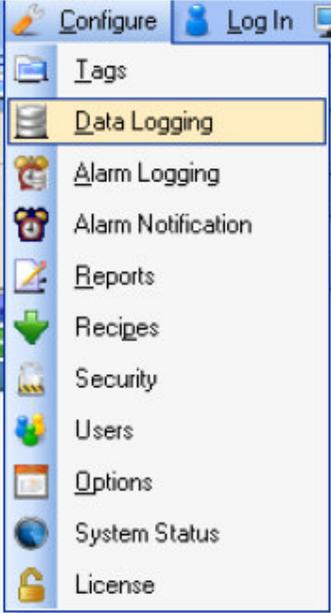
當 CSV 檔正被 Microsoft Excel 開啓時，無法進行匯出和匯入作業。欲進行匯出和匯入作業時，請記得將正被 Microsoft Excel 開啓的 CSV 檔關閉。

Tags 程式設計界面

使用 OPC Systems Component 在你的 Visual Studio 應用程式中，可以利用程式來進行 Tags 的編修作業。

第三章 Data Logging

設定 Data Logging

Step	Task
1	<p>啓動 Configure OPC Systems 應用程式，如果它尚未啓動的話。</p> 
2	<p>選擇 Configure → Data Logging。</p> 
3	<p>按下 Select 按鈕或點選畫面左方的 Local 來選擇本機 OPC Systems Service。</p>  <p>注：</p> <p>Configure 程式可以用來連接遠端系統。點選 Network Node 中的電腦名稱，或是在 NetworkNode 輸入方塊中鍵入電腦名稱或是 IP 位址，來連接遠端執行 OPC Systems Service 的電腦。</p>
4	<p>將 Simulation 鍵入右上方的 Logging Group Name 欄位中。</p> 

5 點選 Common 頁籤並勾選 Logging Active 選項。

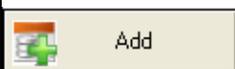


我們將使用預設的記錄方式，以一秒鐘的時間間隔持續儲存。

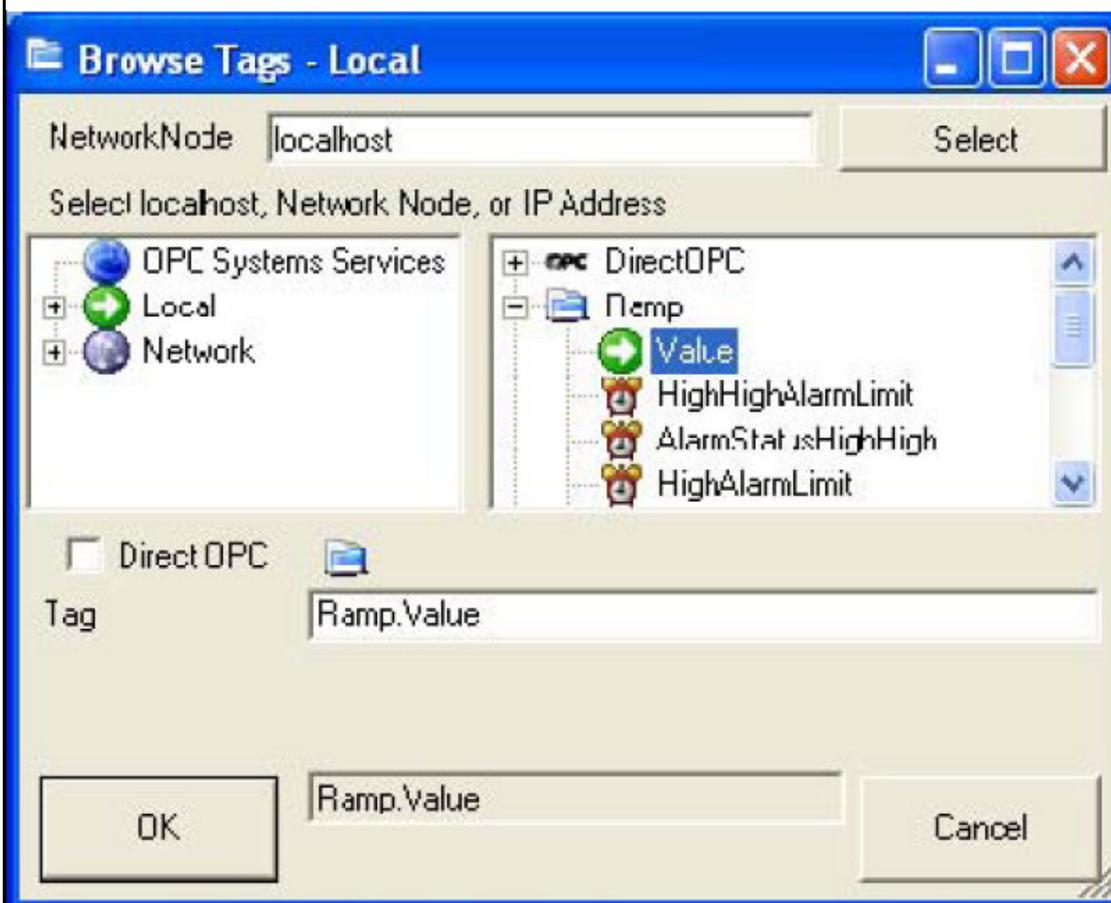
6 點選 Tags 頁籤。



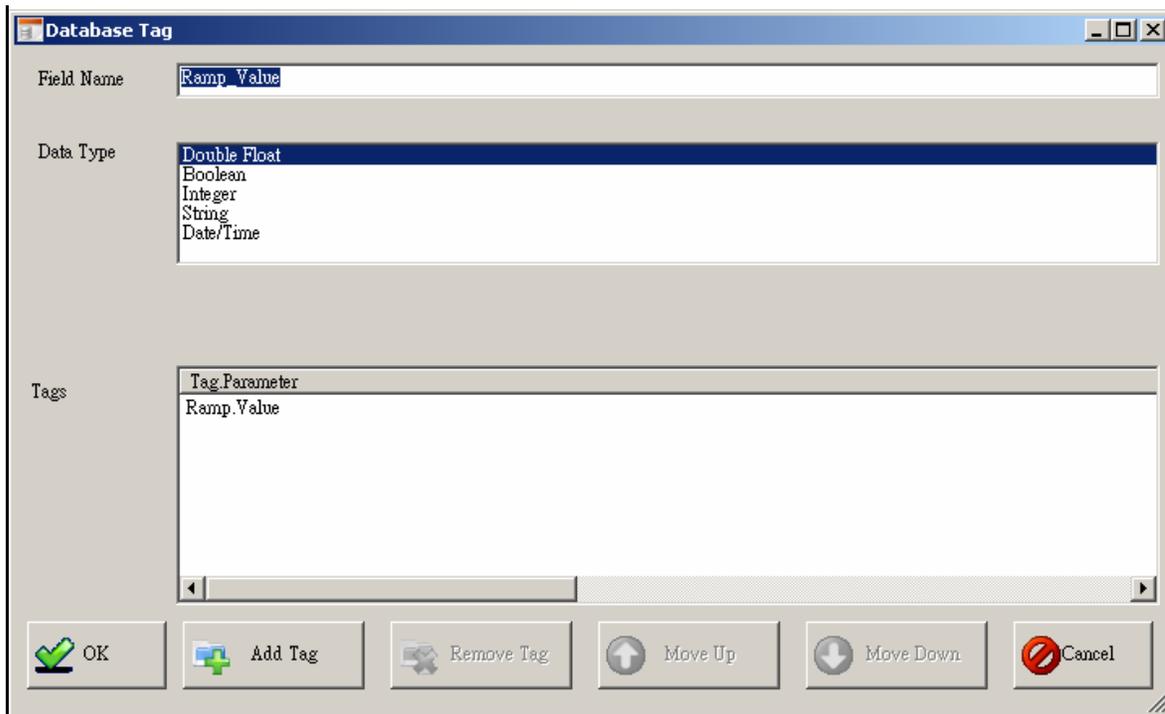
7 按下 Add 按鈕。



8 在 Browse Tag and Parameter 對話方塊中選取 Ramp.Value，按下 OK 按鈕。



按下 OK 按鈕。



選取 Double Float 為 Data Type 並按下 OK 按鈕。

- 9 為 Random.Value 參數重覆步驟 7 和 8。
 為 Sine.Value 參數重覆步驟 7 和 8。

Field Name	Data Type	Tag.Parameter
Ramp_Value	Double	Ramp.Value
Random_Value	Double	Random.Value
Sine_Value	Double	Sine.Value

- 10 點選 Database 頁籤。



- 11 使用下列設定來將資料儲存進 SQL Server。Database 和 Table 將會自動被建立。

Logging Group Name: Simulation

Common  Tags  Database  CSV Logging

Log To Database 

Provider: SQLServer

Server: localhost

Set Server Name with Tag



Database: OPCSystemsDemo

Set Database Name with Tag

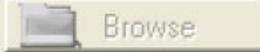


Table: SimulationTable

Set Table Name with Tag



Use WinNT Authentication

User Name: _____

Password: _____

Log Only One Record

 Add  Delete  Apply Changes

Server 的名稱必須將其從 localhost 調整成適當的 SQL Server engine 名稱。此名稱在第一次啓動 SQL Server Management Studio 的 Connect to Server 對話方盒中必須被找到。



如果你想要使用 Microsoft Access 來儲存資料，則使用下列設定。Database 和 Table 將會自動被建立。

Common Tags Database CSV Logging

Log To Database 

Provider: MSAccess

Server: (local)

Set Server Name with Tag

Database: C:\QPCSystemsDemo\QPCSystemsDemo.mdb

Set Database Name with Tag

Table: SimulationTable

Set Table Name with Tag

Use WinNT Authentication

User Name:

Password:

Log Only One Record

若為 CSV 文字檔，則使用下列設定。但請注意，欲存放資料的資料夾必須已事先建立妥。

Common
 Tags
 Database
 CSV Logging

Log To CSV File 

Path

Set Path with Tag

File Name

Set File Name with Tag

Append Date to File Name 

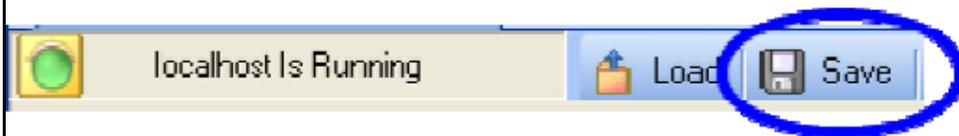
Append Hour to File Name

Append Minute to File Name

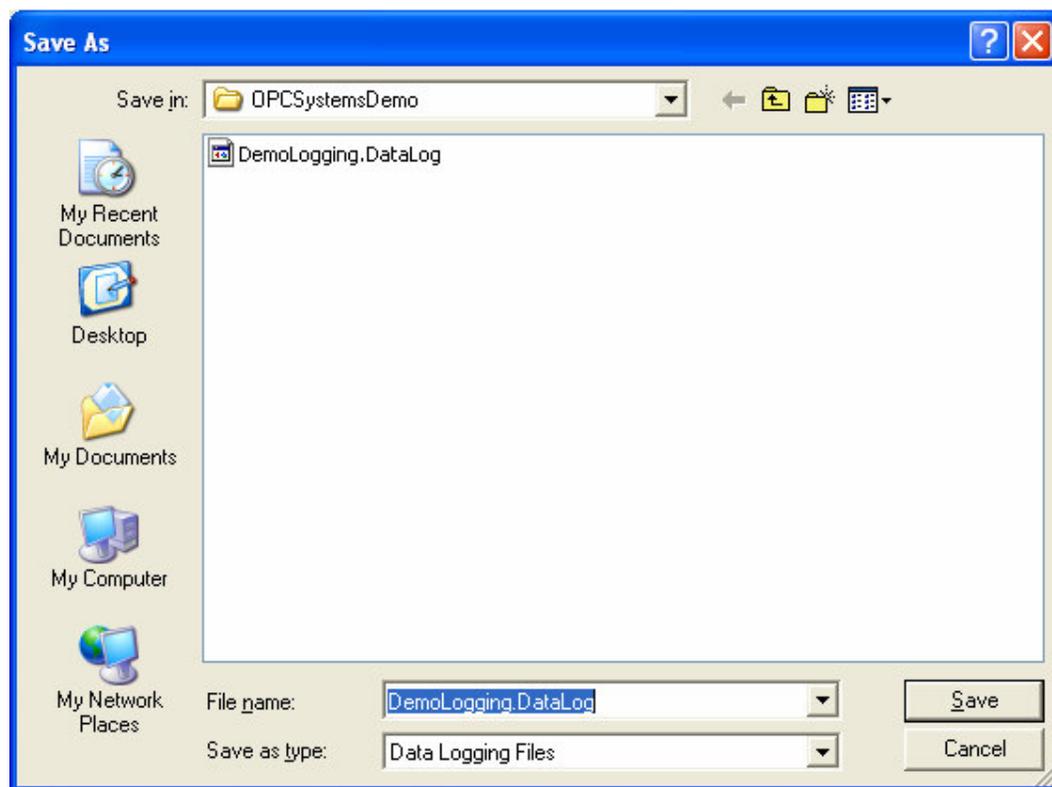
12 按下 Add 按鈕來加入 Data Logging group 。



13 按下上方的  Save 按鈕來儲存剛才之設定。



14 以 DemoLogging.DataLog 名稱儲存到 C:\OPCSystemsDemo 。



參考 OPC Systems.NET help 檔中的 OPC Systems Configuration – Data Logging 部份可得知每個屬性的說明。

從 Visual Studio 應用程式中進行高速資料記錄

OPC Systems.NET 支援從 Visual Studio 應用程式中進行高速資料記錄，速率可達 100 nanoseconds。

為了直接從 Visual Studio 應用程式來記錄資料，我們必須採用展示在 VB.NET 範例 FormWriteValues 中的 OPC Controls.NET Data Component。使用 OPC Controls.NET Data Component 的 WriteTags 方法來寫入 Tags。如果我們希望在 WriteTags 方法中包含 TimeStamps 陣列，OPC Systems.NET Service 將以我們的 TimeStamps 來寫入資料庫，而不是以電腦的時間。

以下是範例程式碼：

```
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)
```

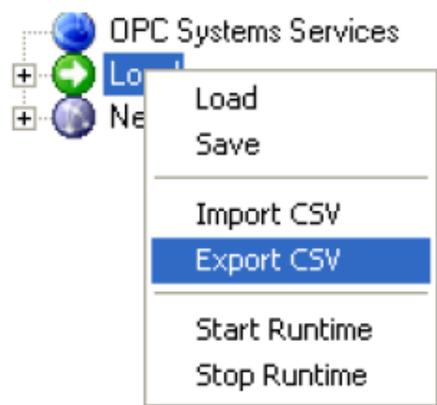
OpcControlsData1.WriteTags(TagsToWrite, ValuesToWrite, TimeStampToWrite)

此 WriteTags 方法的執行結果，將有 3 筆記錄儲存在資料庫中。

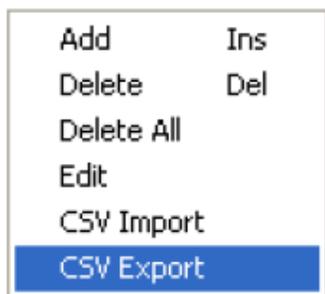
DateAndTime	ms	us	ns	Value01	Value02	Value03
3/10/2009 11:04:50 AM	984	375	0	1	1	1
3/10/2009 11:04:50 AM	984	375	100	2	2	2
3/10/2009 11:04:50 AM	984	375	200	3	3	3

Data Logging CSV 匯出和匯入

在電腦名稱上按下滑鼠右鍵並選擇 Export CSV，即可將所有的 logging group 匯出至 CSV 檔中。



我們可以在 Tags 欄位清單中按下滑鼠右鍵，來匯出某 logging group 中的個別欄位。



Data Logging 程式開發界面

在 Visual Studio 應用程式中使用 OPC Systems component，可以程式來修改 data logging groups 的設定。參考 VB.NET 範例中的 Configure Data Logging Form 和 Configure CSV Form，來得知如何新增和修改 data logging groups。請參照 OPC Systems Component help 檔來得知所有 method 的語法。

第四章 Trend

Trending Components

總共有三種 trending components 可以被使用在即時資料的趨勢呈現：

1. 使用在 WinForm 應用程式中的 OPC Trend.NET 100% Managed component。
2. 使用在 Web 應用程式中的 OPC Web Trend.NET 100% Managed ASP.NET component。
3. 使用在過時的應用程式（如 VB6）中的 OPC Trend ActiveX control。

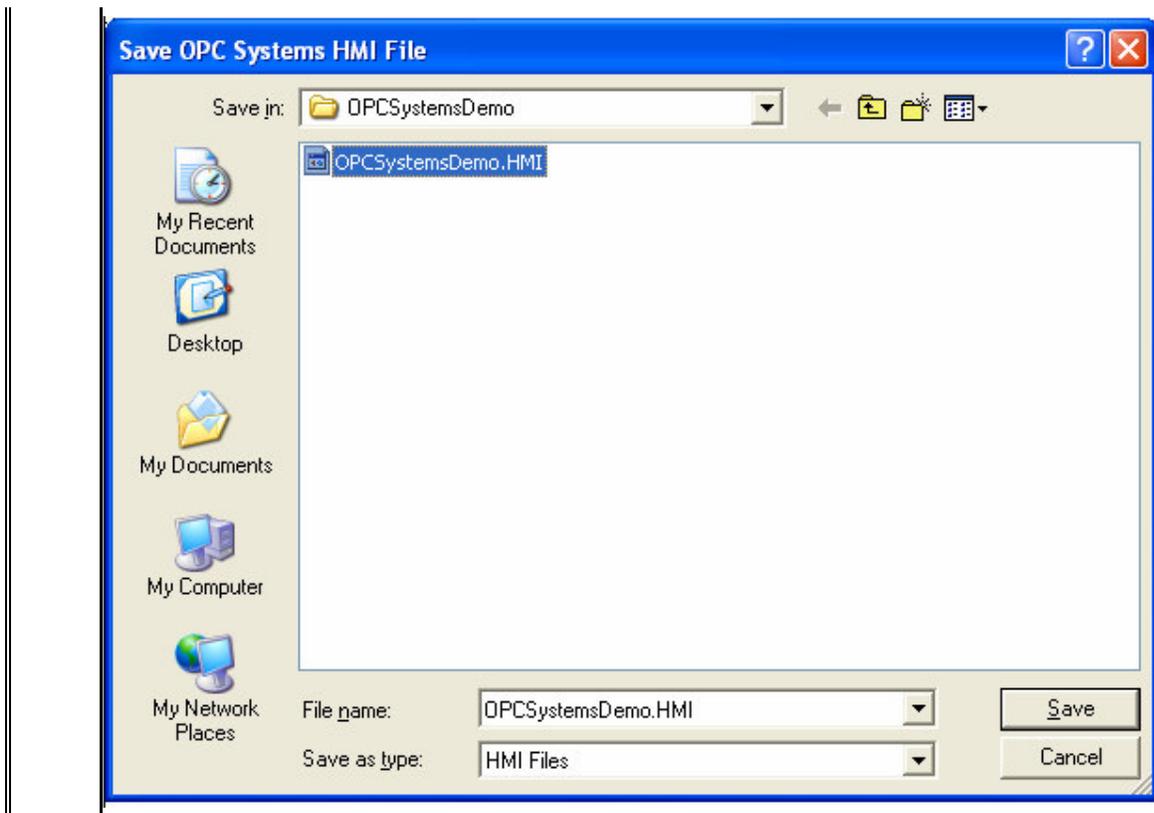
此所有 3 種 component 可以被執行在本機或遠端電腦中，來呈現從任何購買有 OPC Trend.NET 或 OPC Web Trend.NET 授權的 OPC Systems.NET 服務中所取得的即時資料。

此所有 3 種 component 也都支援搭配 OPC Database.NET 授權的 OPC Systems.NET 服務，且設定有第三章中所敘述的 data logging groups，來進行歷史資料的重播。

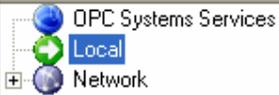
OPC Systems HMI

OPC Systems HMI 允許我們直接使用 OPC Trend.NET 和 Alarm.NET component，而不必使用 Visual Studio 來另行開發程式。

Step	Task
1	啓動 OPC Systems HMI 應用程式。
2	選擇 File → New 並儲存檔名 OPCSystemsDemo 到 C:\OPCSystemsDemo\目錄。

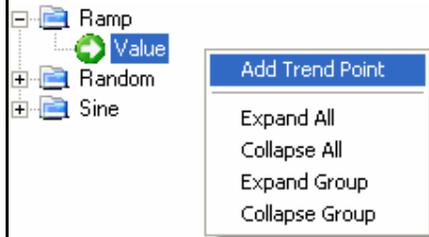


設定 Trend

Step	Task
1	啓動 OPC Systems HMI 應用程式，如果它還未被執行。
2	按下左方的 Add 按鈕。 
3	變更此 Trend Window Name 爲 OPC Systems Demo。
4	按下 Modify Trend Window 按鈕並選取 Pens。 
5	在 Trend Points Tags 對話方塊中選取 Local OPC Systems Service。  <p>*注意：如果你想要設定的 Trend Window 不是與 OPC Systems Service 在同一電</p>

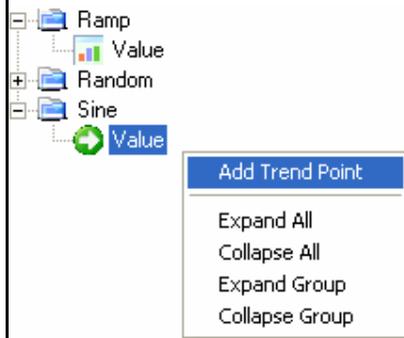
腦，則必須按下 **Select** 按鈕找到該 **NetworkNode**。

6 展開 **Ramp Tag** 並選取參數，再按右鍵選 **Add Trend Point**。

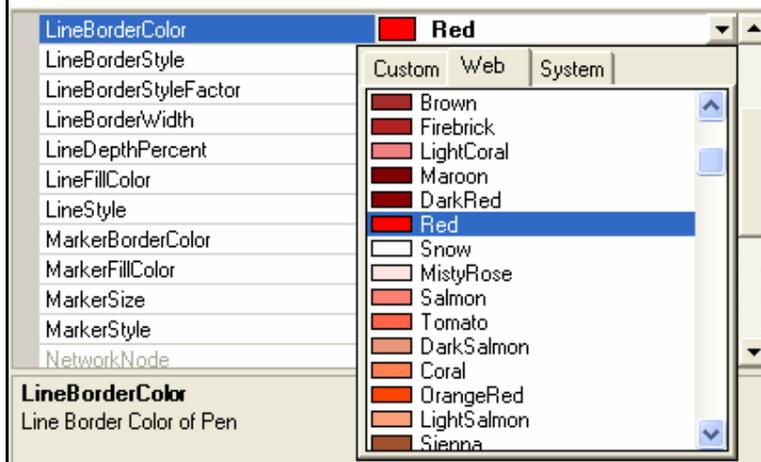


僅有在 **Configure ->Tags** 設定中被勾選 **Trend Point** 的 tag 才會出現供選取。

7 展開 **Sine Tag** 並選取參數，再按右鍵選 **Add Trend Point**。



8 在右下角的屬性窗格中設定 **Sine.Value** 筆的 **LineBorderColor** 屬性為 **Red**。



9 設定 **YAxisRangeHigh** 為 1 和 **YAxisRangeLow** 為 -1。



10 按下左下角的 **OK** 按鈕。



11 按下 **Modify Trend Window** 按鈕並選擇 **Update Rates** 。



12 設定 **TimeFrame** 到 60 秒。然後關閉 **Chart Rates** 對話方塊。

OPC Systems	
SampleRate	1
TimeFrame	60
UpdateRate	1

13 按下 **Modify Trend Window** 按鈕並選擇 **View** 。



14 設定 **Lighting** 為 **MetallicLustre**。然後關閉 **Views** 對話方塊。

OPC Systems	
Lighting	MetallicLustre
PerspectiveDepth	20
PerspectiveView	Perspective
ViewAxisDepthNumbers	False

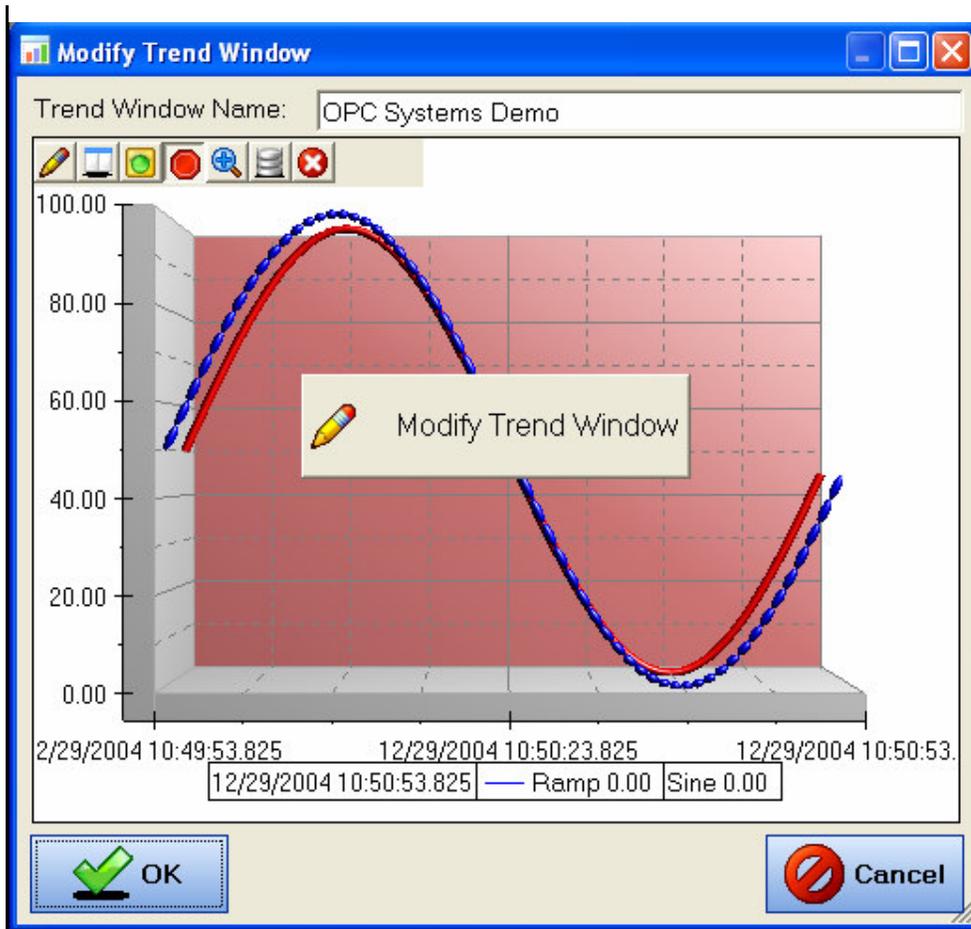
15 按下 **Modify Trend Window** 按鈕並選擇 **Walls** 。



16 設定 **BackWallColor** 為 **LightCoral**。然後關閉 **Walls** 對話方塊。

OPC Systems	
BackWallColor	 LightCoral
BackWallVisible	True
FloorColor	 White
FloorVisible	True
FrontWallColor	 White
FrontWallVisible	False
LeftWallColor	 White
LeftWallVisible	True
RightWallColor	 White
RightWallVisible	False

17 **Trend Window** 呈現如下。



18 按下左下角的 OK 按鈕。

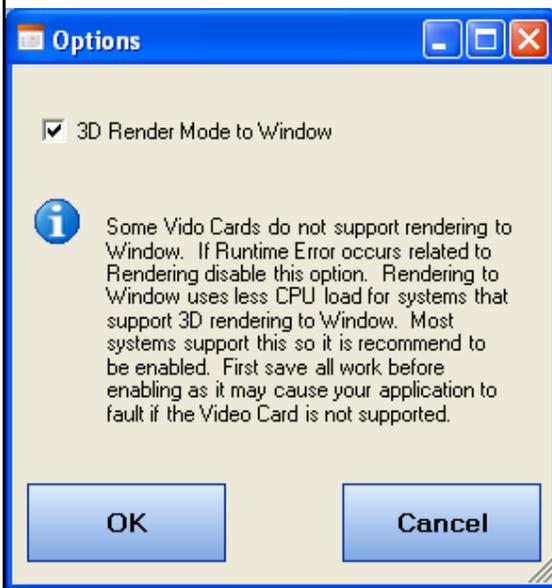


19 選擇 File → Save。



這是當新增或刪除一個 Trend Window、Alarm Window 或 View Window 時是必要的。在 Trend 或 Alarm Window 中的修改會自動地儲存在自己的設定檔中。

20 選擇 Configure → Options 和勾選 3D Render Mode to Window，並按下 OK 按鈕。



點選 Yes 按鈕以儲存 Render Mode。

21 點選 OPC Systems Demo Trend Window 並按下 Modify 按鈕。



如果 Window 正常呈現，則按下 Cancel 按鈕取消即可。

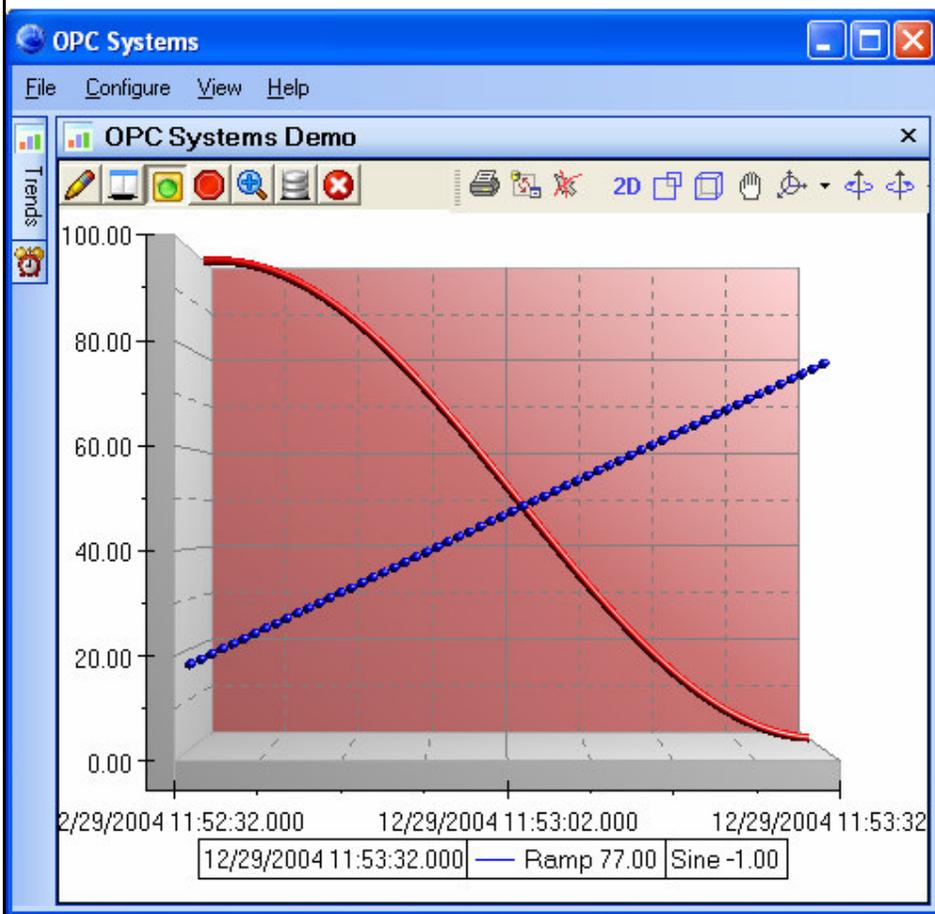


如果造成 OPC Systems HMI 無回應，則請記得再到 Configure → Options 中將 3D Render Mode to Window 的勾選移除。

22 此 Trend Window 現在已經可以被 Run Trend Window in OPC Systems HMI 部份來使用。

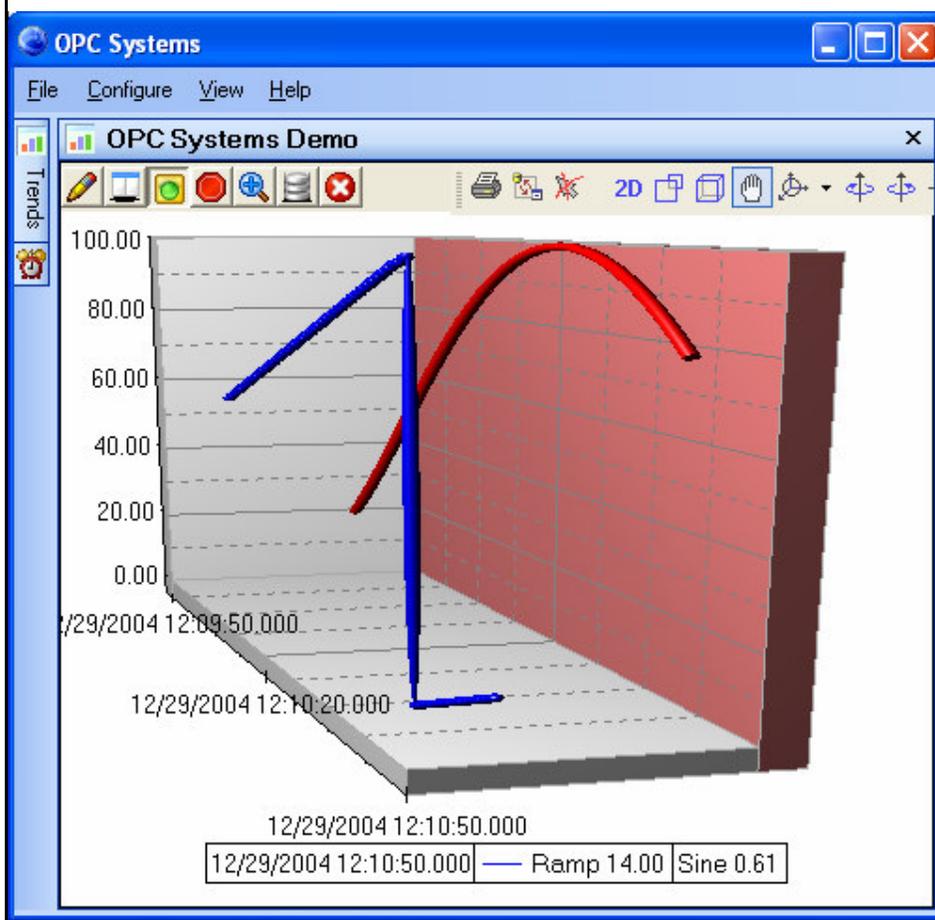
在 OPC Systems HMI 中執行 Trend Window

Step	Task
1	啟動 OPC Systems HMI 應用程式。
2	按下左下角的 Trends 按鈕，並點選 OPC Systems Demo Trend Window，再按下 Add Trend To Window 按鈕。 
3	Trend Window 應該以如下的方式出現，並根據 OPC Systems Service 的資料即時更新。



如果資料沒有在幾秒內出現，請確認已為 OPCSystemsHMI.exe 和 OPCSystemsService.exe 打開 port 58723 防火牆限制。然後重新啓動 OPC Systems Service 試試看。

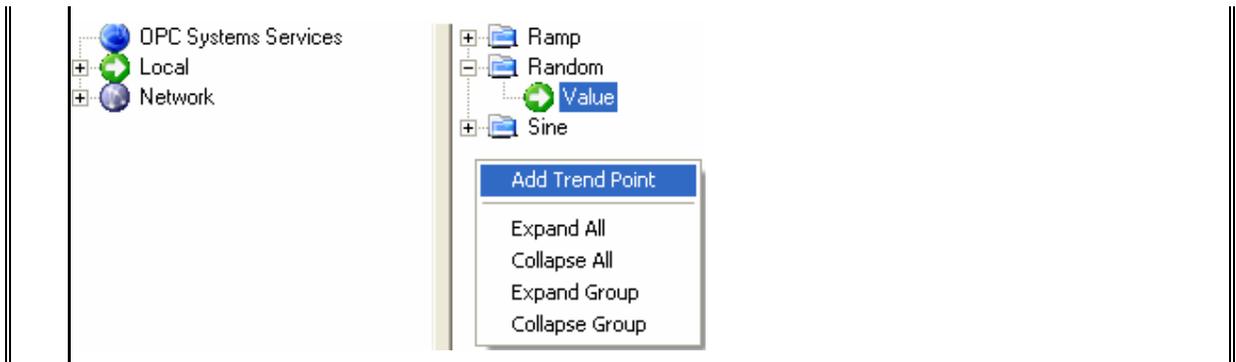
4 在工具列中選取手的形狀指標，然後移動滑鼠來為 3D 趨勢圖改變檢視的角度。



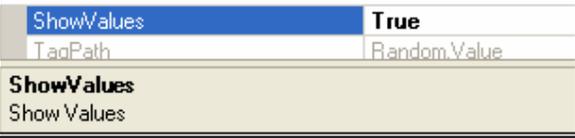
5 在 Trend Window 上按下滑鼠右鍵並選擇 Modify Pens。



6 選取 Local OPC Systems Service 並展開 Random Tag，點選 Value 參數，然後按下滑鼠右鍵以 Add Trend Point。



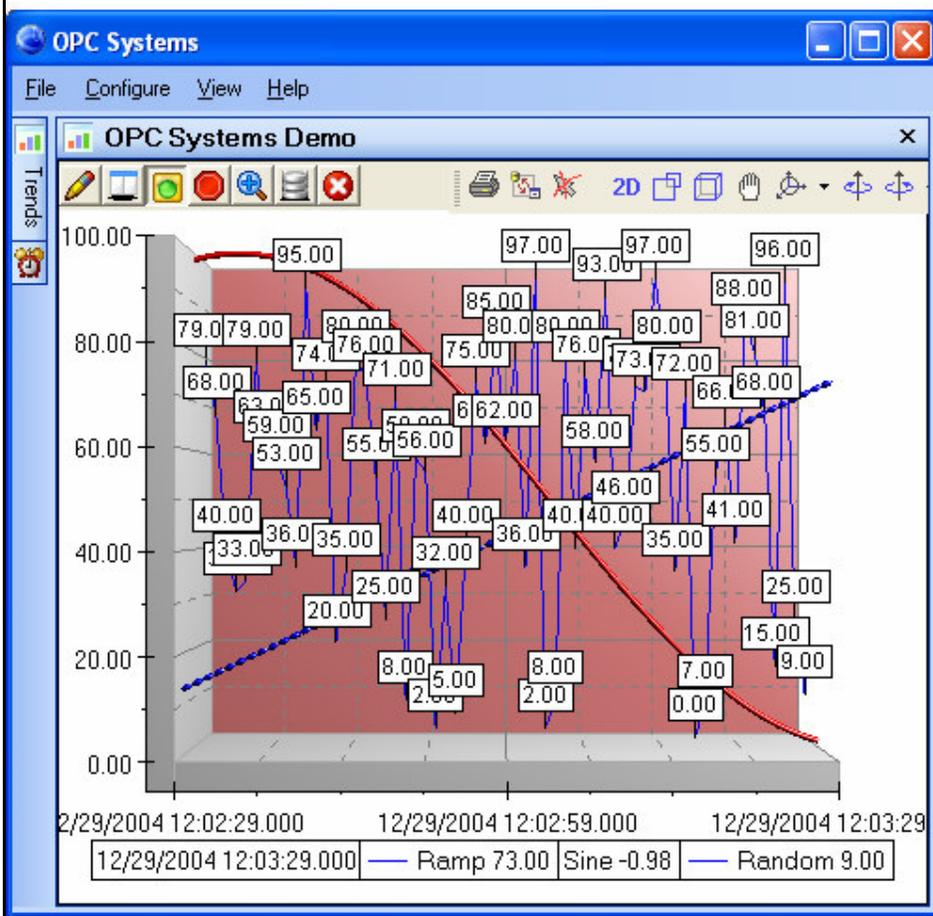
7 在左下角的 Pen 清單中選擇 Random.Value，並設定 ShowValue 屬性為 True。



8 按下 Trend Point Tags 對話方塊左下角的 OK 按鈕。

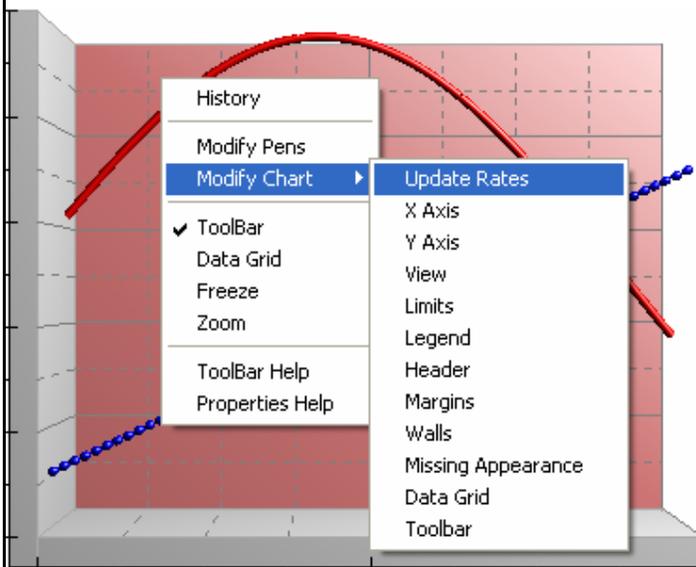


9 請注意，Random Pen 的值會被 OPC Systems Service 放在 buffer 中，即使它先前沒有被加入 Trend Window。

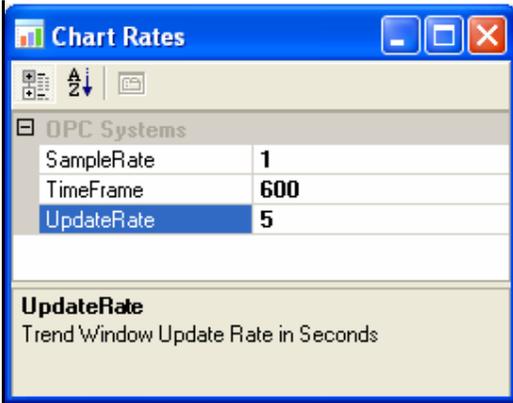


我們可以簡單地從 Pen 清單中，選取 Modify Pens 及刪除 Pen。

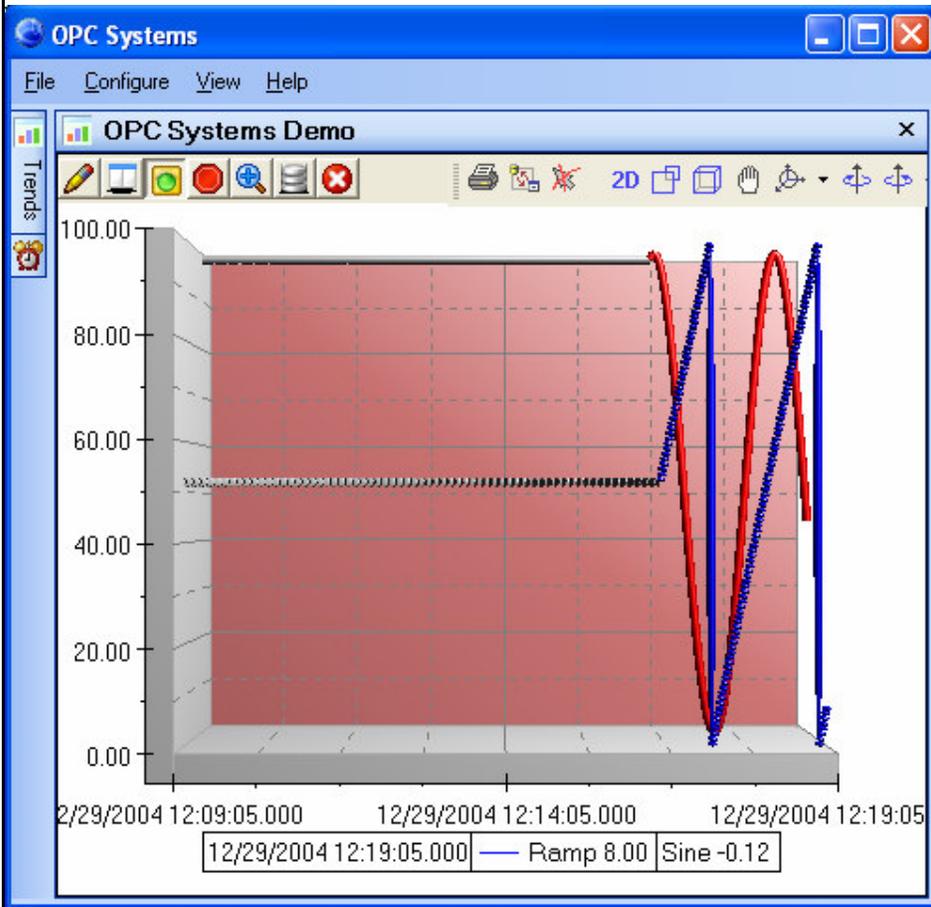
10 在 Trend Window 上按下滑鼠右鍵並選擇 Modify Chart → Update Rates。



11 設定 TimeFrame 為 600。



- 12 此新的 Time Frame 將會讓 OPC Systems Service 中所有的 Trend Points 具有 1 秒的 Sample Rate。為讓在 3D 模式下能有快速的更新速率，Trend Window 更新速率是設成 6 秒。

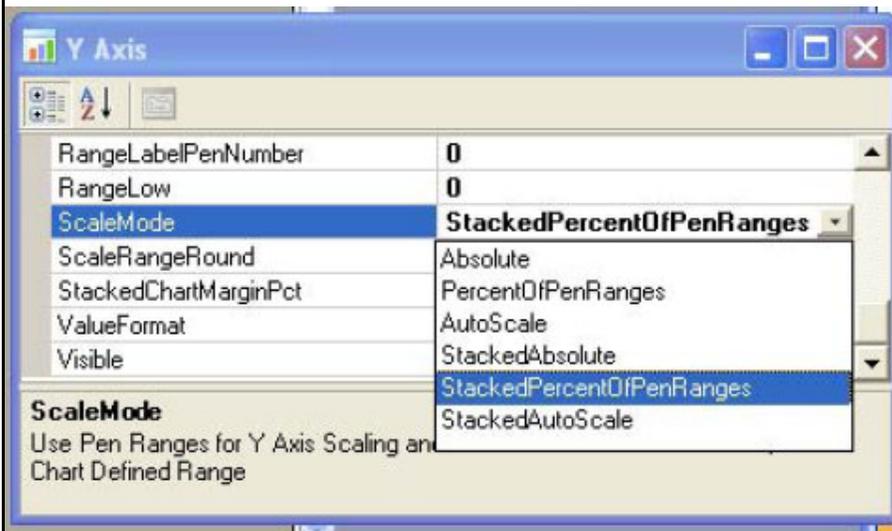


- 13  選擇 File → Save，如果我們想要儲存此 on-line Window name 並保持現況安排。所有對 Trend Window 屬性的變更都會被保持，即使我們沒有再次選擇 Save。僅當 Window 被加入或移除才需要再選擇 Save。

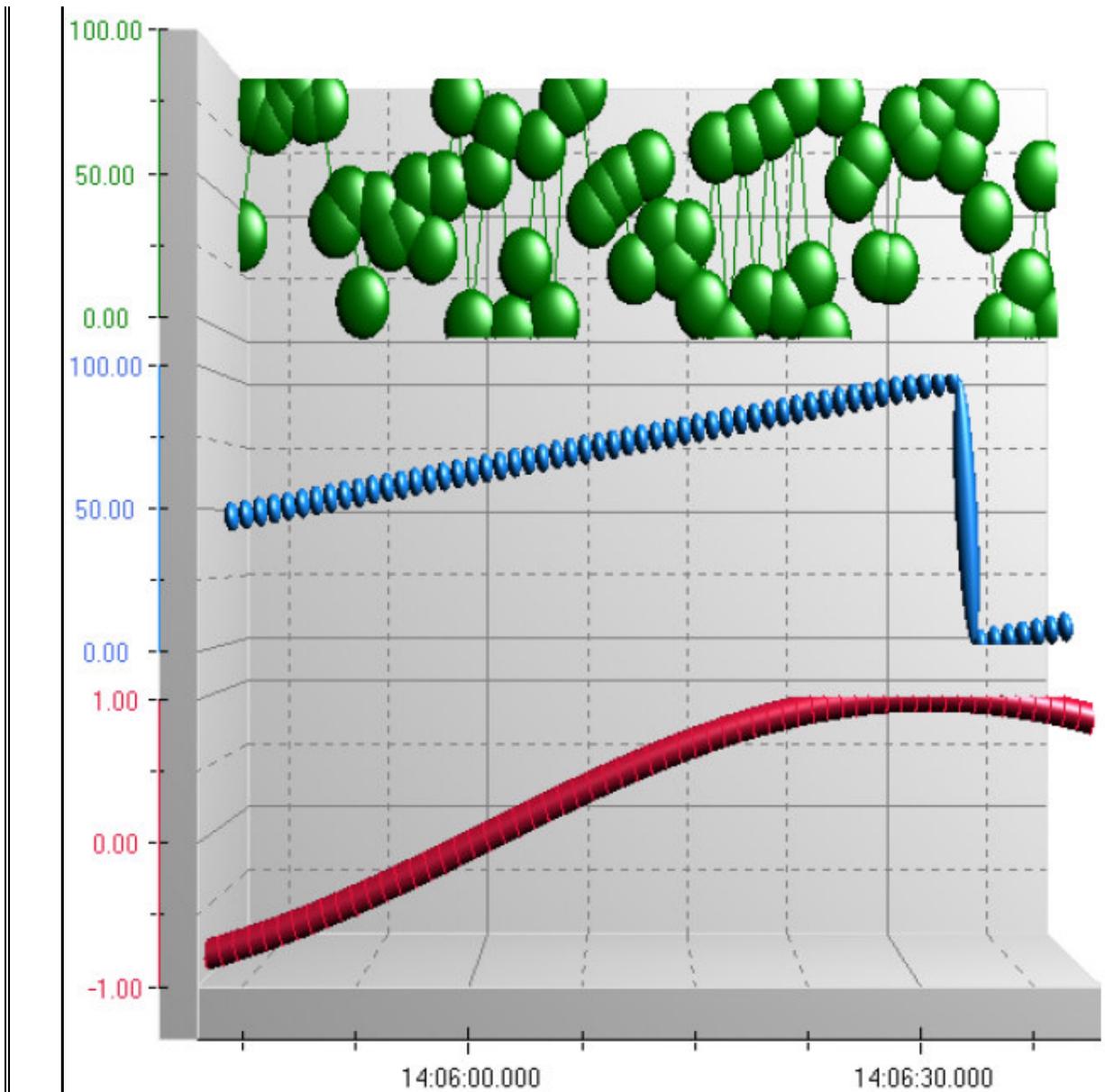
- 14 如果設定有 Data Logging，則可在 Trend Window 上按下滑鼠右鍵並選取 History 來取出歷史資料。



- 15 在 Trend Window 上按下滑鼠右鍵並選取 Modify Chart-Y Axis 並改變 ScaleMode 從 PercentOfPenRanges 到 StackedPercentOfPenRanges。



- 16 在 Trend Window 上按下滑鼠右鍵並選取 Modify Pens 並設定每個 Pen 的 StackedChartNumber 屬性為個別獨特數字，然後按下 OK 按鈕來確認此一變更。



17 瀏覽 Trend Window 工具按鈕列上的其他功能,包括 Data Grid, Data Cursors, Data Zoom 等等。

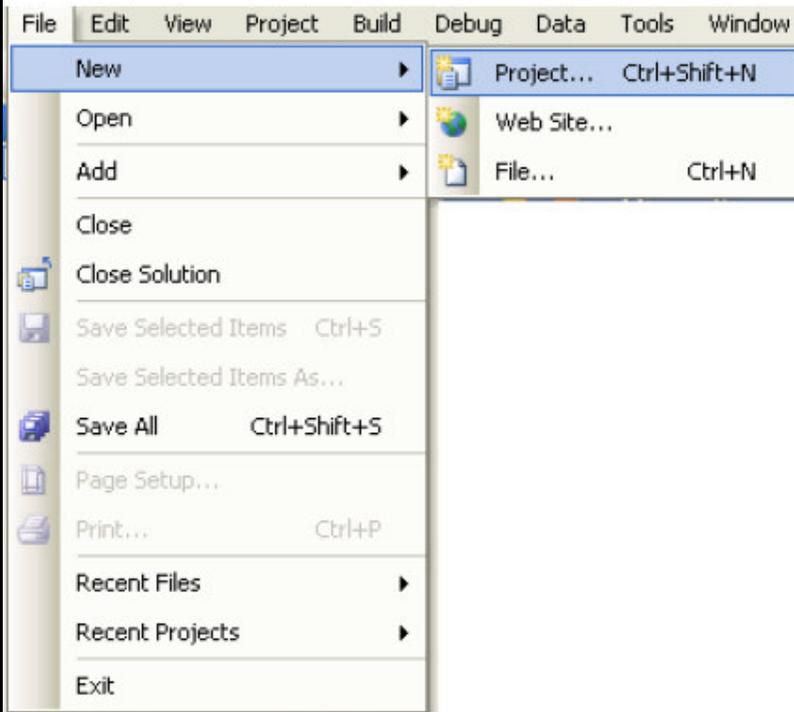
此同一 Trend control 可以被加進我們的 Visual Studio 應用程式。

新增 Trend 到 Visual Studio 應用程式

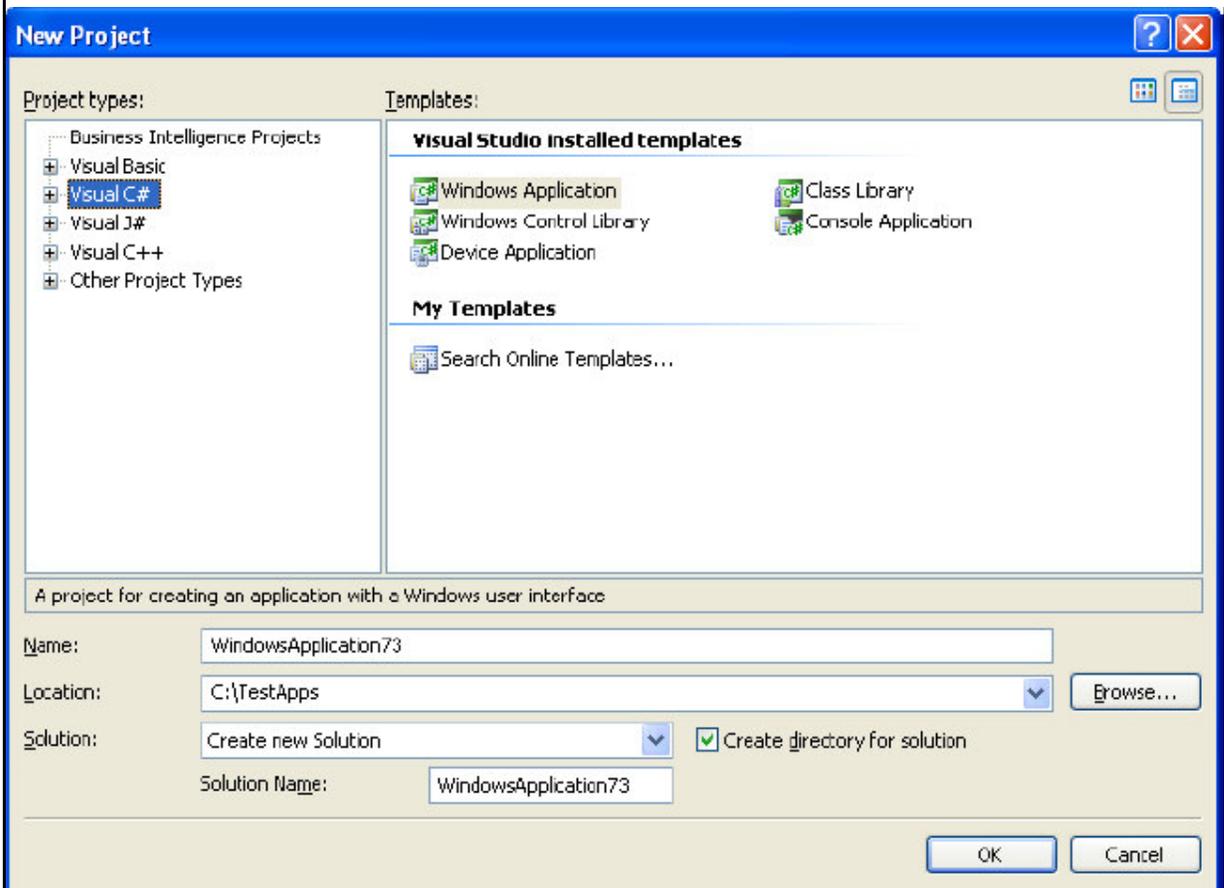
以下的步驟可以被用來新增 Trend Window 到一個 C#、C++或 Visual Basic.NET 應用程式中。請參考 VB.NET 範例程式來知道如何新增和移除 Trend Window 中的 pen。Trend Window 的所有屬性都可以利用程式來讀取和修改。此例子展示不撰寫程式碼的方式來建立 Trend Window。

Step	Task
------	------

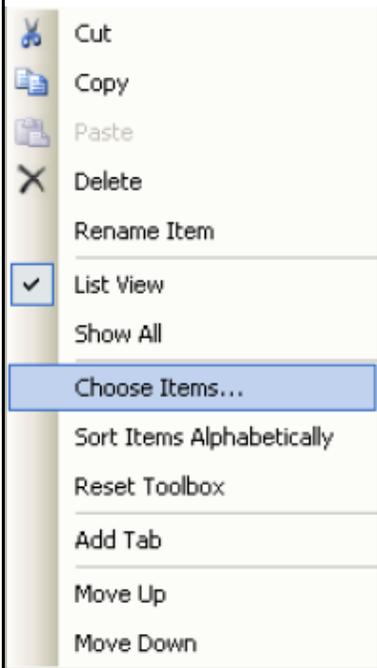
- 1 啓動 Visual Studio 2005 或 Visual Studio 2008，並選擇 File->New->Project 來建立新的 C#、C++或 Visual Basic.NET 應用程式專案。



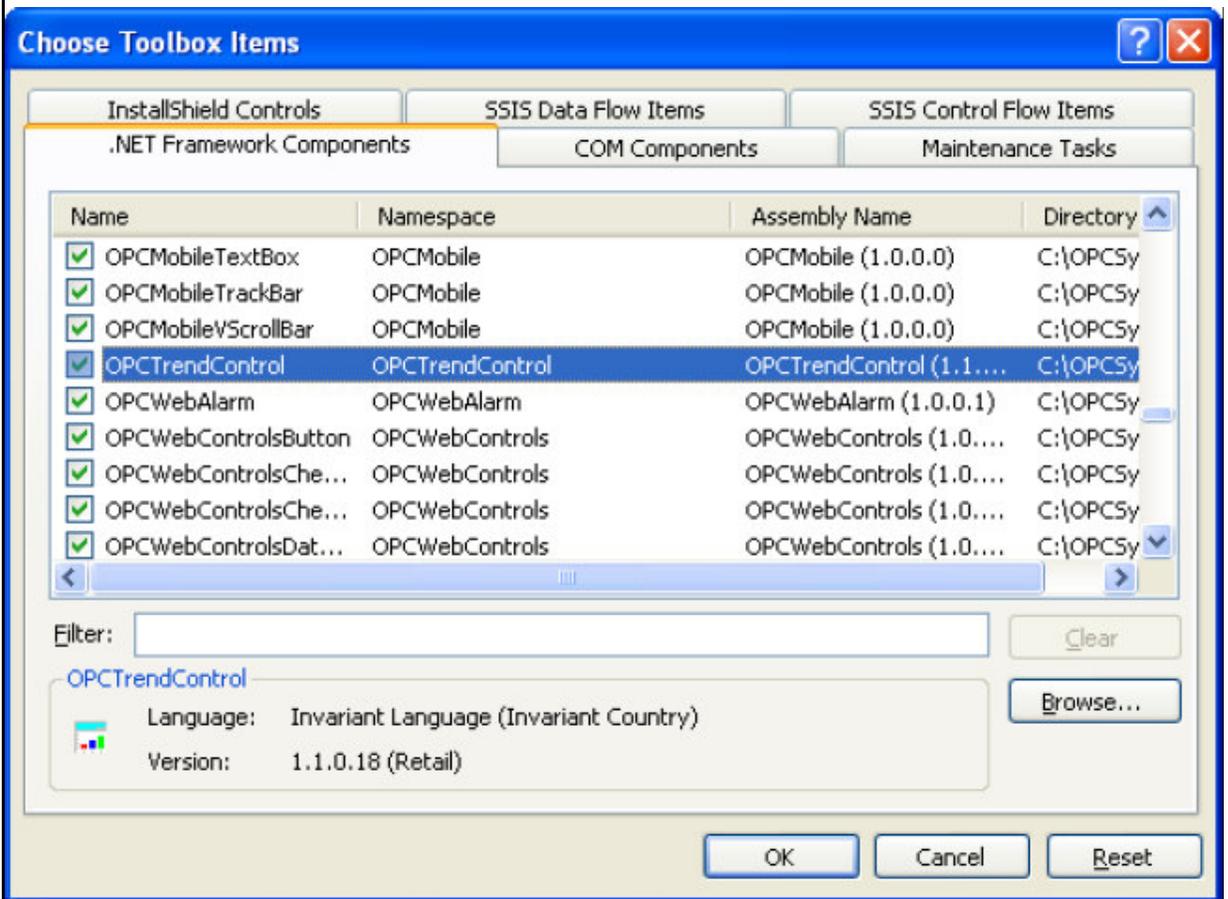
- 2 選擇 Windows Application 作為專案類型。



- 3 如果 OPCTrendControl 尚未存在於工具箱中，則請在工具箱中按下滑鼠右鍵並選取 Choose Items...。



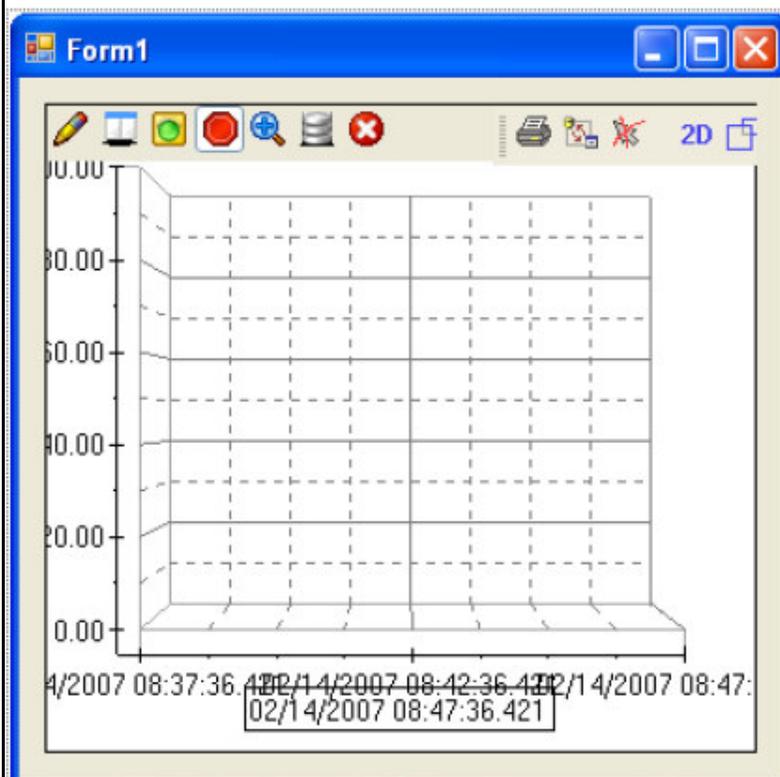
從 .NET Framework Components 中勾選 OPCTrendControl 並按下 OK 按鈕。



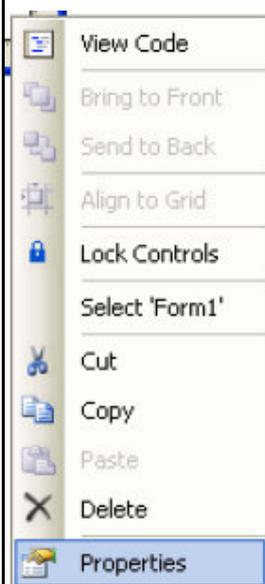
4 新增 OPCTrendControl component 到 Form 中。



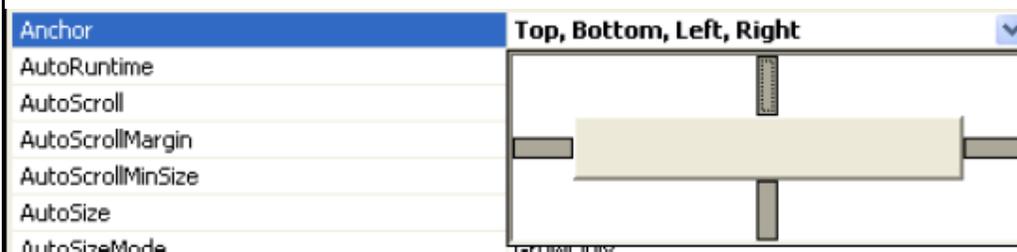
調整 Form 和 Trend Window 的大小至適當的。



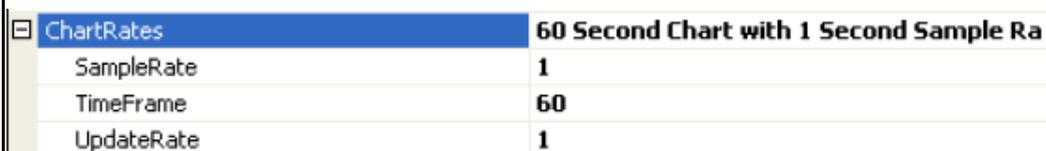
5 在 Trend Window 上按下滑鼠右鍵並選取 Properties。



6 設定 Anchor 屬性為 Top, Bottom, Left, Right。



7 展開 ChartRates 屬性並設定 TimeFrame 為 60 秒。



8 選擇 Pens 屬性並點按右方  按鈕。

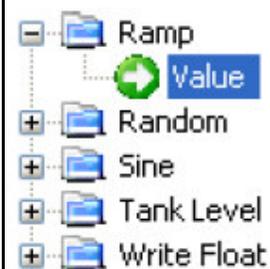


9 選擇 Local OPC Systems Service 來列出可用的 Tags。

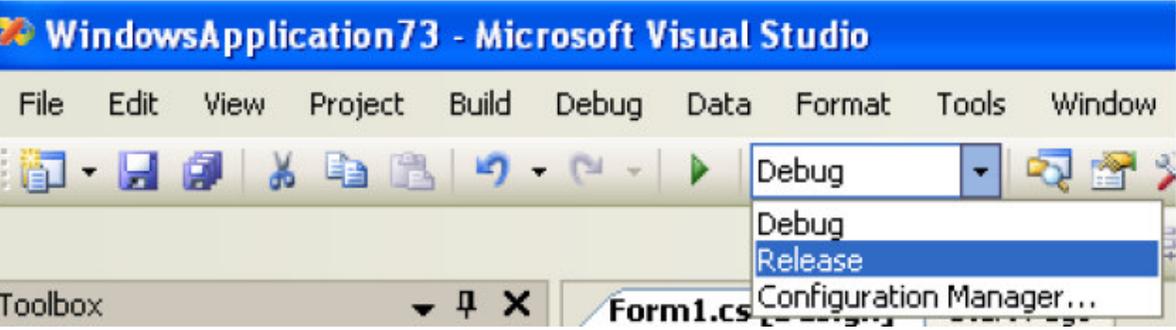


注意：如果你看不到想要用的 Tag 在清單中，則請回到 Configure->Tags，在欲使用的 Tag 參數設定中並勾選 Trend Point。

10 展開 Ramp Tag 並選取 Value。

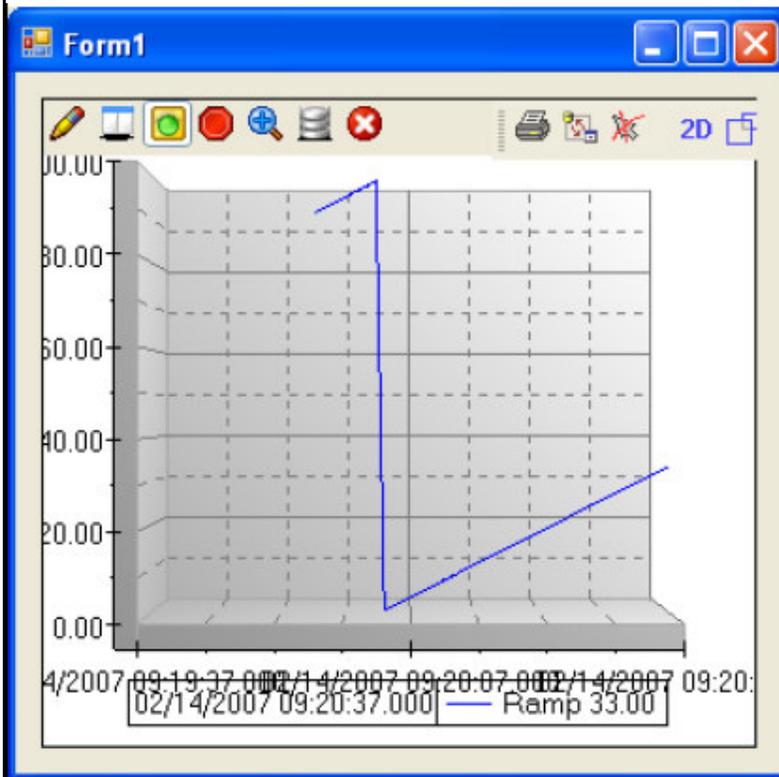


按下 Add Pen 按鈕。

	
11	<p>Ramp.Value 會出現在左下角的 pens 清單中。我們可以選擇 pen 來變更位於右方的屬性。當 Trend Window 的 YAxis.ScaleMode 屬性被設定成 PercentOfPenRanges 時，YAxisRangeHigh 和 YAxisRangeLow 屬性是非常重要的。</p>
12	<p>在 Pens 對話方盒中按下 OK 按鈕。</p>
13	<p>如果此電腦上的顯示卡支援 OpenGL rednering，則可設定 RenderMode3D 屬性為 True。</p>  <p>如果不確定的話，則保留此屬性為 False。</p>
14	<p>展開 Views 屬性並設定 Lighting 和 PerspectiveView 的為想要使用的值。</p> 
15	<p>如果我們想要讓操作者在執行期間對 Trend Window 的設定進行保留，請設定 ConfigurationFile 屬性到一個有效的檔案路徑。</p>  <p>注意：如果你設定了此一屬性到某一個檔案，則請確實將該檔案放置於你指定的目錄中。</p> <p>如果我們希望在每次應用程式重啟動時都維持最初的原始設定，則讓此屬性保留為空白。</p>
16	<p>設定 Visual Studio 工具列中的 compile mode 為 Release。</p> 
17	<p>從 Visual Studio 的 Build 功能表中選擇 Build 此應用程式。</p>



18 使用檔案總管來瀏覽放置在 bin\Release 目錄中的執行檔並執行。

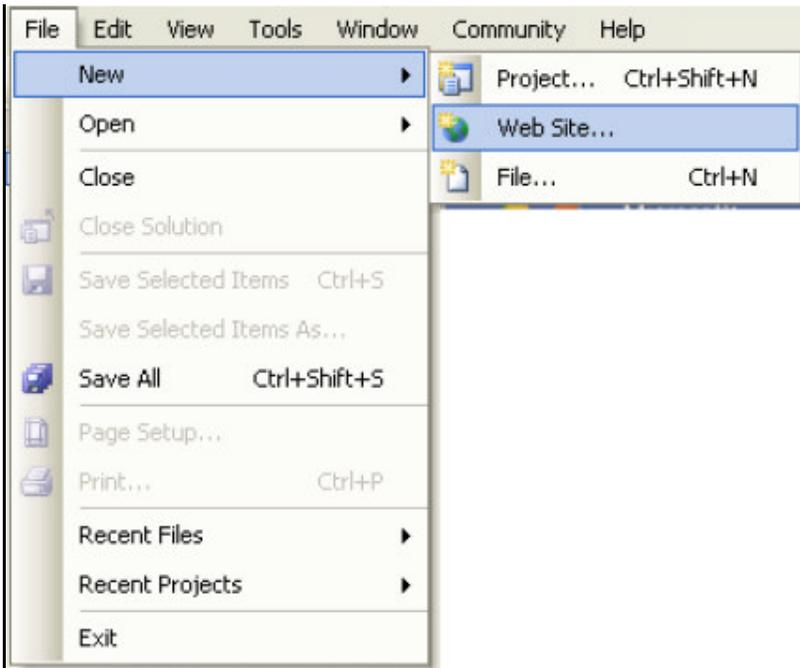


19 欲發佈此應用程式到遠端電腦中，則請確定在步驟 9 的 pen 選擇中，是選擇網路上的電腦或是 IP 位址。然後複製放置在 bin\Release 目錄中的所有檔案至目的端電腦。

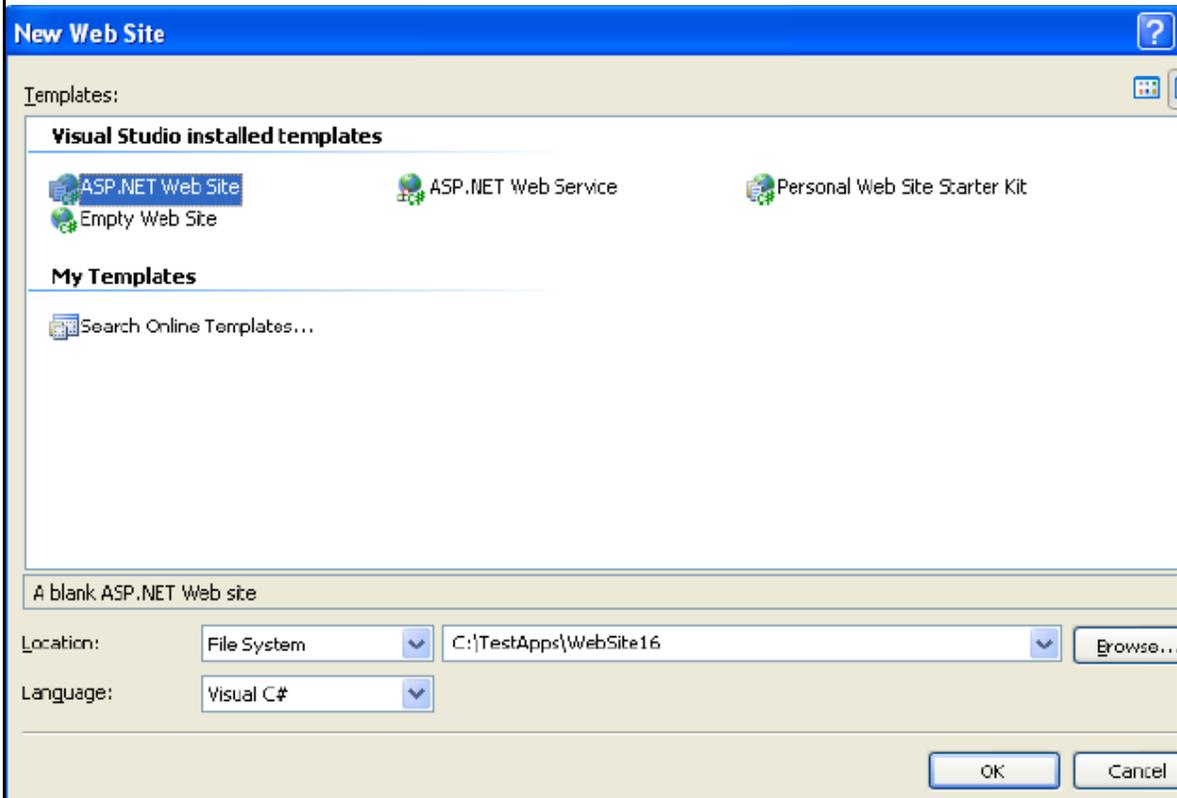
新增 Trend 到 ASP.NET Web 應用程式

以下的步驟可以被用來新增 Trend Window 到一個 C#、C++或 Visual Basic.NET Web 應用程式中。Trend Window 的所有屬性都可以利用程式來讀取和修改。此例子展示不撰寫程式碼的方式來建立 Trend Window。

Step	Task
1	啟動 Visual Studio 2005 或 Visual Studio 2008，並選擇 File->New->Web Site 來建立新的 C#、J#或 VB ASP.NET Web 應用程式專案。

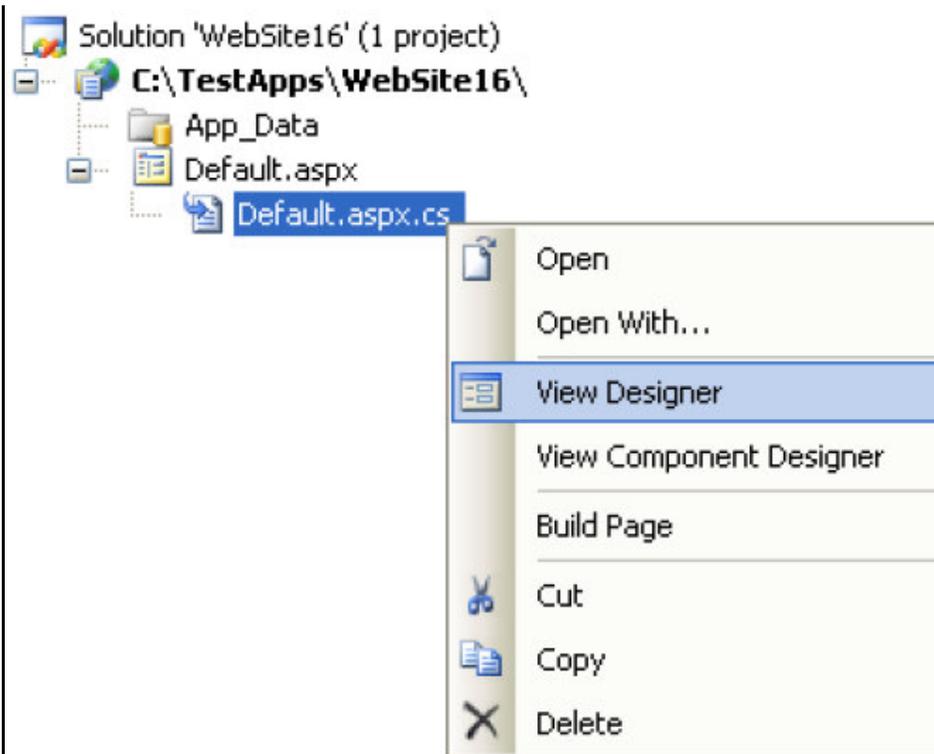


2 選擇 ASP.NET Web Site 專案類型，並指定左下方所欲使用的程式語言。

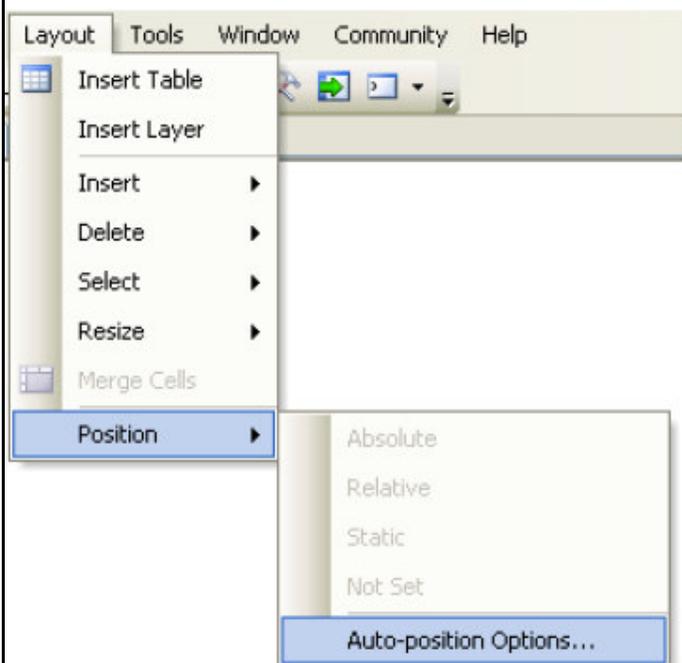


3 展開方案管理員中的 Default.aspx 網頁，並選取 View Designer。

注：如果你沒有看到方案管理員，則請選擇功能表 View->Solution Explorer。

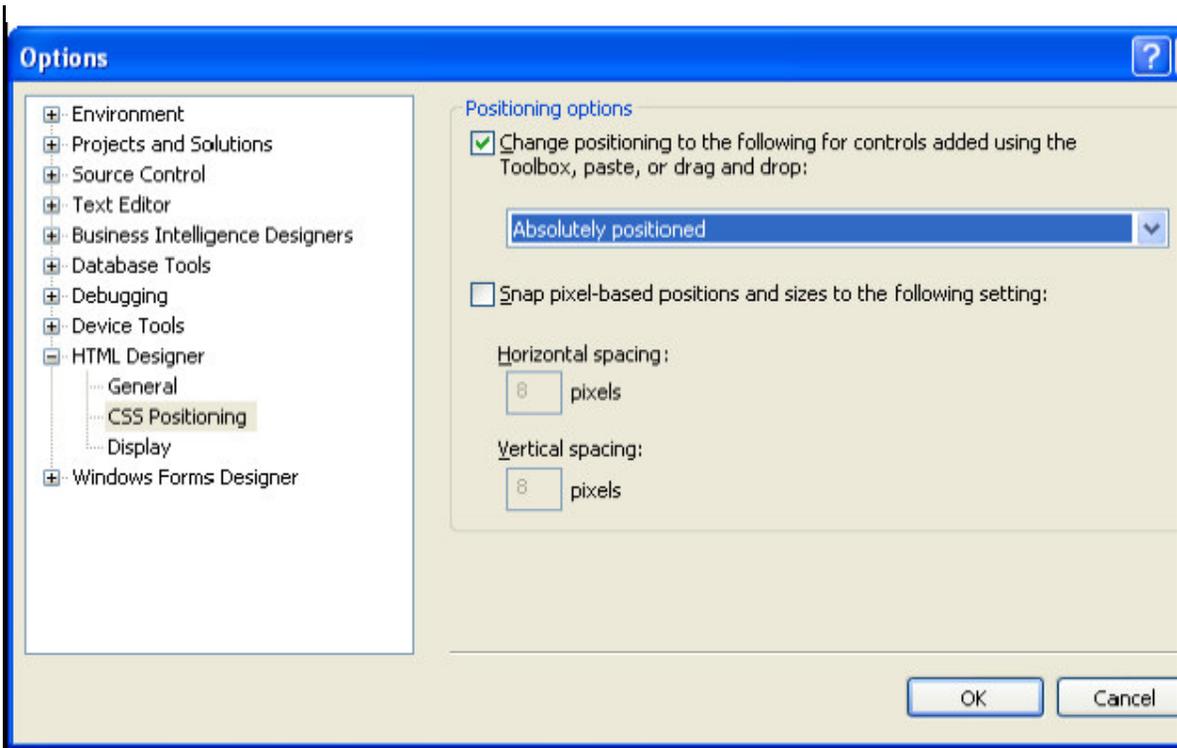


4 從 Visual Studio 功能表中選擇 Layout->Position->Auto-position 選項。

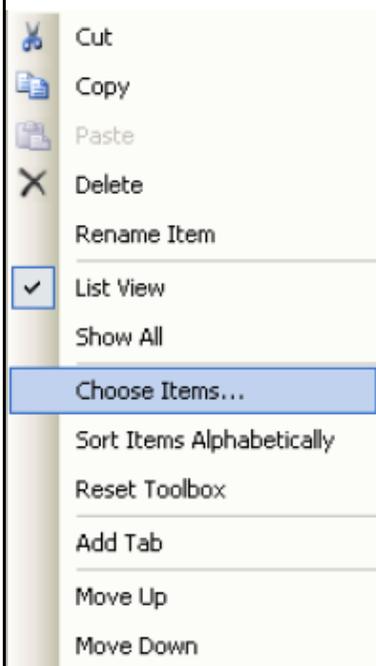


5 在 HTML Designer 中設定 CSS Positioning 為 Absolutely positioned。

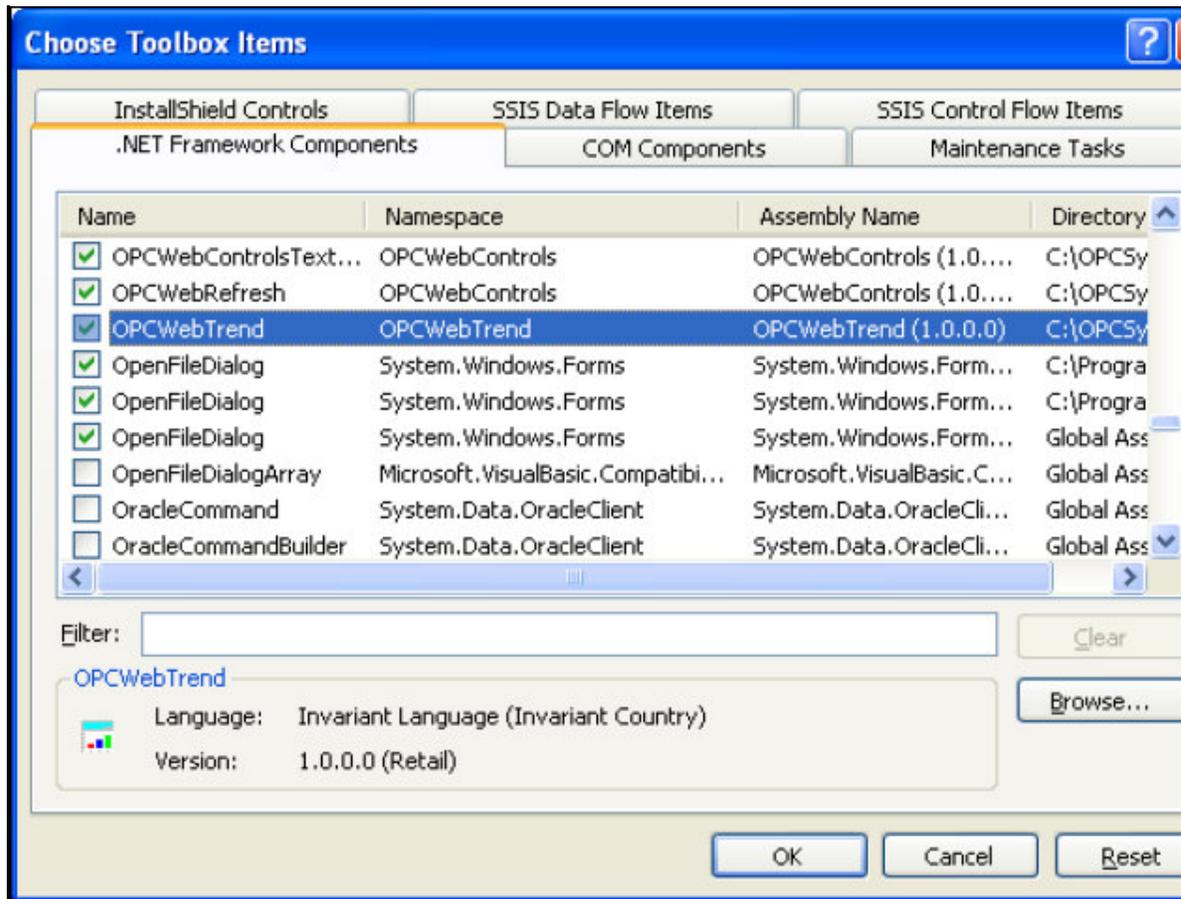
注：我們可以使用任何想要使用的定位型態。



- 6 如果 OPCWebTrend 和 OPCWebRefresh 尚未出現在工具箱中，則請在工具箱中按下滑鼠右鍵並選擇 Choose Items 來選取。



從 .NET Framework Components 中勾選 OPCWebTrend、OPCWebControlsLabel 和 OPCWebRefresh 並按下 OK 按鈕。



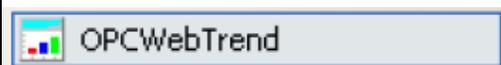
7 新增一個 OPCWebRefresh 控制項至 WebForm 中。



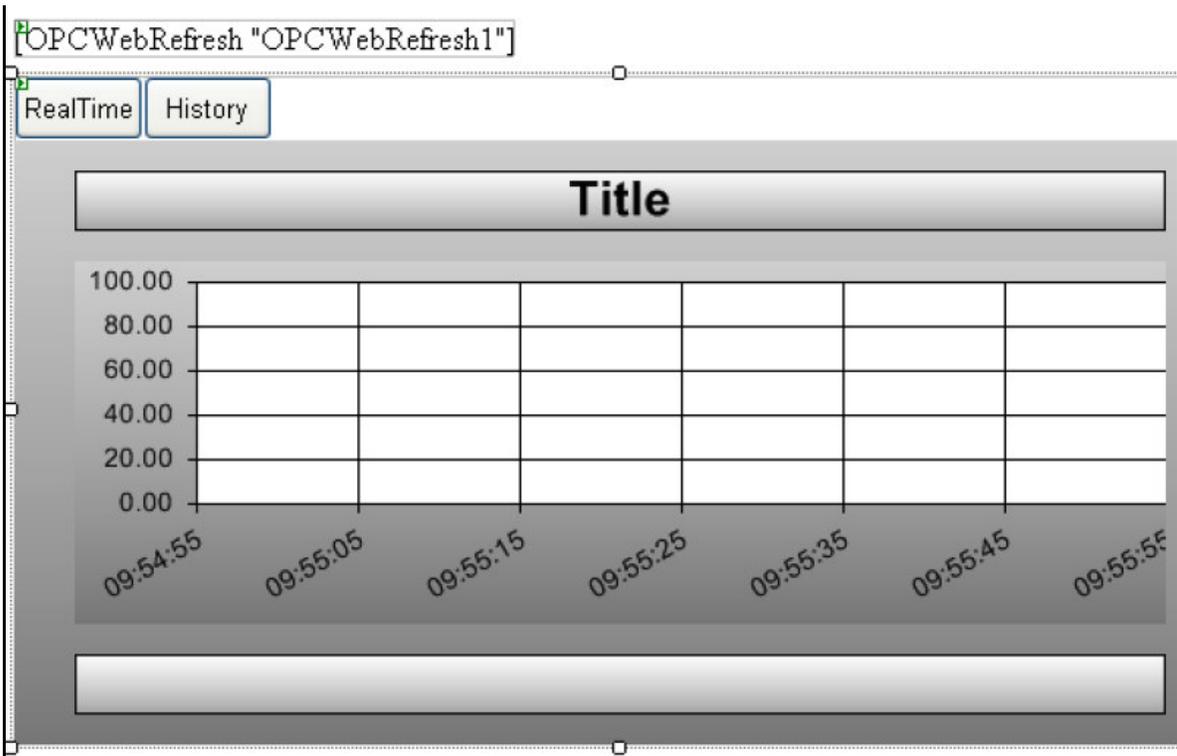
```
[OPCWebRefresh "OPCWebRefresh1"]
```

為確定專案的 web.config 檔已適當地修改為能夠執行 AJAX runtime，新增一個 OPCWebControlsLabel 到 WebForm 中，然後再刪除之。

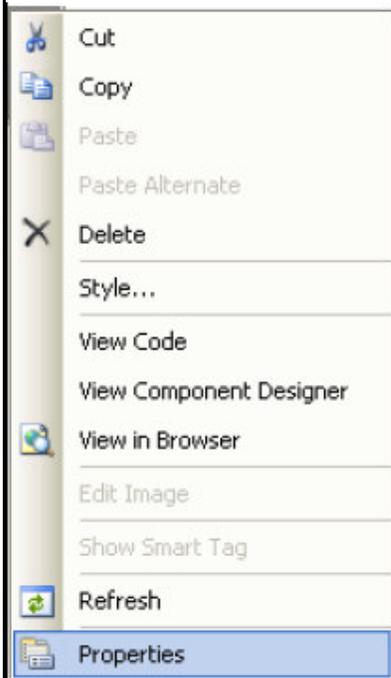
8 新增 OPCWebTrend 控制項至 WebForm 中。



調整 Trend Window 到想要的大小。



9 在 Trend Window 上按滑鼠右鍵並選擇 Properties。



10 選擇 Layout 型態至需要的設定。

Layout	ChartMiddleTitleTopLegendBottom
LegendBackgroundGradientEndColor	ChartMiddleTitleTopLegendBottom
LegendBackgroundGradientStartColor	ChartMiddleTitleBottomLegendTop
LegendBackgroundGradientType	ChartBottomTitleMiddleLegendTop
LegendBackgroundSimpleColor	ChartBottomTitleTopLegendMiddle
LegendBackgroundType	ChartLeftTitleBottomLegendRight
LegendBorderColor	ChartLeftTitleTopLegendRight
LegendBorderVisible	ChartRightTitleBottomLegendLeft
LegendFont	ChartRightTitleTopLegendLeft
LegendIconMode	ChartTopTitleBottomLegendMiddle
LegendLayout	ChartTopTitleMiddleLegendBottom

11 設定 SampleRate 和 TimeFrame 屬性至需要的設定。預設的 TimeFrame 是 60 秒，Sample Rate 是 1 秒。

SampleRate	1
ScaleMode	Absolute
SkinID	
TabIndex	0
TimeFrame	60

12 選擇 Pens 屬性並按下右方  按鈕。

Pens	ClassPen[] Array
------	------------------

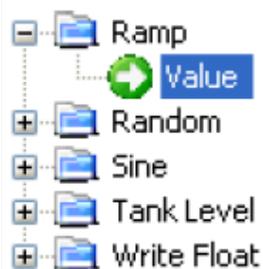
13 選擇 Local OPC Systems Service 來列出可用的 Tags。



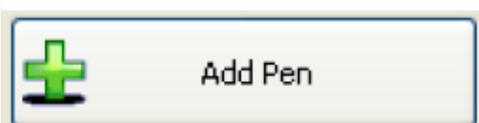
注意：如果你看不到想要用的 Tag 在清單中，則請回到 Configure->Tags，在欲使用的 Tag 參數設定中勾選 Trend Point。

注意：如果你希望將 ASP.NET Web 應用程式透過網路安裝在遠端的電腦上，則請選擇 Network Node 或在 NetworkNode 欄位中鍵入 IP 位址，並使用 Select 按鈕來含括執行 OPC Systems Service 來源的 Network Node 或 IP 位址。

14 展開 Ramp Tag 並選擇 Value。



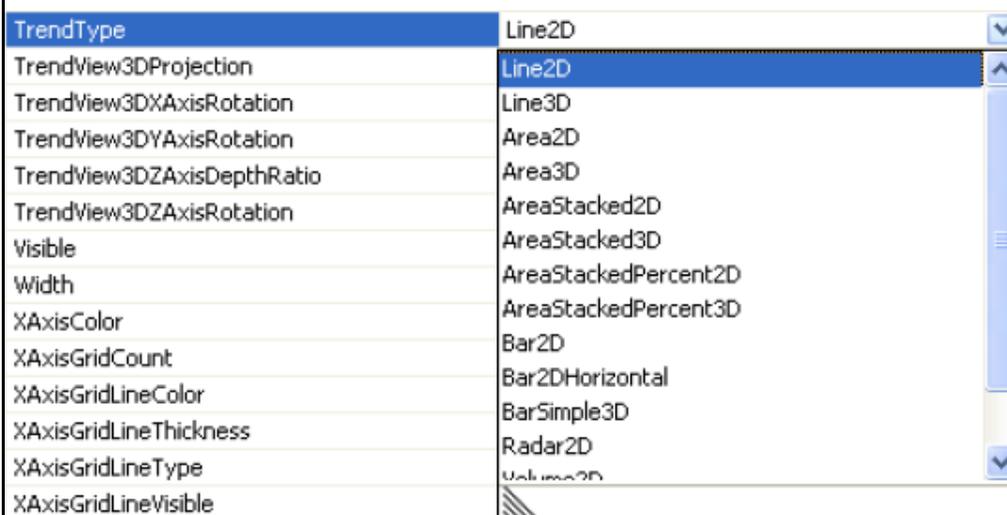
按下 Add Pen 按鈕。



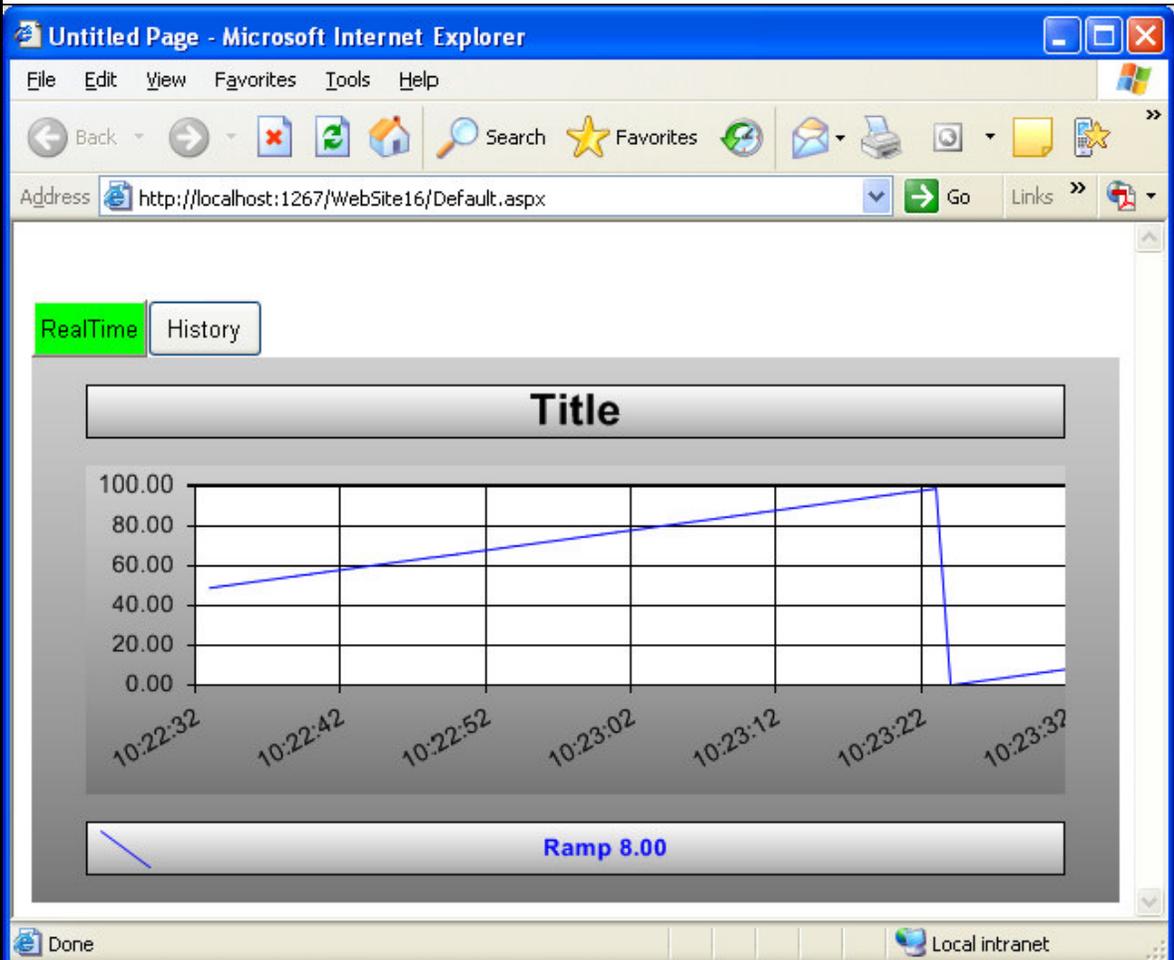
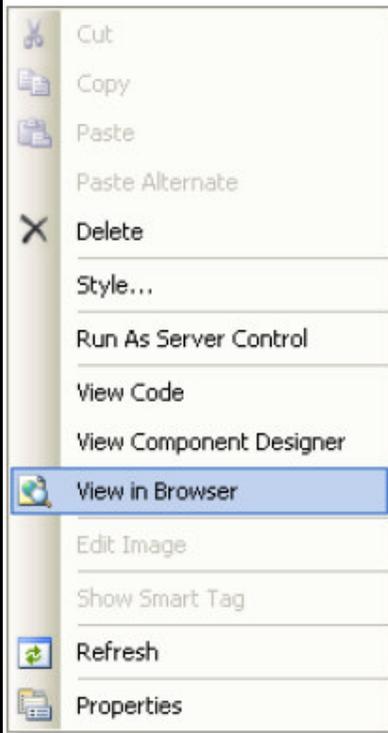
15 Ramp.Value 會出現在左下角的 pens 清單中。我們可以選擇 pen 來變更位於右方的屬性。當 Trend Window 的 YAxis.ScaleMode 屬性被設定成 PercentOfPenRanges 時，YAxisRangeHigh 和 YAxisRangeLow 屬性是非常重要的。

16 在 Pens 對話方盒中按下 OK 按鈕。

17 設定 TrendType 屬性為想要的設定值。



18 在 WebForm 上按滑鼠右鍵並選擇 View in Browser。



19 設定此應用程式為 Release 模式，修改 web.config 檔以確認 Debug 參數是設為 False，並

且 Build 此應用程式。通常 Web 應用程式檔案都會被放在 Inetpub\wwwroot 目錄中。

參考 OPC Systems.NET help 檔中的 OPC Web Trend.NET 部份來了解特殊的考慮和開發步驟。

如 Web Trend 無法顯示出來，則請以上步驟來解決：

首先，請確定你的 Web Form 中是否有加入 OPCWebRefresh 這個用來即時更新的控制項。使用預設的 OPCWebRefresh1 為名稱即可。每個 Web Form 中只需要一個 OPCWebRefresh 控制項。

適當地變更 OPCWebRefresh 控制項更新速度屬性。

檢查 C:\Program Files\Open Automation Software\OPC Systems.NET\OPCWebControlsExample\資料夾中是否存在 OPCWebTrendImages 資料夾。若沒有，請建立它。

執行命令提示字元 (cmd.exe)，將下列的設定參數指定給 OPCWebTrendImages 資料夾。
Cacls.exe "C:\Program Files\Open Automation Software\OPC Systems.NET\OPCWebControlsExample\OPCWebTrendImages" /T /E /C /G "[COMPUTERNAME]\ASPNET":C

其中 [COMPUTERNAME] 是指正在進行設定的這部電腦的名稱，電腦名稱不必加上中括號。

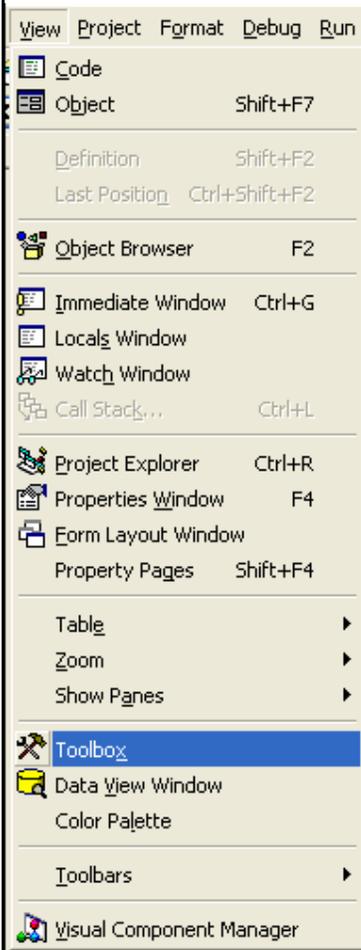
新增 Trend ActiveX 控制項到過時的應用程式

OPC Trend ActiveX 控制項可以被直接整合進任何 ActiveX container，較常用的是 HMI 圖控軟體或是 Visual Studio 6.0。

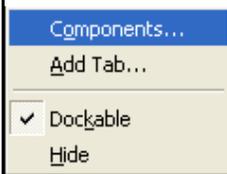
注：ActiveX 控制項並不是 100% managed 程式，所以不能被使用在 Visual Studio.NET 應用程式中。

以下是使用 OPC Trend ActiveX 控制項整合進 Visual Basic 6.0 或是其他 ActiveX container 程式的範例。

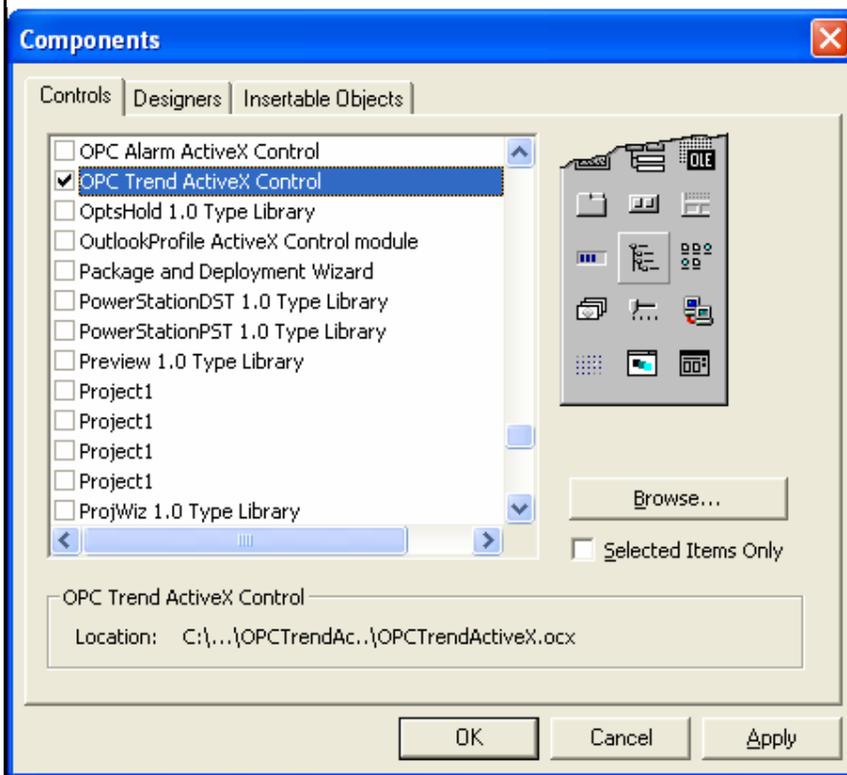
Step	Task
1	啓動 Visual Basic 6.0 的專案，或是其他的 ActiveX container 程式。 如果是使用其他的 ActiveX container 而不是 Visual Studio，則只要選擇該程式中適當的方法，以選擇 OPCTrendActiveX.OPCTrend 來加入此 ActiveX control，並跳至步驟 6。
2	在 Visual Studio 6.0 開發環境中選擇 View → Toolbox 來開啓 Toolbox。



3 在 Toolbox 上按右鍵並選擇 Components。

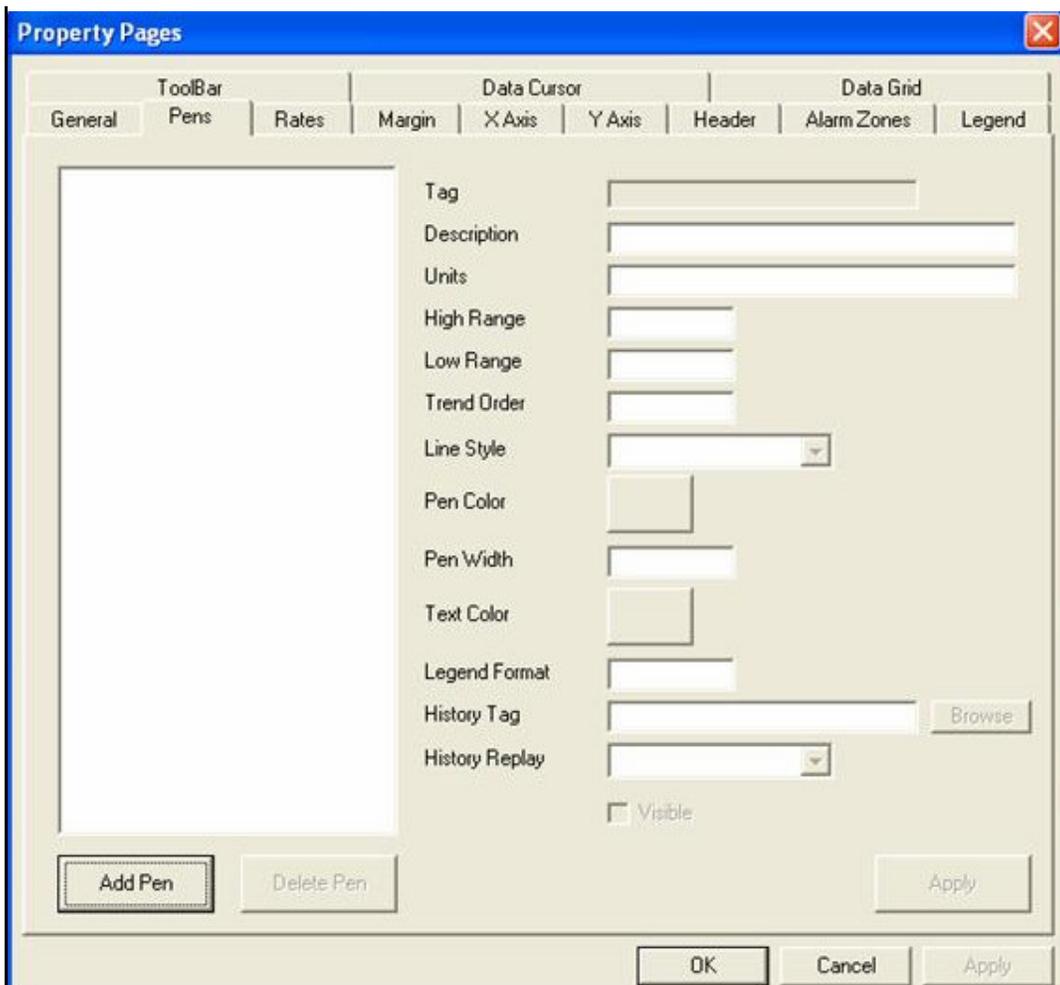


4 從 Controls 清單中勾選 OPC Trend ActiveX Control。



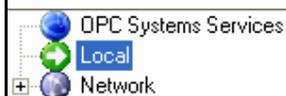
- | | |
|---|--|
| 5 | 從 ToolBox 中點選 OPC Trend ActiveX Control  圖示並加入 Form 中。 |
| 6 | 改變此 Trend Window 到想要的大小。

請注意，如果此 Trend Window 太小的話，有些的 Runtime 屬性在執行期間改變並不生效。 |
| 7 | 選擇 Trend Window 的屬性和 Pens 。 |



8 按下 Add Pen 按鈕。

在 Trend Point Tag 對話方塊中選取 Local OPC Systems Service



*注意：如果你想要設定的 Trend Window 不是與 OPC Systems Service 在同一電腦，則必須按下 Select 按鈕找到該 Network Node。

9 展開 Ramp Tag 並選取 Ramp 的 Value 參數，再按下左下角的 OK 按鈕。

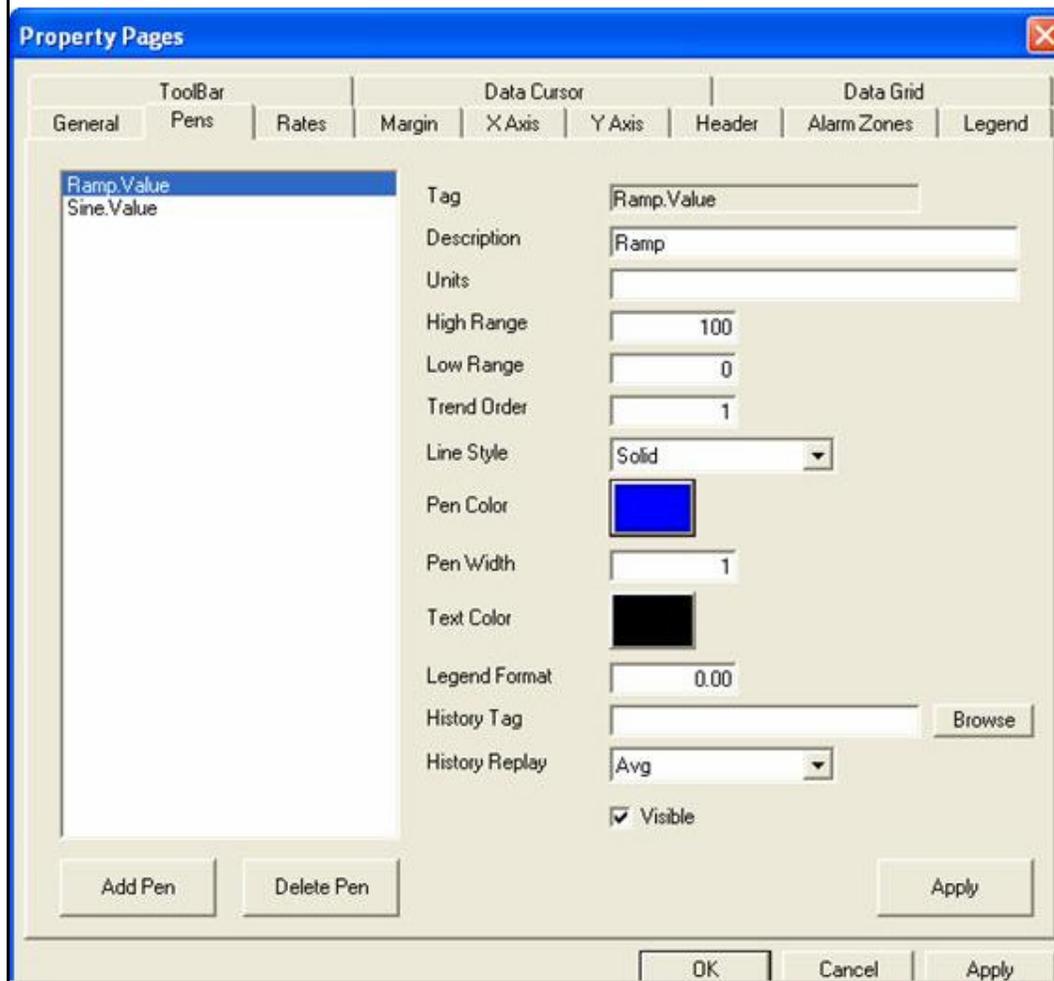


10 按下 Add Pen 按鈕，展開 Ramp Tag 並選取 Sine 的 Value 參數，再按下左下角

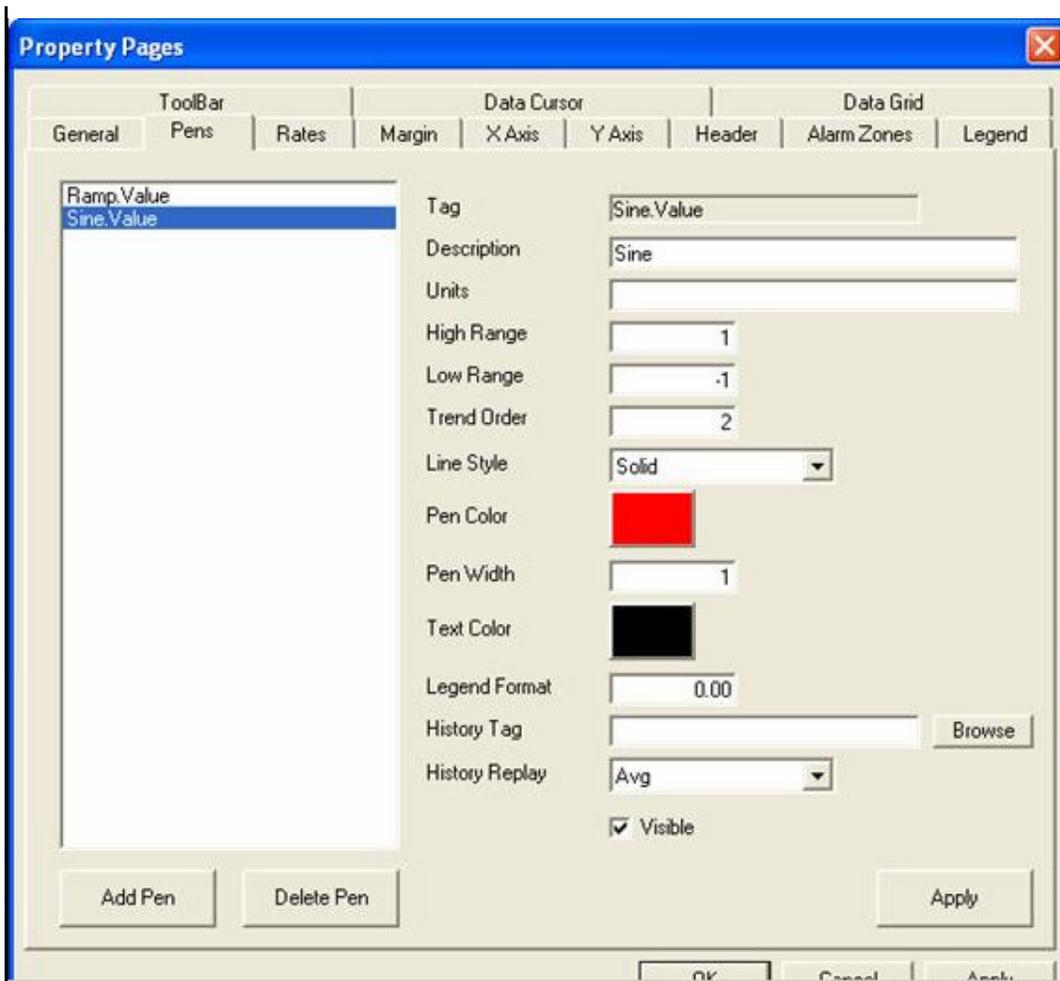
的 OK 按鈕。



- 11 從 Pen 清單中點選 Ramp.Value，並設定 Pen Color 為 Blue，然後按下 Apply 按鈕。



- 12 從 Pen 清單中點選 Sine.Value，並設定 High Range 為 1，Low Range 為-1，然後按下 Apply 按鈕。



13 點選 Rates 屬性頁籤並設定 TimeFrame 的值為 60 秒。

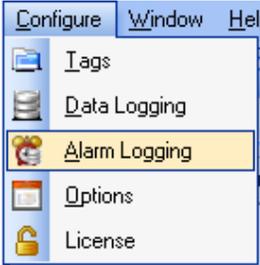
Sample Rate:	<input type="text" value="1"/>	Seconds
Time Frame:	<input type="text" value="60"/>	Seconds
Chart Percent Forward:	<input type="text" value="5"/>	Percent

14 按下屬性頁籤下方的 OK 按鈕。

15 儲存你的應用程式以準備執行。

第五章 Alarming

設定 Alarm Logging

Step	Task
1	啓動 Configure OPC Systems 應用程式。
2	選擇 Configure → Alarm Logging。 
3	選取 Local OPC Systems Service。 
4	鍵入 Simulation 在右上角的 Logging Group Name 欄位中。 
5	在 Common 頁籤中勾選 Logging Active。 
6	點選 Database 頁籤。 
7	爲 SQL Server 使用下列的設定。Database 和 Table 將會自動被建立。

Common Filters Nodes Database CSV Logging

Log To Database 

Provider

Server

Set Server Name with Tag

Database

Set Database Name with Tag

Table

Set Table Name with Tag

Use WinNT Authentication

User Name

Password

Server 的名稱必須將其從 localhost 調整成適當的 SQL Server engine 名稱。此名稱在第一次啟動 SQL Server Management Studio 的 Connect to Server 對話方盒中必須被找到。



如為 Microsoft Access，則使用下列設定。Database 和 Table 將會自動被建立。

Common
 Filters
 Nodes
 Database
 CSV Logging

Log To Database 

Provider:

Server:

Set Server Name with Tag

Database:

Set Database Name with Tag

Table:

Set Table Name with Tag

Use WinNT Authentication

User Name:

Password:

若為 CSV 文字檔，則使用下列設定。

Common
 Filters
 Nodes
 Database
 CSV Logging

Log To CSV File 

Path:

Set Path with Tag

File Name:

Set File Name with Tag

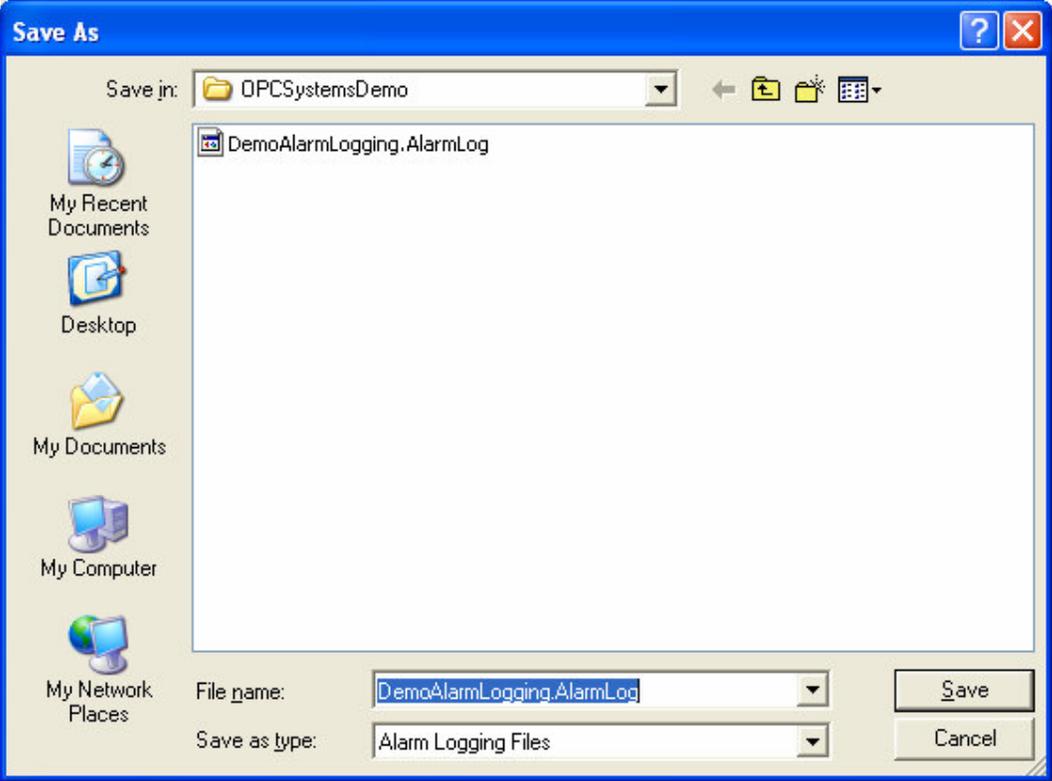
Append Date to File Name 

Append Hour to File Name

Append Minute to File Name

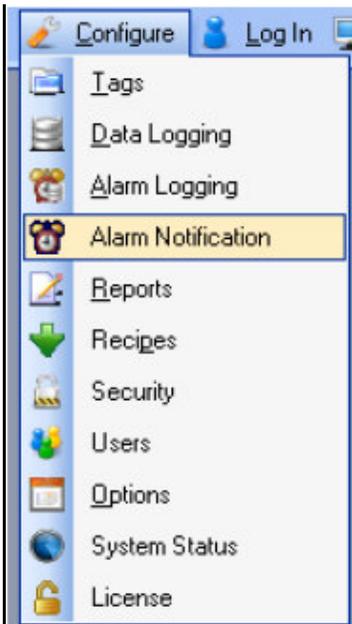
8 按下 Add 按鈕來加入 Alarm Logging group。

	
9	<p>按下上方的  按鈕來儲存剛才之設定。</p> 

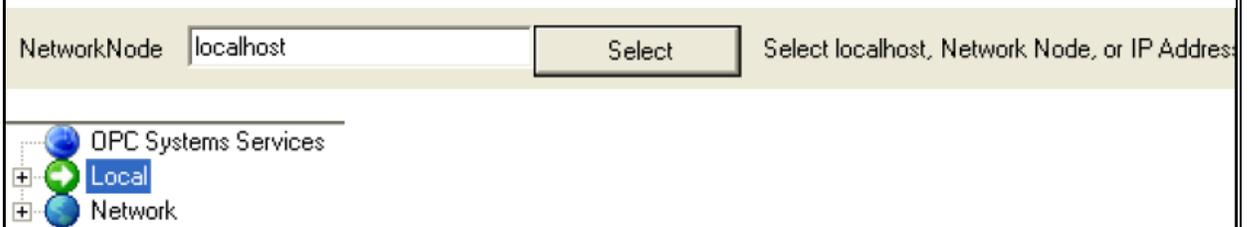
10	<p>以 DemoAlarmLogging.AlarmLog 為檔名儲存到 C:\OPCSystemsDemo 中。</p> 
----	--

設定 Alarm Notification

Step	Task
1	啟動 Configure OPC Systems 應用程式。
2	選擇 Configure->Alarm Notification。



3 直接按下 Select 按鈕來選擇 Local OPC Systems Service，或是在左下方的 Service Tree 中選取欲管理的 Network Node，然後再按下 Select 按鈕。



4 在右上方的 Notification Group Name 欄位中輸入 Simulation。

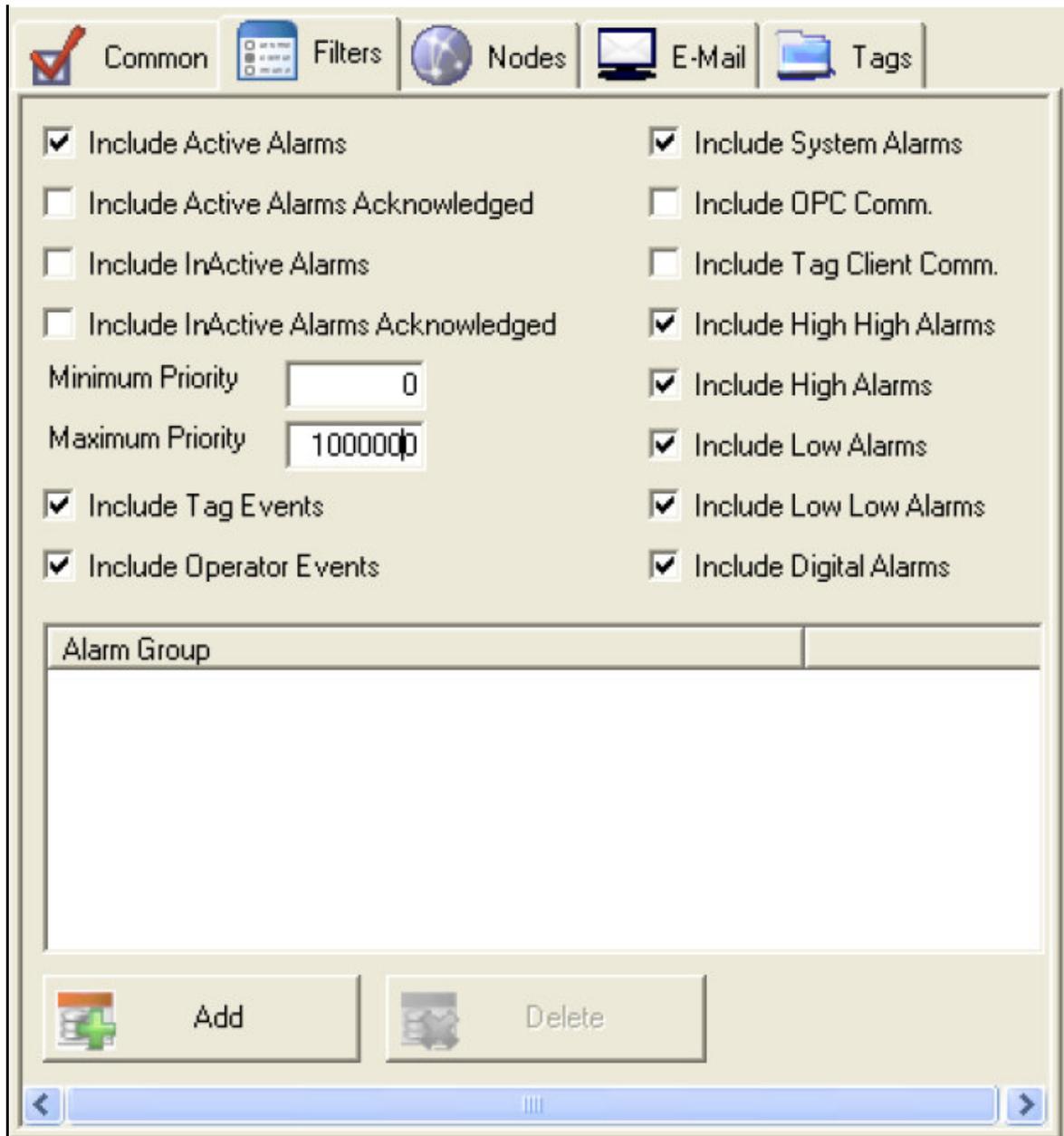
5 在 Common 屬性頁籤中勾選 Notification Active。



6 選擇 Filters 頁籤。



你可以依據 Alarm Priority、Alarm Groups 和 Alarm Types 來訂定過濾條件。



7 選擇 E-Mail 頁籤。



我們可以訂定當某過濾條件達到時寄送電子郵件給收件者。使用分號(;)區隔多個電子郵件地址，可以同時寄給多收件者。

Common
 Filters
 Nodes
 E-Mail
 Tags

Enable E-Mail 

From Address

To Addresses

CC Addresses

Outgoing Mail Server

Outgoing Port Number

Enable User Credentials

Username

Password

Subject

Message Body

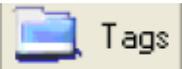
<input type="checkbox"/> Include Alarm ID	<input checked="" type="checkbox"/> Include Alarm Type
<input checked="" type="checkbox"/> Include Alarm Group	<input checked="" type="checkbox"/> Include Alarm Priority
<input checked="" type="checkbox"/> Include Alarm Text	<input checked="" type="checkbox"/> Include Alarm Active Status
<input checked="" type="checkbox"/> Include Alarm Acknowledged	<input type="checkbox"/> Include Time Delay
<input checked="" type="checkbox"/> Include Alarm Value	<input type="checkbox"/> Include Cleared Value
<input checked="" type="checkbox"/> Include Alarm Time	<input type="checkbox"/> Include Cleared Time
<input checked="" type="checkbox"/> Include Acknowledged Time	

Date / Time Format

Disable Sending Multiple Emails

Time Period Minutes

8 選擇 Tags 頁籤。



使用 Tags 屬性，我們可以指定去自動地寫入 alarm 統計資料到 OPC Systems.NET Tags。通常使用來驅動外部 alarm 喇叭，或是顯示共有多少個 alarms 是在生效當中。

Common
 Filters
 Nodes
 E-Mail
 Tags

Set Tag When Alarm Active

Set Tag When Alarm Active and Not Acknowledged

Set Tag When Alarm Active and Acknowledged

Set Tag When Alarm Not Active and Not Acknowledged

Set Tag When Alarm Not Active and Acknowledged

Set Integer Tag With Alarm Active Count

Set Integer Tag With Alarm Active and Not Acknowledged Count

Set Integer Tag With Alarm Active and Acknowledged Count

Set Integer Tag With Alarm Not Active and Not Acknowledged Count

Set Integer Tag With Alarm Not Active and Acknowledged Count

Set Integer Tag With Number Of Alarms In A Given Period

Time Period For Number Of Active Alarms: Hours

9 進入 Configure->Tags 並新增一個名稱命名為 Alarm Is Active 的 Boolean 點。新增一個名稱命名為 Number Of Alarms Active 的 Integer 點。新增一個名稱命名為 Number Of Alarms In Last 24 Hours 的 Integer 點。

10	<p>回到 Configure-Alarm Notification 的 Tags 頁籤。</p> <p>勾選 Set Tag When Alarm Active。指定 Tag 名稱爲 Alarm Is Active。</p> <p>勾選 Set Tag Integer Tag with Alarm Active Count。指定 Tag 名稱爲 Number Of Alarms Active。</p> <p>勾選 Set Tag Integer Tag with Number of Alarms In A Given Period。指定 Tag 名稱爲 Number Of Alarms In Last 24 Hours。</p>
11	<p>按下左下角的 Add 按鈕來新增 Alarm Notification Group。</p> 
12	<p>按下上方工具列的 Save 按鈕。</p> 
13	<p>儲存設定檔 DemoAlarmNotification.AlarmNotification 到 C:\OPCSystemsDemo。</p>
14	<p>進到 Configure->Tags 並選擇上面所建立的這幾個 Tag，看看目前的 alarm 狀態是否有在更新。</p>

Alarm Components

總共有三種 alarming components 可以被使用來呈現即時 alarm：

1. 使用在 WinForm 應用程式中的 OPC Alarm.NET 100% Managed component。
2. 使用在 Web 應用程式中的 OPC Web Alarm.NET 100% Managed ASP.NET component。
3. 使用在過時的應用程式（如 VB6）中的 OPC Alarm ActiveX control。

此所有 3 種 component 可以被執行在本機或遠端電腦中，來呈現從任何購買有 OPC Alarm.NET 或 OPC Web Alarm.NET 授權的 OPC Systems.NET 服務中所取得的即時資料。

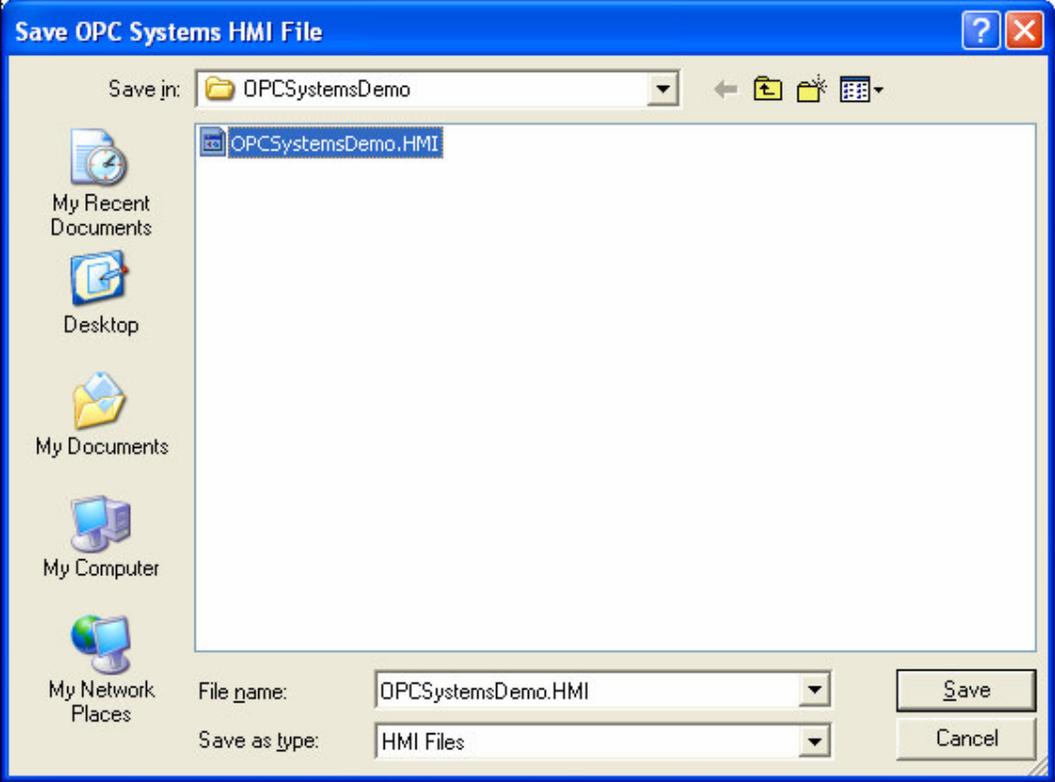
此所有 3 種 component 也都支援搭配 OPC Database.NET 授權的 OPC Systems.NET 服務，且設定有本章中 Configure Alarm Logging 所敘述的 alarm logging groups，來進行歷史 alarm 的重播。

OPC Systems HMI

OPC Systems HMI 允許我們直接使用 OPC Trend.NET 和 Alarm.NET component，而不必使用 Visual Studio 來另行開發程式。

Step	Task
------	------

1	啓動 OPC Systems HMI 應用程式。
2	選擇 File → New 並儲存檔名 OPCSystemsDemo 到 C:\OPCSystemsDemo\目錄。



設定 Alarm Window

OPC Alarm.NET component 可以直接被整合進任何 .NET container，較常用的是 Visual Studio.NET 或 OPC Systems HMI。

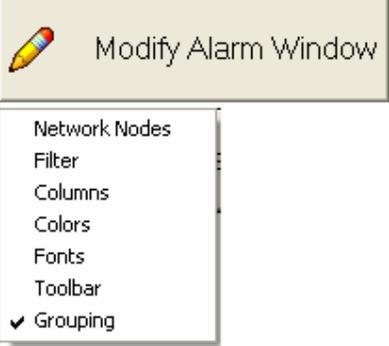
OPC Web Alarm.NET component 可以直接被整合進任何 ASP.NET Web 應用程式。

OPC Alarm ActiveX control 可以直接被整合進任何 ActiveX container，較常用的是 HMI 圖控軟體或是 Visual Studio 6.0。可參考本教學的 Configure Alarm ActiveX 部份。

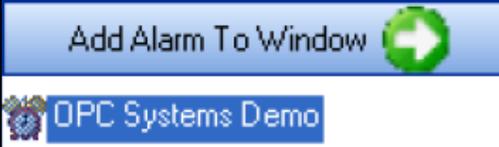
以下是使用 OPC Alarm.NET component 整合進 OPC Systems HMI .NET container 的範例。

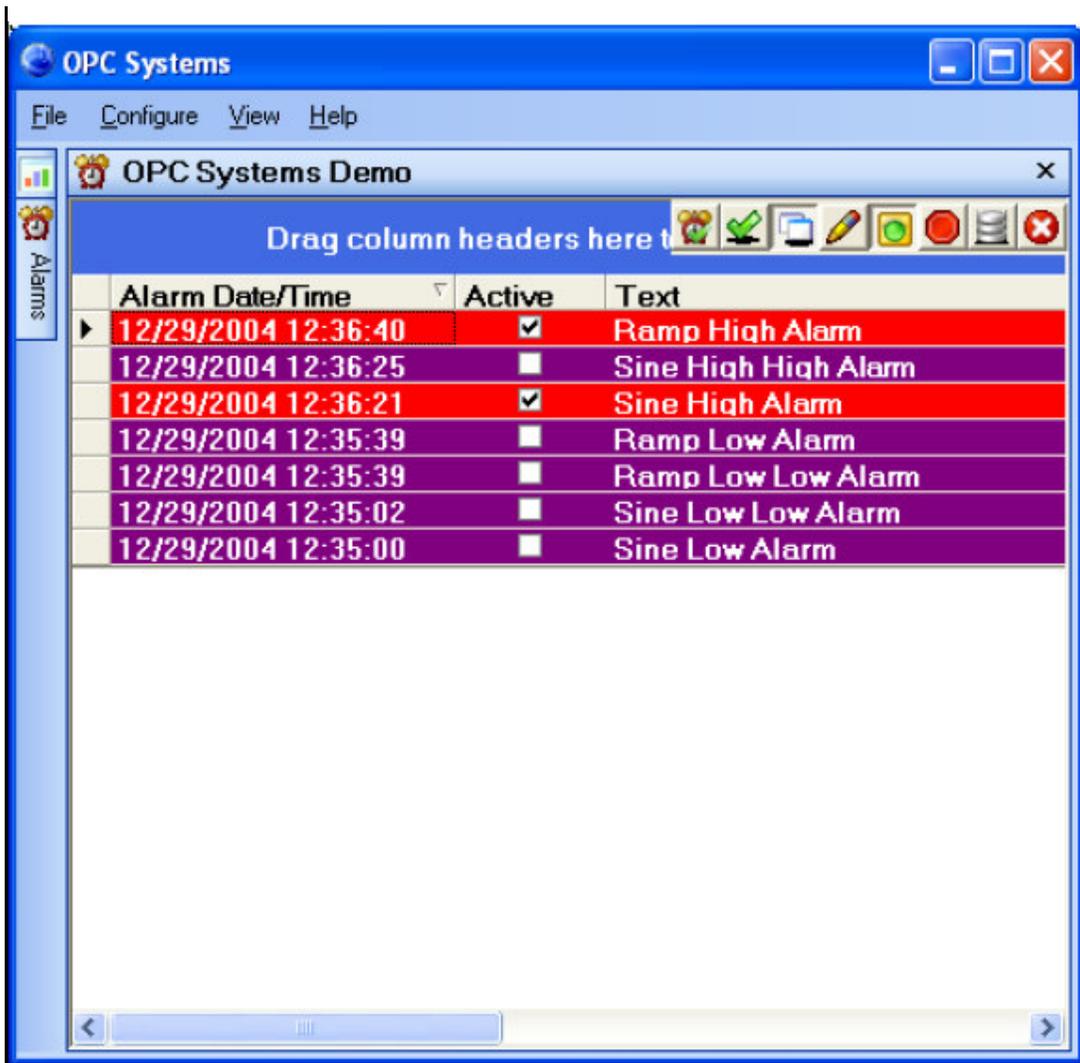
Step	Task
1	在 OPC Systems HMI 應用程式中按下左下角的 Alarm 按鈕。



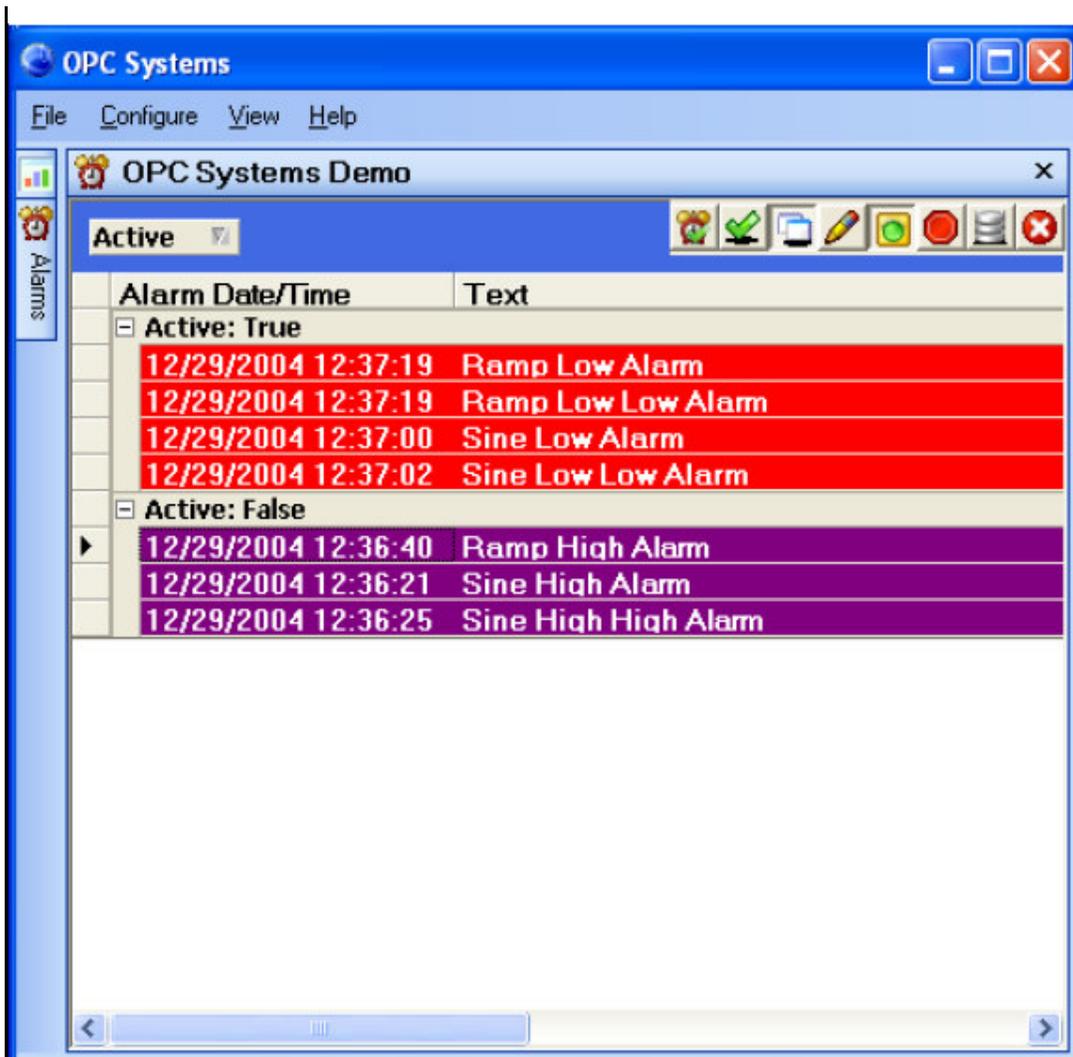
2	<p>按下左上方的 Add 按鈕。</p> 
3	<p>變更 Alarm Window Name 為 OPC Systems Demo 。</p>
4	<p>按下 Modify Alarm Window 按鈕，我們可以加入其他的 OPC Systems Nodes、修改 Filters、Colors 或其他屬性。</p>  <p>預設情況下，Network Nodes 是選定 localhost，Filters 包含所有 alarms，除了被 acknowledged 和不再作動之 alarms、OPC 通訊失敗、和 Tag Client 與其他 OPC Systems Services 通訊的部份。我們可以調整此 Filter，以我們想要的 Priority、Alarm Groups、和 Alarm Types。</p> <p>*注意：如果你想要設定的 Alarm Window 不是與 OPC Systems Service 在同一電腦，則必須按下 Select 按鈕找到該 Network Node。</p>
5	<p>按下左下角的 OK 按鈕。</p> 
6	<p>選擇 File → Save 來儲存設定。</p>  <p>這是當新增或刪除一個 Trend Window、Alarm Window 或 View Window 時是必要的。在 Trend 或 Alarm Window 中的修改會自動地儲存在自己的設定檔中。</p>
7	<p>此 Alarm Window 現在已經可以被 Run Alarm Window in OPC Systems HMI 部份來使用。</p>

執行 Alarm Window

Step	Task
1	如果 OPC Systems HMI 應用程式尚未啓動，則請執行之。
2	按下左下方的 Alarms 按鈕，並選擇 OPC Systems Demo Alarm Window，然後再按下 Add Alarm To Window 按鈕。 
3	如果 Alarm 沒有在幾秒內出現，請確認已為 OPCSystemsHMI.exe 和 OPCSystemsService.exe 打開 port 58723 防火牆限制。然後重新啓動 OPC Systems Service 試試看。
4	Alarm Window 應該如下圖般出現，並由 OPC Systems Service 即時更新。



5 點選 Active 欄位並拖曳至欄位名稱上方,以進行以 Active 或 Non-Active 來區分群組。



6 在 alarm 記錄上連接滑鼠左鍵二次可 acknowledge 該 alarm。根據最新的過濾條件，當某個 alarm 已經不再 active 且已 acknowledged 過，則會從 Alarm Window 中自動消失。

按下 Ack All 按鈕就可 acknowledge 在 Alarm Window 中的所有 alarms。

7 如果想要儲存目前設定的 Window 名稱和設定，則請選擇 File->Save 來儲存。所有 Window 屬性的修改將會自動保留，即使沒有再次選擇 File->Save。僅僅只有當新增一個 Window 或移除時，才需要再次選擇 File->Save。

8 如果你在此範例中已設定 Alarm Logging 完成，則可在 Alarm Window 中按下滑鼠右鍵並選擇 History 來取用歷史 alarms。



9 使用 Alarm Window 工具列上的按鈕，並在 Alarm Window 上按下滑鼠右鍵來瀏覽可使用的功能。

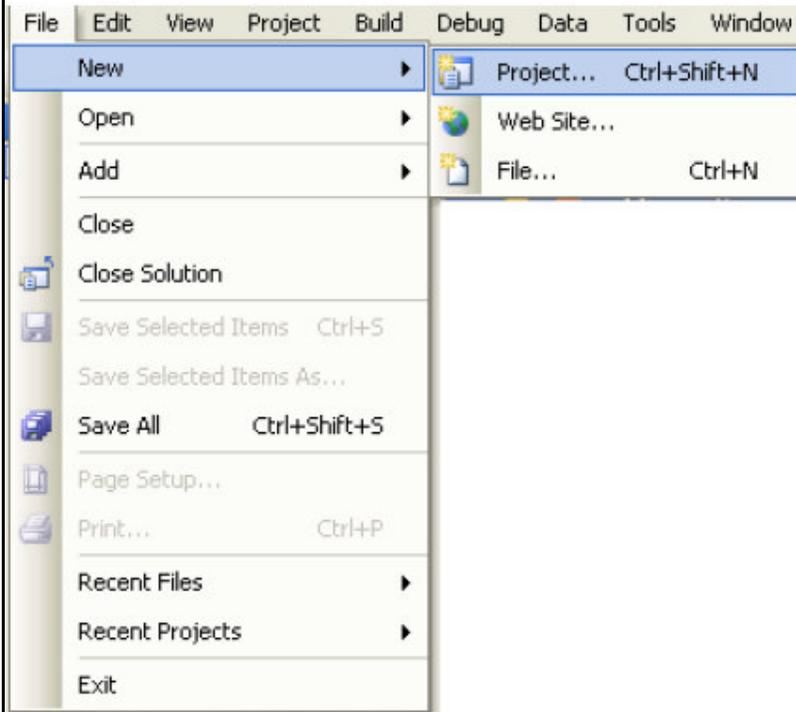


新增 Alarm Window 到 Visual Studio 應用程式

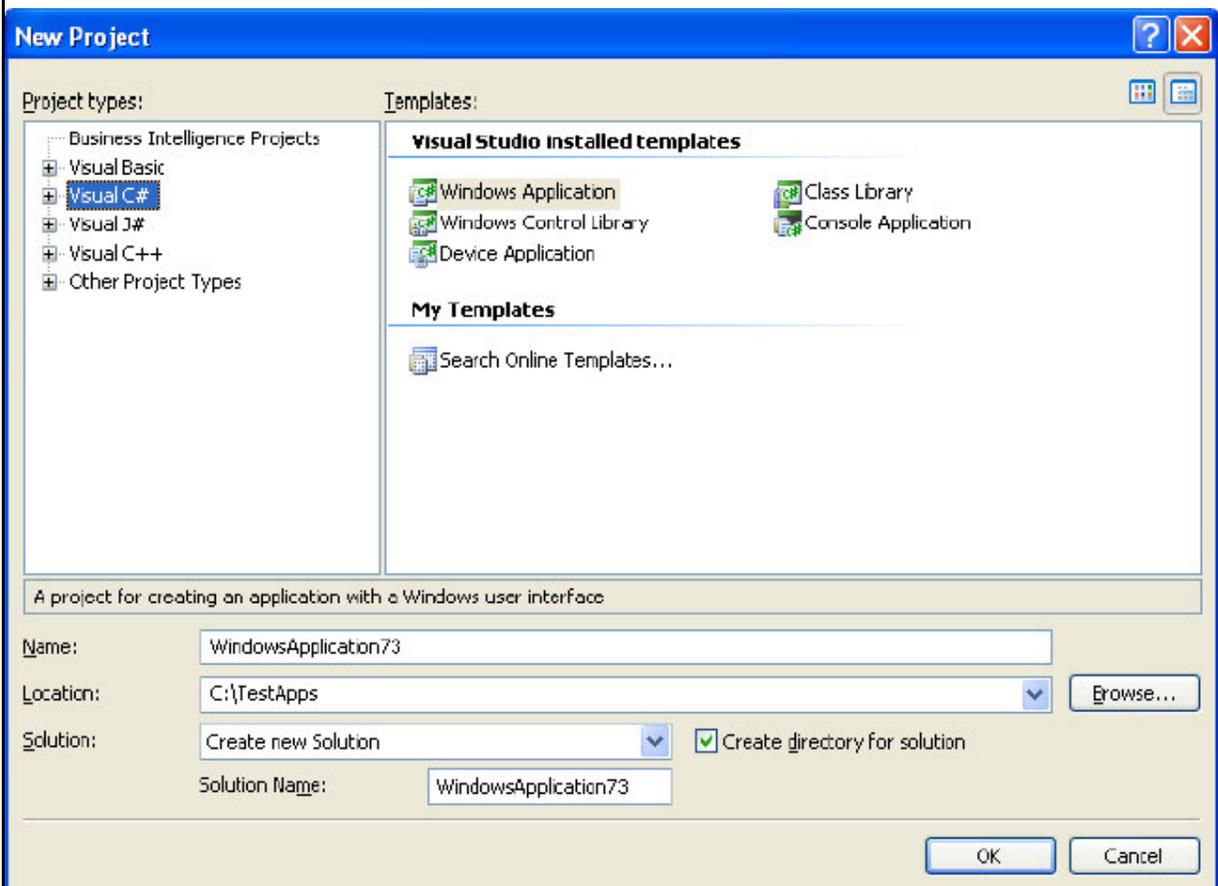
以下的步驟可以被用來新增 Alarm Window 到一個 C#、C++ 或 Visual Basic.NET 應用程式中。Alarm Window 的所有屬性都可以利用程式來讀取和修改。此例子展示不撰寫程式碼的方式來建立 Alarm Window。

Step	Task
1	啟動 Visual Studio 2005 或 Visual Studio 2008，並選擇 File->New->Project 來建立新的 C#、C++ 或

Visual Basic.NET 應用程式專案。

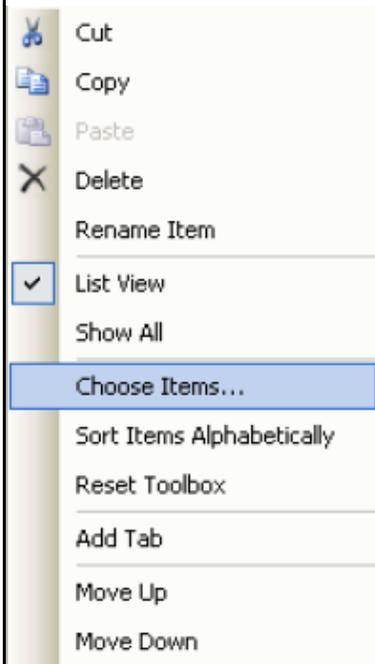


2 選擇 Windows Application 作為專案類型。

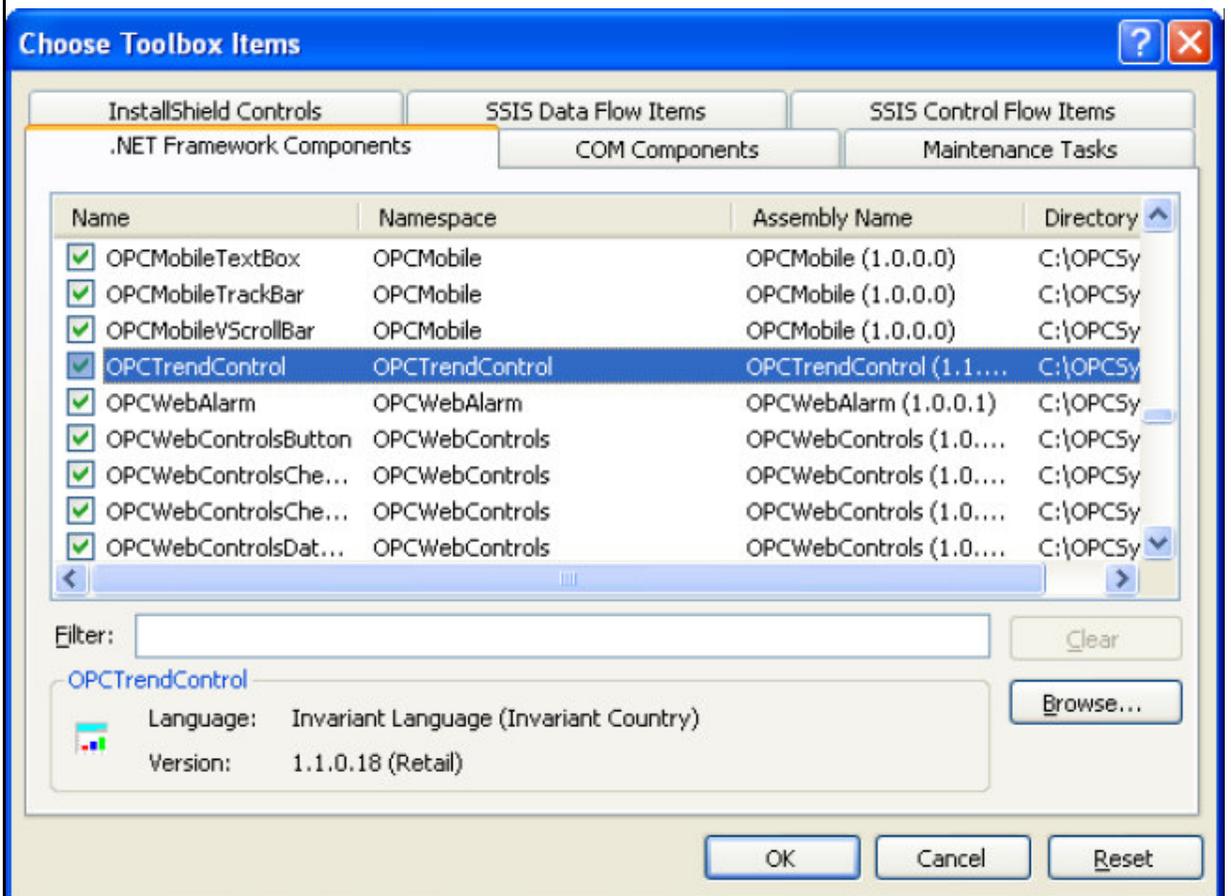


3 如果 OPCAlarmControl 尚未存在於工具箱中，則請在工具箱中按下滑鼠右鍵並選取 Choose

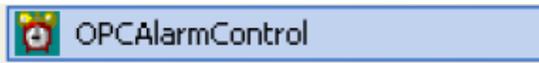
Items...



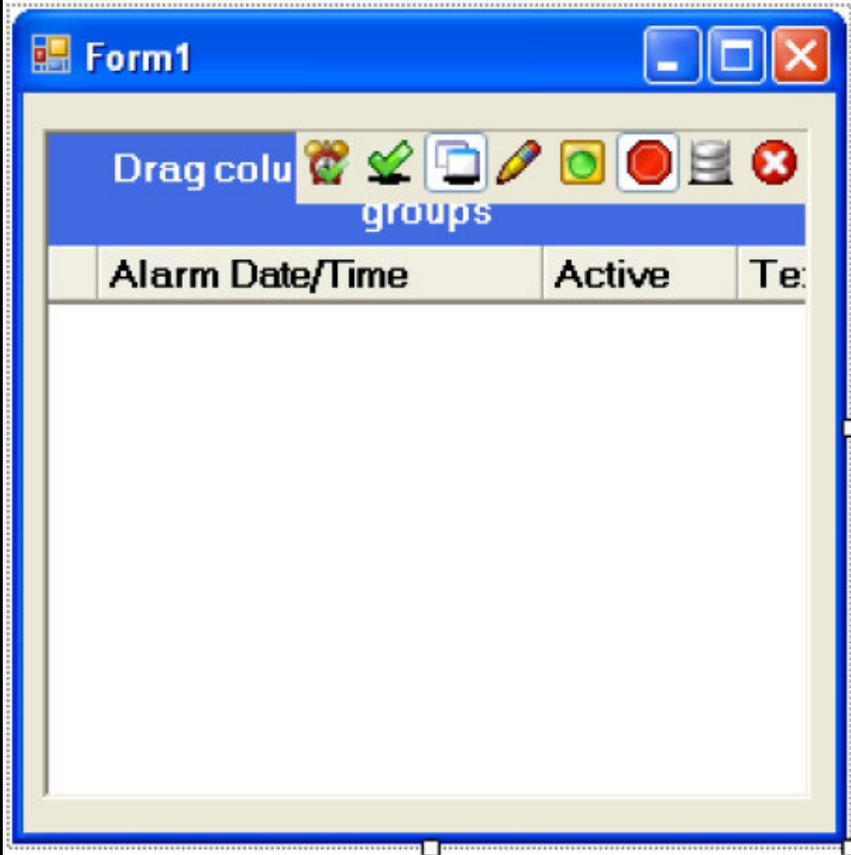
從 .NET Framework Components 中勾選 OPCAlarmControl 並按下 OK 按鈕。



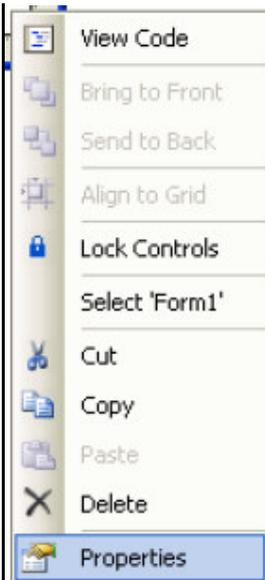
4 新增 OPCAlarmControl component 到 Form 中。



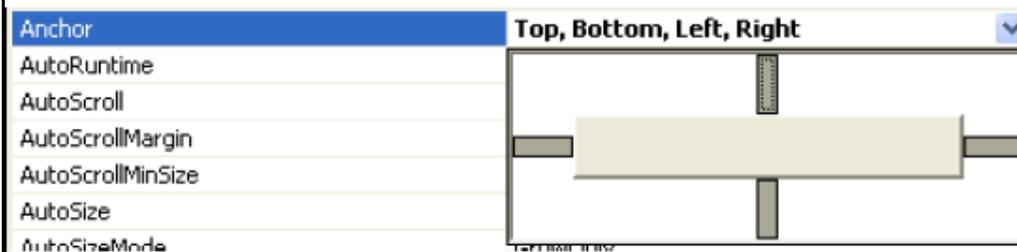
調整 Form 和 Alarm Window 的大小至適當的。



5 在 Alarm Window 上按下滑鼠右鍵並選取 Properties。



6 設定 Anchor 屬性為 Top, Bottom, Left, Right。



7 展開 AlarmFilter 屬性並設定想要的過濾設定。

AlarmFilter	Filter 0 to 1000000
AlarmActive	True
AlarmActiveAcked	True
AlarmNotActive	True
AlarmNotActiveAcked	False
⊕ Groups	String[] Array
IncludeDigital	True
IncludeHigh	True
IncludeHighHigh	True
IncludeLow	True
IncludeLowLow	True
IncludeOPC	False
IncludeSystem	True
IncludeTagClient	False
MaximumPriority	1000000
MinimumPriority	0

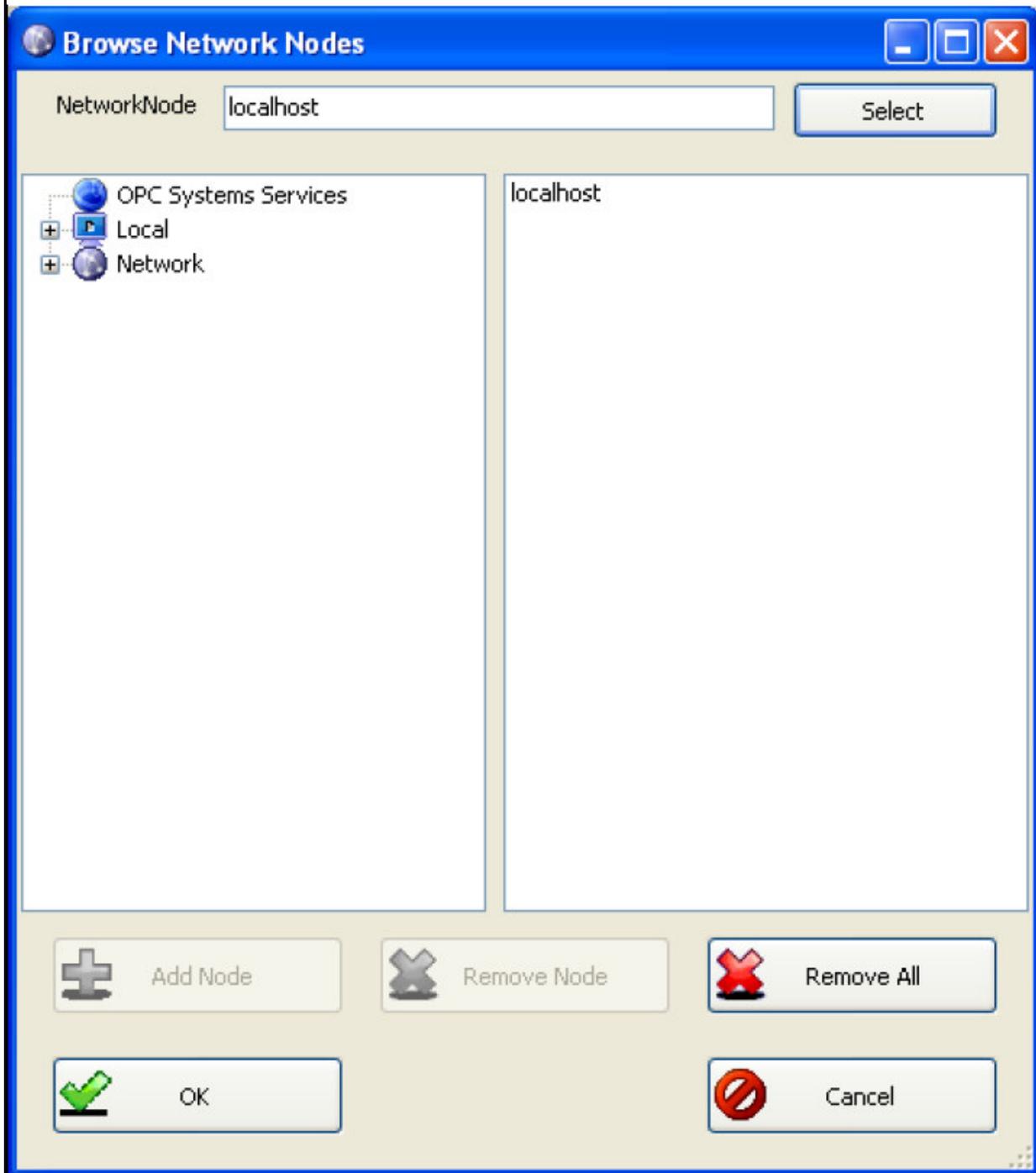
8

選擇 AlarmNetworkNodes 屬性並點按右方  按鈕。



9

選擇所有想要將 alarm 出現在 Alarm Window 中的電腦名稱。



10

如果我們想要讓操作者在執行期間對 Trend Window 的設定進行保留，請設定 ConfigurationFile

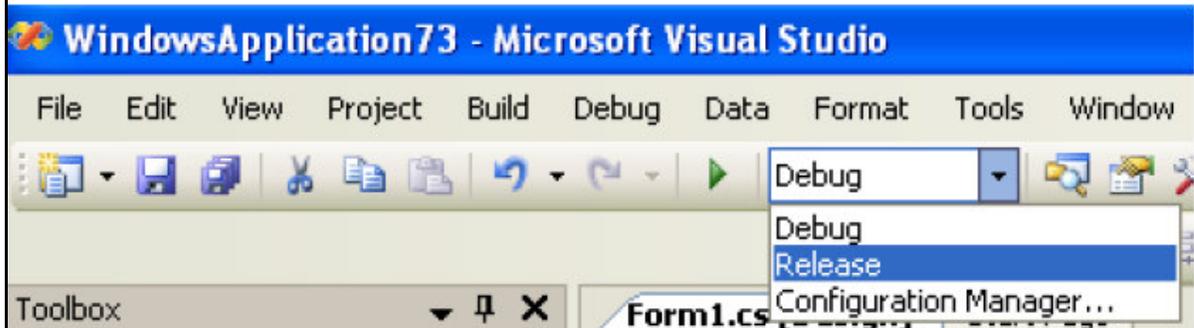
屬性到一個有效的檔案路徑。

ConfigurationFile

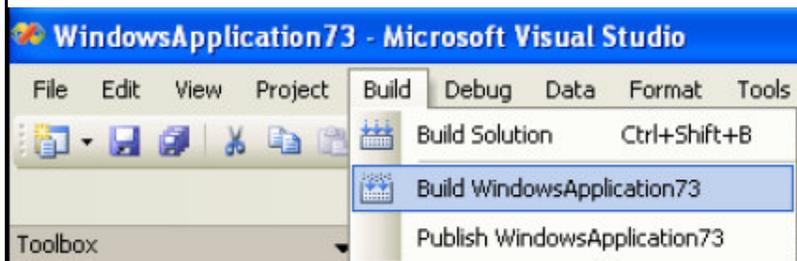
注意：如果你設定了此一屬性到某一個檔案，則請確實將該檔案放置於你指定的目錄中。

如果我們希望在每次應用程式重啟動時都維持最初的原始設定，則讓此屬性保留為空白。

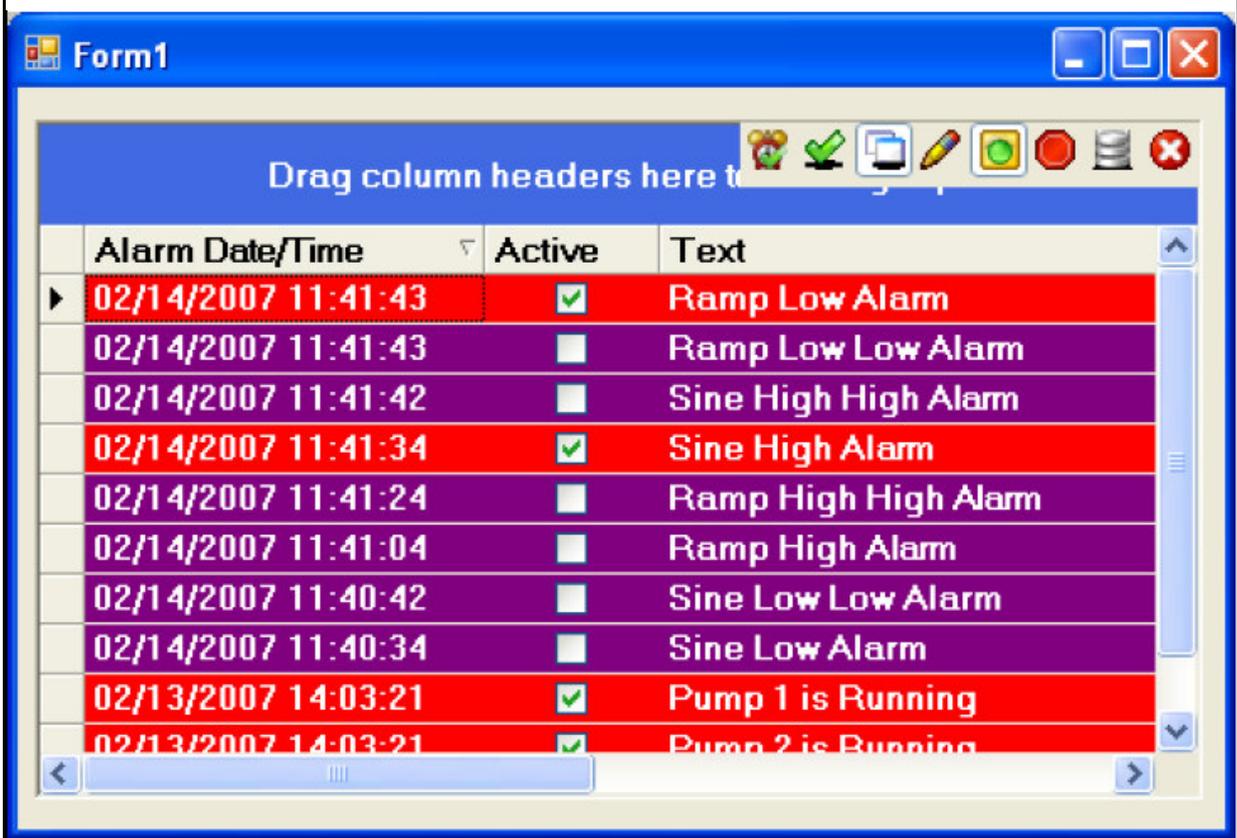
11 設定 Visual Studio 工具列中的 compile mode 為 Release。



12 從 Visual Studio 的 Build 功能表中選擇 Build 此應用程式。



13 使用檔案總管來瀏覽放置在 bin\Release 目錄中的執行檔並執行。

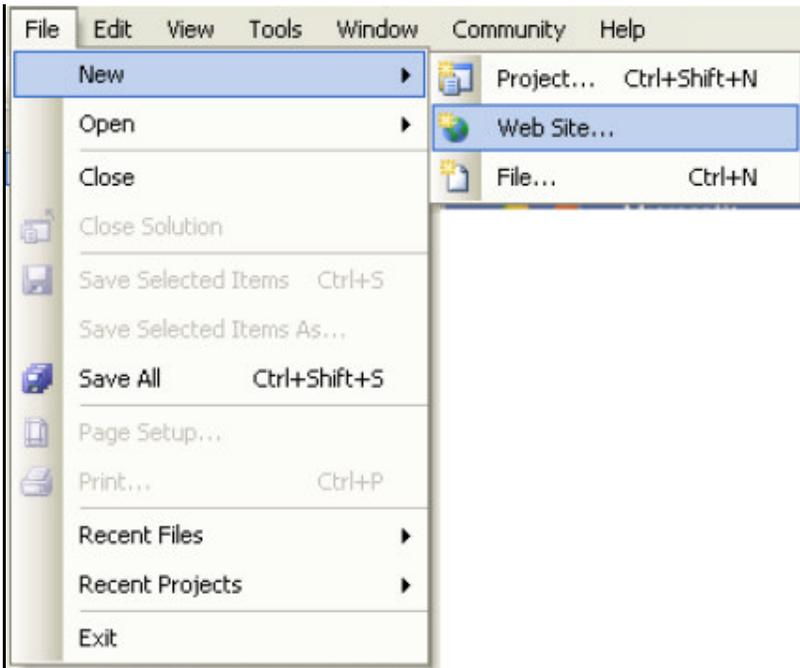


- 14 欲發佈此應用程式到遠端電腦中，則請確定在步驟 9 的 AlarmNetworkNodes 選擇中，是選擇網路上的電腦或是 IP 位址。然後複製放置在 bin\Release 目錄中的所有檔案至目的端電腦。

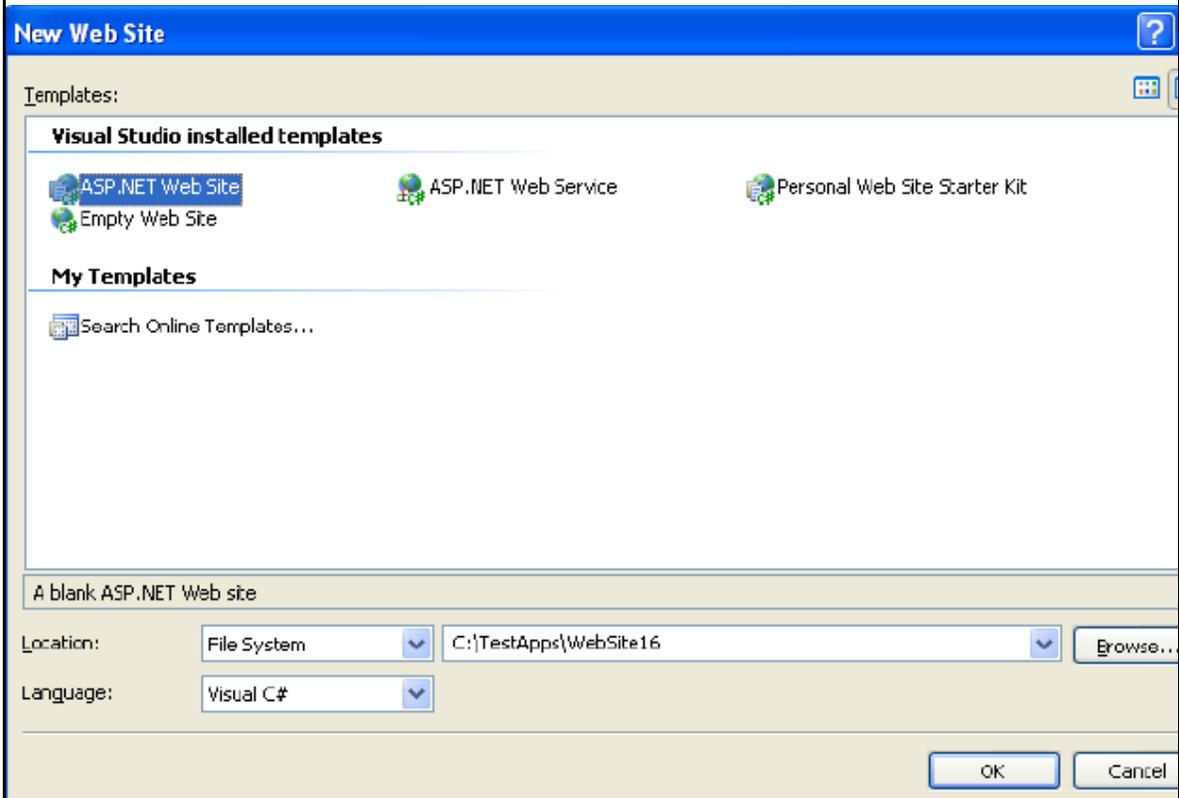
新增 Alarm Window 到 ASP.NET Web 應用程式

以下的步驟可以被用來新增 Alarm Window 到一個 C#、C++或 Visual Basic.NET Web 應用程式中。Alarm Window 的所有屬性都可以利用程式來讀取和修改。此例子展示不撰寫程式碼的方式來建立 Alarm Window。

Step	Task
1	啓動 Visual Studio 2005 或 Visual Studio 2008，並選擇 File->New->Web Site 來建立新的 C#、J#或 VB ASP.NET Web 應用程式專案。

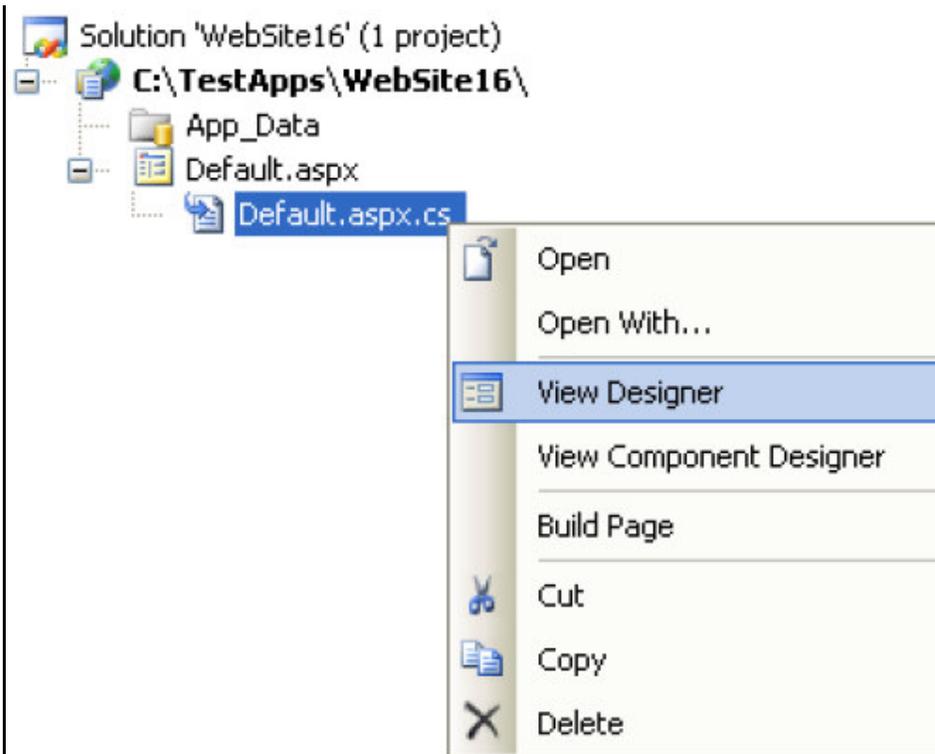


2 選擇 ASP.NET Web Site 專案類型，並指定左下方所欲使用的程式語言。

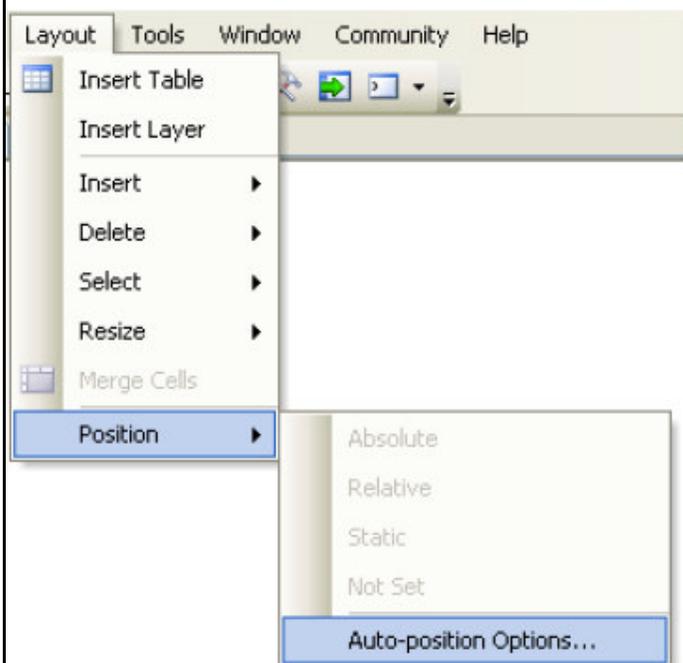


3 展開方案管理員中的 Default.aspx 網頁，並選取 View Designer。

注：如果你沒有看到方案管理員，則請選擇功能表 View->Solution Explorer。

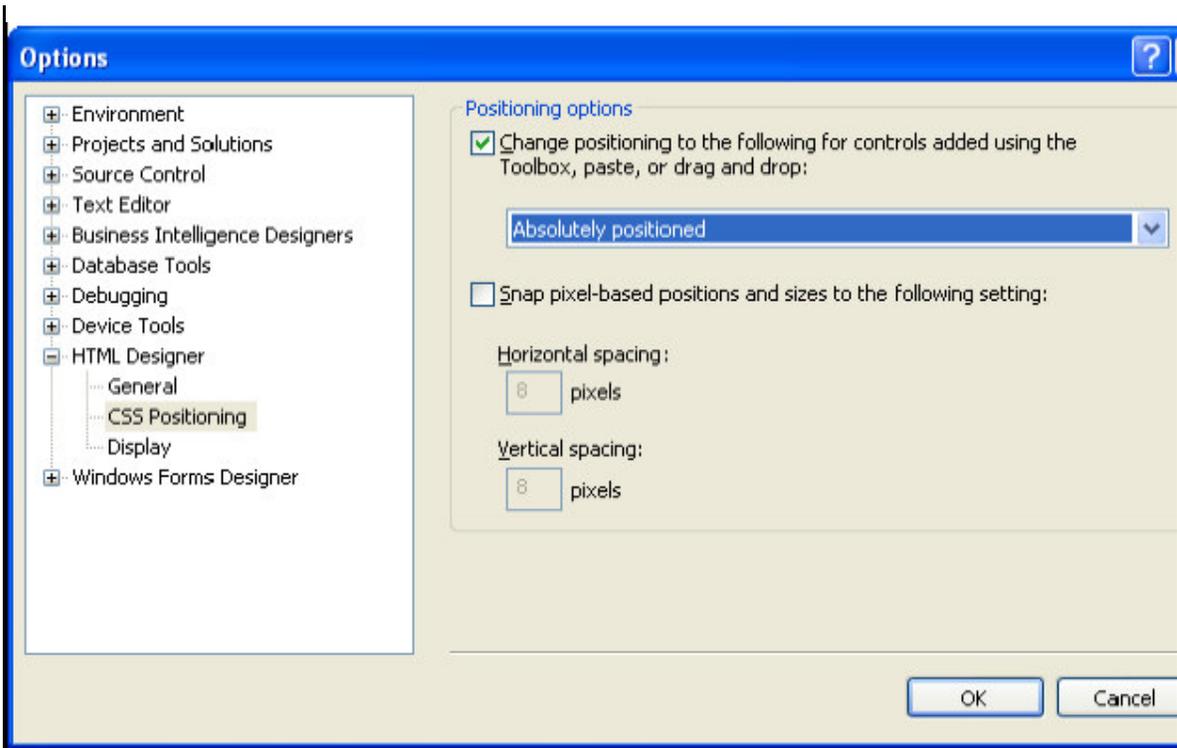


4 從 Visual Studio 功能表中選擇 Layout->Position->Auto-position 選項。

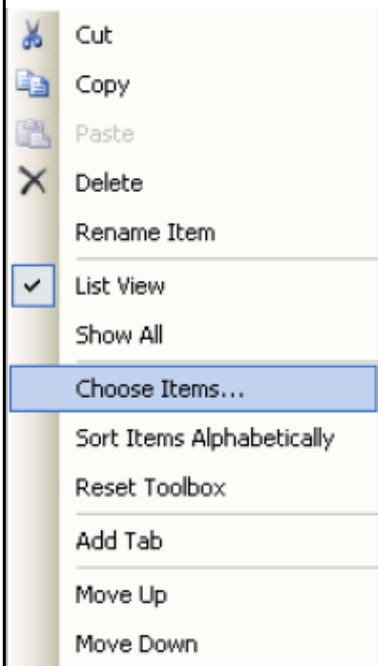


5 在 HTML Designer 中設定 CSS Positioning 為 Absolutely positioned。

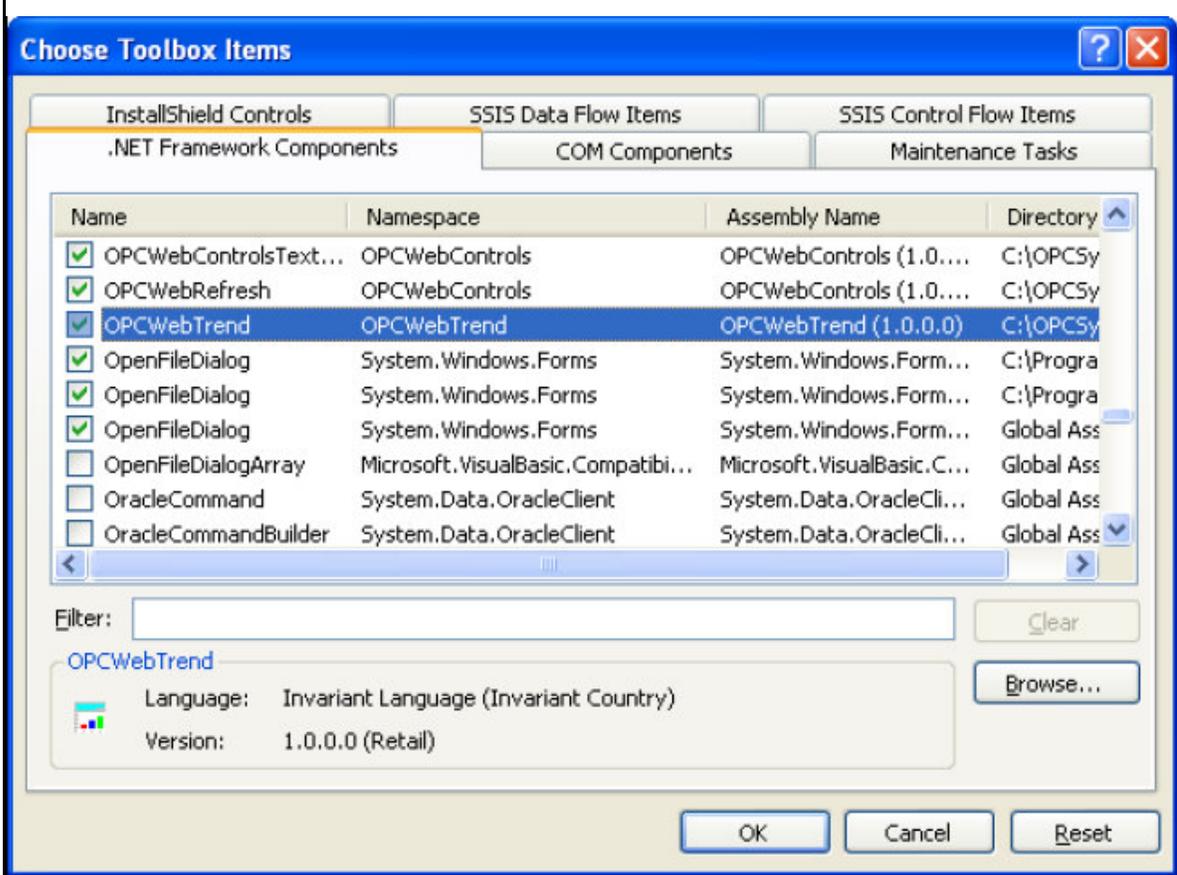
注：我們可以使用任何想要使用的定位型態。



- 6 如果 OPCWebAlarm、WebControlsLabel 和 OPCWebRefresh 尚未出現在工具箱中，則請在工具箱中按下滑鼠右鍵並選擇 Choose Items 來選取。



從 .NET Framework Components 中勾選 OPCWebAlarm、OPCWebControlsLabel 和 OPCWebRefresh 並按下 OK 按鈕。



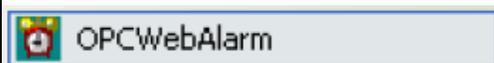
7 新增一個 OPCWebRefresh 控制項至 WebForm 中。



```
[OPCWebRefresh "OPCWebRefresh1"]
```

為確定專案的 web.config 檔已適當地修改為能夠執行 AJAX runtime，新增一個 OPCWebControlsLabel 到 WebForm 中，然後再刪除之。

8 新增 OPCWebAlarm 控制項至 WebForm 中。



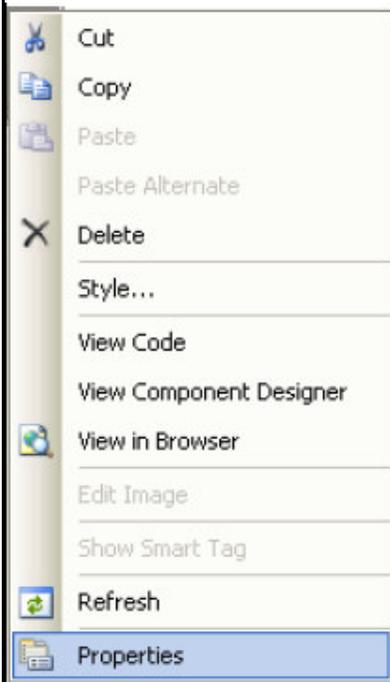
調整 Alarm Window 到想要的大小。

[OPCWebRefresh "OPCWebRefresh1"]

Ack All RealTime History

DateTime	Text	Type	AlarmValue
2/14/2007 11:50:48 AM	Active Alarm Text	High High	1
2/14/2007 11:50:48 AM	Acknowledged Active Alarm Text	High High	1
2/14/2007 11:50:48 AM	InActive Alarm Text	High High	1

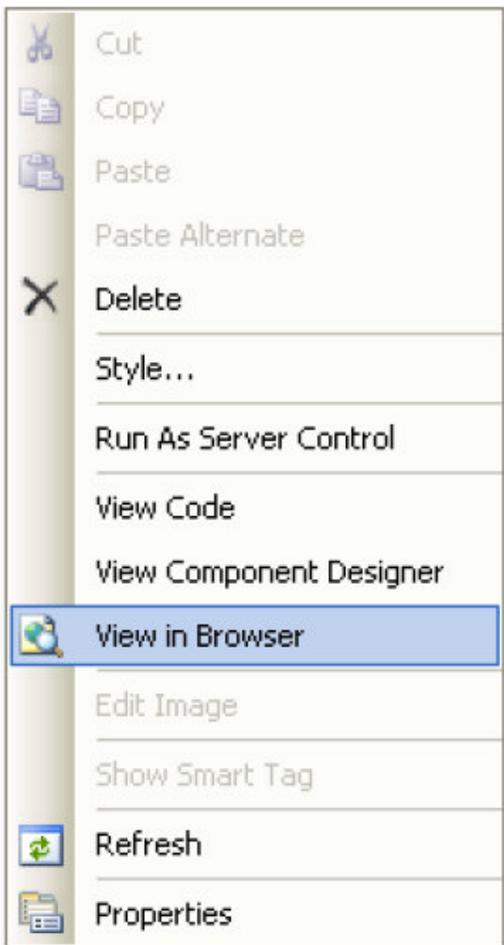
9 在 Alarm Window 上按滑鼠右鍵並選擇 Properties。

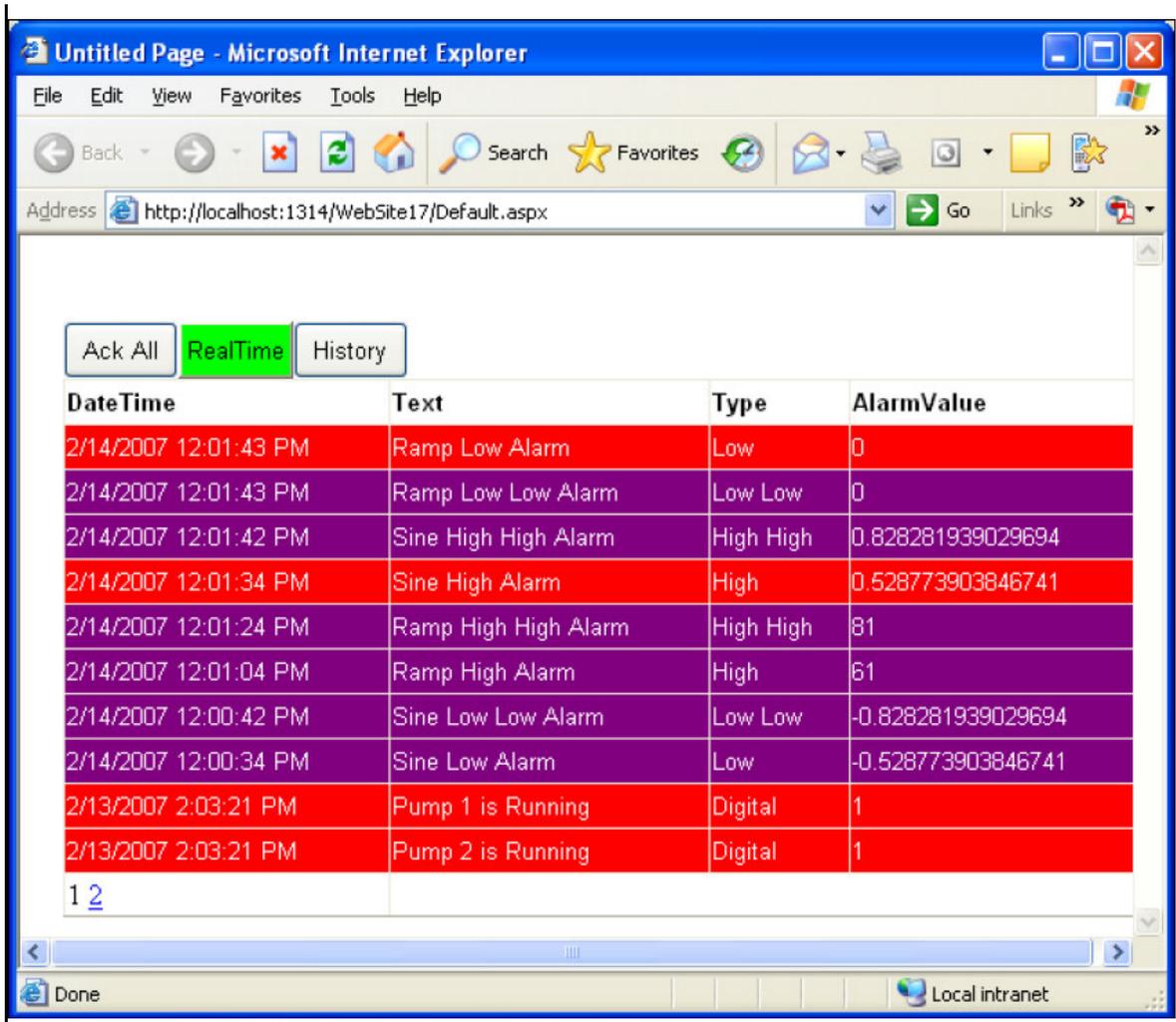


10 選擇 Alarm 型態至需要的設定。

	<table border="1"> <tbody> <tr> <td>AlarmActive</td> <td>True</td> </tr> <tr> <td>AlarmActiveAked</td> <td>True</td> </tr> <tr> <td>AlarmActiveAkedBackColor</td> <td> Yellow</td> </tr> <tr> <td>AlarmActiveAkedFont</td> <td>Arial, 10pt</td> </tr> <tr> <td>AlarmActiveAkedForeColor</td> <td> Black</td> </tr> <tr> <td>AlarmActiveBackColor</td> <td> Red</td> </tr> <tr> <td>AlarmActiveFont</td> <td>Arial, 10pt</td> </tr> <tr> <td>AlarmActiveForeColor</td> <td> White</td> </tr> <tr> <td>AlarmNetworkNodes</td> <td>localhost</td> </tr> <tr> <td>AlarmNotActive</td> <td>True</td> </tr> <tr> <td>AlarmNotActiveAked</td> <td>False</td> </tr> <tr> <td>AlarmNotActiveAkedBackColor</td> <td> Lime</td> </tr> <tr> <td>AlarmNotActiveAkedFont</td> <td>Arial, 10pt</td> </tr> <tr> <td>AlarmNotActiveAkedForeColor</td> <td> Black</td> </tr> <tr> <td>AlarmNotActiveBackColor</td> <td> Purple</td> </tr> <tr> <td>AlarmNotActiveFont</td> <td>Arial, 10pt</td> </tr> <tr> <td>AlarmNotActiveForeColor</td> <td> White</td> </tr> </tbody> </table>	AlarmActive	True	AlarmActiveAked	True	AlarmActiveAkedBackColor	 Yellow	AlarmActiveAkedFont	Arial, 10pt	AlarmActiveAkedForeColor	 Black	AlarmActiveBackColor	 Red	AlarmActiveFont	Arial, 10pt	AlarmActiveForeColor	 White	AlarmNetworkNodes	localhost	AlarmNotActive	True	AlarmNotActiveAked	False	AlarmNotActiveAkedBackColor	 Lime	AlarmNotActiveAkedFont	Arial, 10pt	AlarmNotActiveAkedForeColor	 Black	AlarmNotActiveBackColor	 Purple	AlarmNotActiveFont	Arial, 10pt	AlarmNotActiveForeColor	 White
AlarmActive	True																																		
AlarmActiveAked	True																																		
AlarmActiveAkedBackColor	 Yellow																																		
AlarmActiveAkedFont	Arial, 10pt																																		
AlarmActiveAkedForeColor	 Black																																		
AlarmActiveBackColor	 Red																																		
AlarmActiveFont	Arial, 10pt																																		
AlarmActiveForeColor	 White																																		
AlarmNetworkNodes	localhost																																		
AlarmNotActive	True																																		
AlarmNotActiveAked	False																																		
AlarmNotActiveAkedBackColor	 Lime																																		
AlarmNotActiveAkedFont	Arial, 10pt																																		
AlarmNotActiveAkedForeColor	 Black																																		
AlarmNotActiveBackColor	 Purple																																		
AlarmNotActiveFont	Arial, 10pt																																		
AlarmNotActiveForeColor	 White																																		
11	<p>使用 AlarmNetworkNodes 屬性來指定所有想要接收 alarm 的 OPC Systemes Service 來源。</p> <table border="1"> <tbody> <tr> <td>AlarmNetworkNodes</td> <td>localhost</td> </tr> </tbody> </table>	AlarmNetworkNodes	localhost																																
AlarmNetworkNodes	localhost																																		
12	<p>指定想要在 Alarm Window 中被看到的 alarm 欄位。</p> <table border="1"> <tbody> <tr> <td>ColumnAkedDateTimeText</td> <td>AkedDateTime</td> </tr> <tr> <td>ColumnAkedDateTimeVisible</td> <td>False</td> </tr> <tr> <td>ColumnAkedText</td> <td>Aked</td> </tr> <tr> <td>ColumnAkedVisible</td> <td>False</td> </tr> </tbody> </table>	ColumnAkedDateTimeText	AkedDateTime	ColumnAkedDateTimeVisible	False	ColumnAkedText	Aked	ColumnAkedVisible	False																										
ColumnAkedDateTimeText	AkedDateTime																																		
ColumnAkedDateTimeVisible	False																																		
ColumnAkedText	Aked																																		
ColumnAkedVisible	False																																		
13	<p>使用 Groups、Include、Max 和 Min 等屬性來過濾那些型態的 alarm 要在 Alarm Window 中被看到。</p>																																		

	<table border="1"> <tr> <td colspan="2" data-bbox="256 142 821 184">Groups</td> <td data-bbox="1382 142 1421 184" style="text-align: right;">...</td> </tr> <tr> <td data-bbox="256 191 821 226">HeaderFont</td> <td data-bbox="821 191 1421 226">Arial, 10pt, style=Bold</td> <td></td> </tr> <tr> <td data-bbox="256 233 821 268">HeaderRowHeight</td> <td data-bbox="821 233 1421 268">20</td> <td></td> </tr> <tr> <td data-bbox="256 275 821 310">Height</td> <td data-bbox="821 275 1421 310">300px</td> <td></td> </tr> <tr> <td data-bbox="256 317 821 352">IncludeDigitalAlarms</td> <td data-bbox="821 317 1421 352">True</td> <td></td> </tr> <tr> <td data-bbox="256 359 821 394">IncludeHighAlarms</td> <td data-bbox="821 359 1421 394">True</td> <td></td> </tr> <tr> <td data-bbox="256 401 821 436">IncludeHighHighAlarms</td> <td data-bbox="821 401 1421 436">True</td> <td></td> </tr> <tr> <td data-bbox="256 443 821 478">IncludeLowAlarms</td> <td data-bbox="821 443 1421 478">True</td> <td></td> </tr> <tr> <td data-bbox="256 485 821 520">IncludeLowLowAlarms</td> <td data-bbox="821 485 1421 520">True</td> <td></td> </tr> <tr> <td data-bbox="256 527 821 562">IncludeOPCAlarms</td> <td data-bbox="821 527 1421 562">False</td> <td></td> </tr> <tr> <td data-bbox="256 569 821 604">IncludeSystemAlarms</td> <td data-bbox="821 569 1421 604">True</td> <td></td> </tr> <tr> <td data-bbox="256 611 821 646">IncludeTagClientAlarms</td> <td data-bbox="821 611 1421 646">False</td> <td></td> </tr> <tr> <td data-bbox="256 653 821 688">MaximumAlarms</td> <td data-bbox="821 653 1421 688">50</td> <td></td> </tr> <tr> <td data-bbox="256 695 821 730">MaximumPriority</td> <td data-bbox="821 695 1421 730">1000000</td> <td></td> </tr> <tr> <td data-bbox="256 737 821 772">MinimumPriority</td> <td data-bbox="821 737 1421 772">0</td> <td></td> </tr> </table>	Groups		...	HeaderFont	Arial, 10pt, style=Bold		HeaderRowHeight	20		Height	300px		IncludeDigitalAlarms	True		IncludeHighAlarms	True		IncludeHighHighAlarms	True		IncludeLowAlarms	True		IncludeLowLowAlarms	True		IncludeOPCAlarms	False		IncludeSystemAlarms	True		IncludeTagClientAlarms	False		MaximumAlarms	50		MaximumPriority	1000000		MinimumPriority	0	
Groups		...																																												
HeaderFont	Arial, 10pt, style=Bold																																													
HeaderRowHeight	20																																													
Height	300px																																													
IncludeDigitalAlarms	True																																													
IncludeHighAlarms	True																																													
IncludeHighHighAlarms	True																																													
IncludeLowAlarms	True																																													
IncludeLowLowAlarms	True																																													
IncludeOPCAlarms	False																																													
IncludeSystemAlarms	True																																													
IncludeTagClientAlarms	False																																													
MaximumAlarms	50																																													
MaximumPriority	1000000																																													
MinimumPriority	0																																													
14	<p data-bbox="256 814 1214 846">如果你希望限制 Alarm Window 的大小，則請設定 PageEnable 屬性為 True。</p> <table border="1" data-bbox="256 888 1421 1012"> <tr> <td colspan="2" data-bbox="256 888 821 930">PageEnable</td> <td data-bbox="1382 888 1421 930" style="text-align: right;">False</td> </tr> <tr> <td data-bbox="256 936 821 972">PageMode</td> <td data-bbox="821 936 1421 972">NumericPages</td> <td></td> </tr> <tr> <td data-bbox="256 978 821 1012">PageSize</td> <td data-bbox="821 978 1421 1012">10</td> <td></td> </tr> </table>	PageEnable		False	PageMode	NumericPages		PageSize	10																																					
PageEnable		False																																												
PageMode	NumericPages																																													
PageSize	10																																													
15	<p data-bbox="256 1035 935 1066">在 WebForm 上按下滑鼠右鍵並選擇 View in Browser。</p>																																													





- 16 設定此應用程式為 Release 模式，修改 web.config 檔以確認 Debug 參數是設為 False，並且 Build 此應用程式。通常 Web 應用程式檔案都會被放在 Inetpub\wwwroot 目錄中。
- 參考 OPC Systems.NET help 檔中的 OPC Web Alarm.NET 部份來了解特殊的考慮和開發步驟。

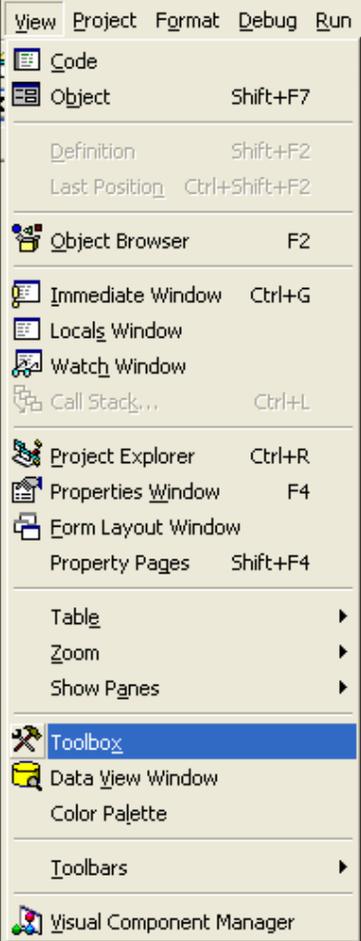
新增 Alarm ActiveX 控制項到過時的應用程式

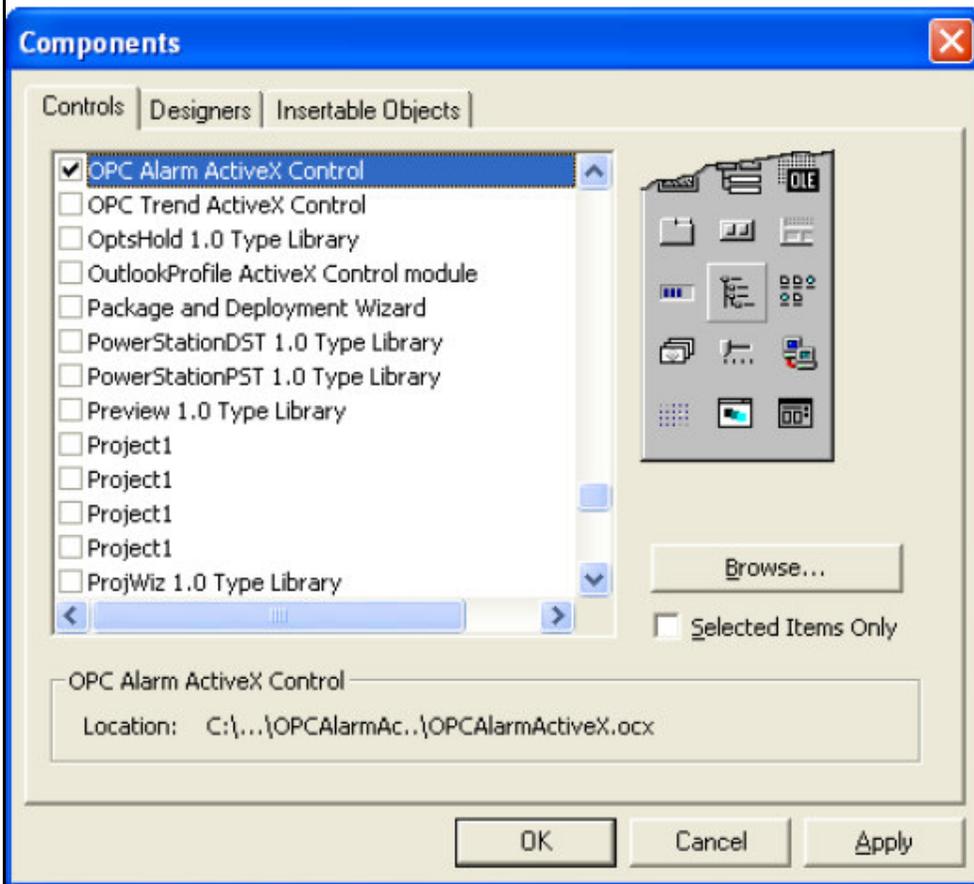
OPC Alarm ActiveX 控制項可以被直接整合進任何 ActiveX container，較常用的是 HMI 圖控軟體或是 Visual Studio 6.0。

注：ActiveX 控制項並不是 100% managed 程式，所以不能被使用在 Visual Studio.NET 應用程式中。

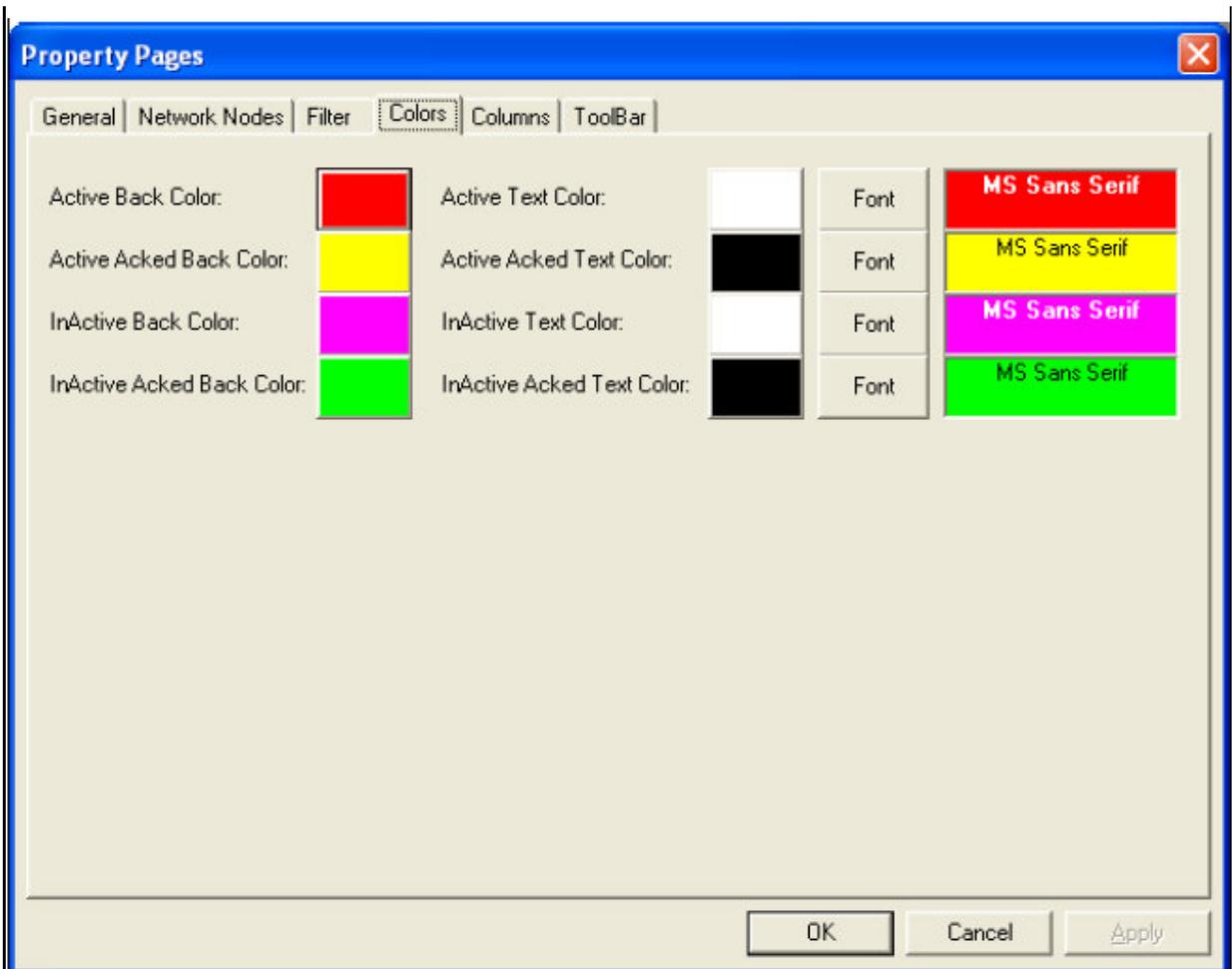
以下是使用 OPC Alarm ActiveX 控制項整合進 Visual Basic 6.0 或是其他 ActiveX container 程式的範例。

Step	Task
------	------

1	<p>啓動 Visual Basic 6.0 的專案，或是其他的 ActiveX container 程式。</p> <p>如果是使用其他的 ActiveX container 而不是 Visual Studio ，則只要選擇該程式中適當的方法，以選擇 OPCAlarmActiveX.OPCAlarm 來加入此 ActiveX control ，並跳至步驟 6 。</p>
2	<p>在 Visual Studio 6.0 開發環境中選擇 View → Toolbox 來開啓 Toolbox 。</p>  <p>The screenshot shows the 'View' menu in Visual Studio 6.0. The menu items are: Code, Object (Shift+F7), Definition (Shift+F2), Last Position (Ctrl+Shift+F2), Object Browser (F2), Immediate Window (Ctrl+G), Locals Window, Watch Window, Call Stack..., Project Explorer (Ctrl+R), Properties Window (F4), Form Layout Window, Property Pages (Shift+F4), Table, Zoom, Show Panes, Toolbox (highlighted), Data View Window, Color Palette, Toolbars, and Visual Component Manager.</p>
3	<p>在 Toolbox 上按右鍵並選擇 Components 。</p>  <p>The screenshot shows a context menu for the Toolbox. The menu items are: Components... (highlighted), Add Tab..., Dockable (checked), and Hide.</p>
4	<p>從 Controls 清單中勾選 OPC Alarm ActiveX Control 。</p>



- | | |
|---|---|
| 5 | 從 ToolBox 中點選 OPC Alarm ActiveX Control  圖示並加入 Form 中。 |
| 6 | 改變此 Alarm Window 到想要的大小。
請注意，如果此 Alarm Window 太小的話，有些的 Runtime 屬性在執行期間改變並不生效。 |
| 7 | 選擇並設定 Alarm Window 的各種屬性。 |



預設情況下，Network Nodes 是選定 localhost，Filters 包含所有 alarms，除了被 acknowledged 和不再作動之 alarms、OPC 通訊失敗、和 Tag Client 與其他 OPC Systems Services 通訊的部份。我們可以調整此 Filter，以我們想要的 Priority、Alarm Groups、和 Alarm Types。

*注意：如果你想要設定的 Alarm Window 不是與 OPC Systems Service 在同一電腦，則必須按下 Select 按鈕找到該 Network Node。

- | | |
|----|--|
| 8 | 按下 Property Pages 上的 OK 按鈕。 |
| 9 | 展開 Ramp Tag 並選取 Ramp 的 Value 參數，再按下左下角的 OK 按鈕。 |
| 10 | 儲存你的應用程式，如準備執行則加以編譯。 |

第六章 Human Machine Interface

HMI Components

總共有三種 HMI components 可以被使用來顯示和變更 Tag 設定中的資料：

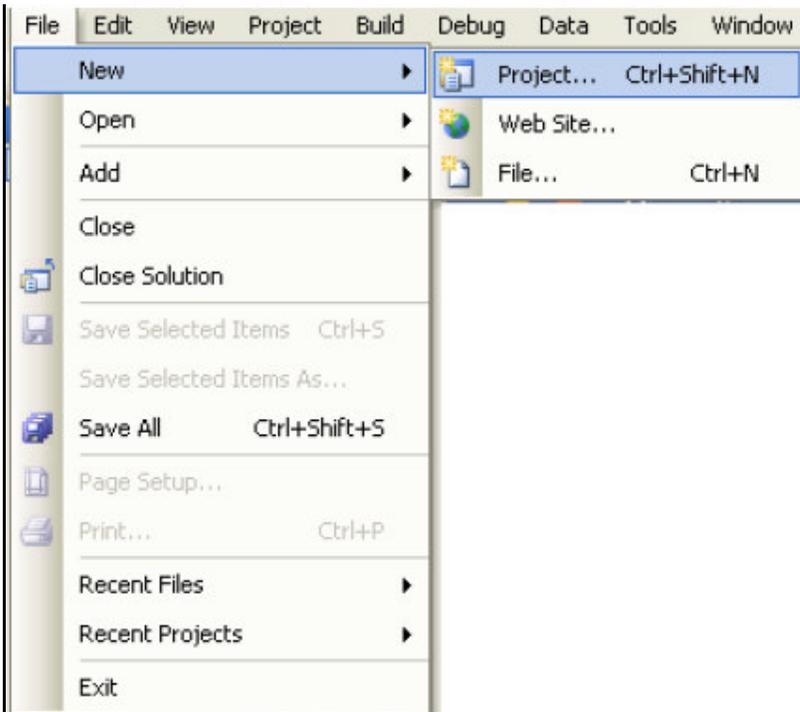
1. 使用在 WinForm 應用程式中的 OPC Controls.NET 100% Managed component 。
2. 使用在 Web 應用程式中的 OPC Web Controls.NET 100% Managed ASP.NET component 。
3. 使用在 Mobile 2003 和 5.0 中的 Pocket PC 應用程式中的 OPC Mobile.NET 100% Managed component 。

此所有 3 種 component 可以被執行在本機或遠端電腦中，來呈現從任何購買有相對應的產品授權的 OPC Systems.NET 服務中所取得的即時資料。

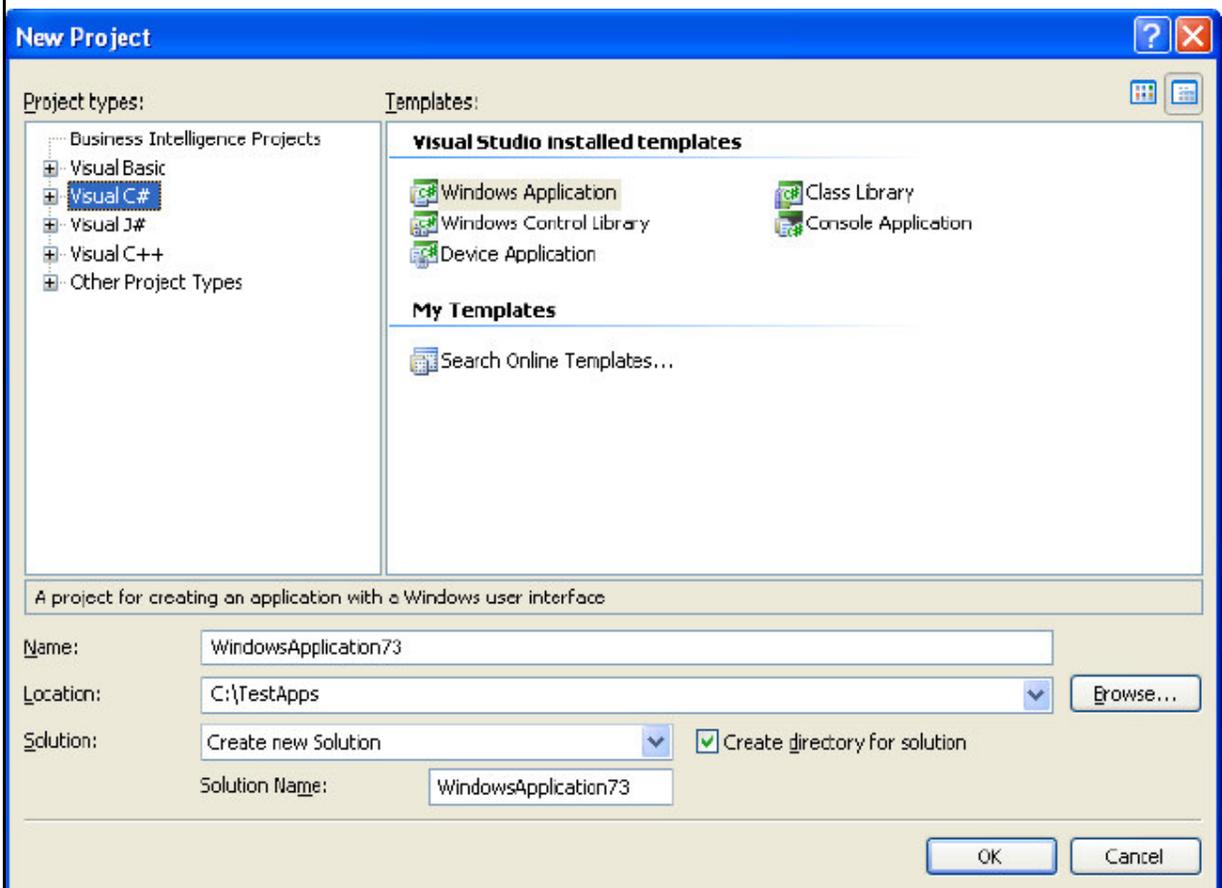
新增 HMI Controls 到 Visual Studio 應用程式

以下的步驟可以被用來新增視覺物件到一個 C#、C++或 Visual Basic.NET 應用程式中。請參考 VB.NET 範例程式來知道如何使用 OPC Controls。OPC Controls 的所有屬性都可以利用程式來讀取和修改。此例子展示不撰寫程式碼的方式來使用 OPC Controls。

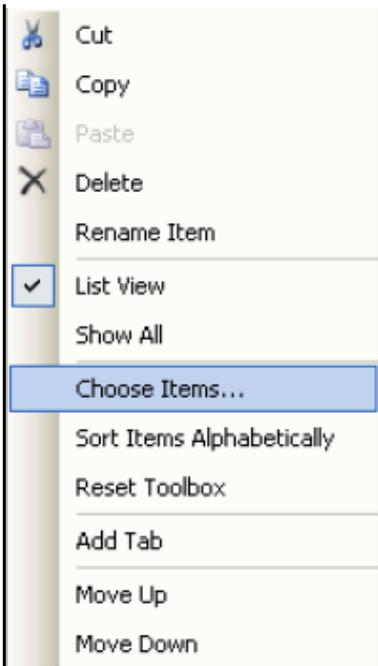
Step	Task
1	如果你已經取代設定為自己建立的 Tags，則請重新載入預設的 DemoTags 設定檔。 啓動 Visual Studio 2005 或 Visual Studio 2008，並選擇 File->New->Project 來建立新的 C#、C++或 Visual Basic.NET 應用程式專案。



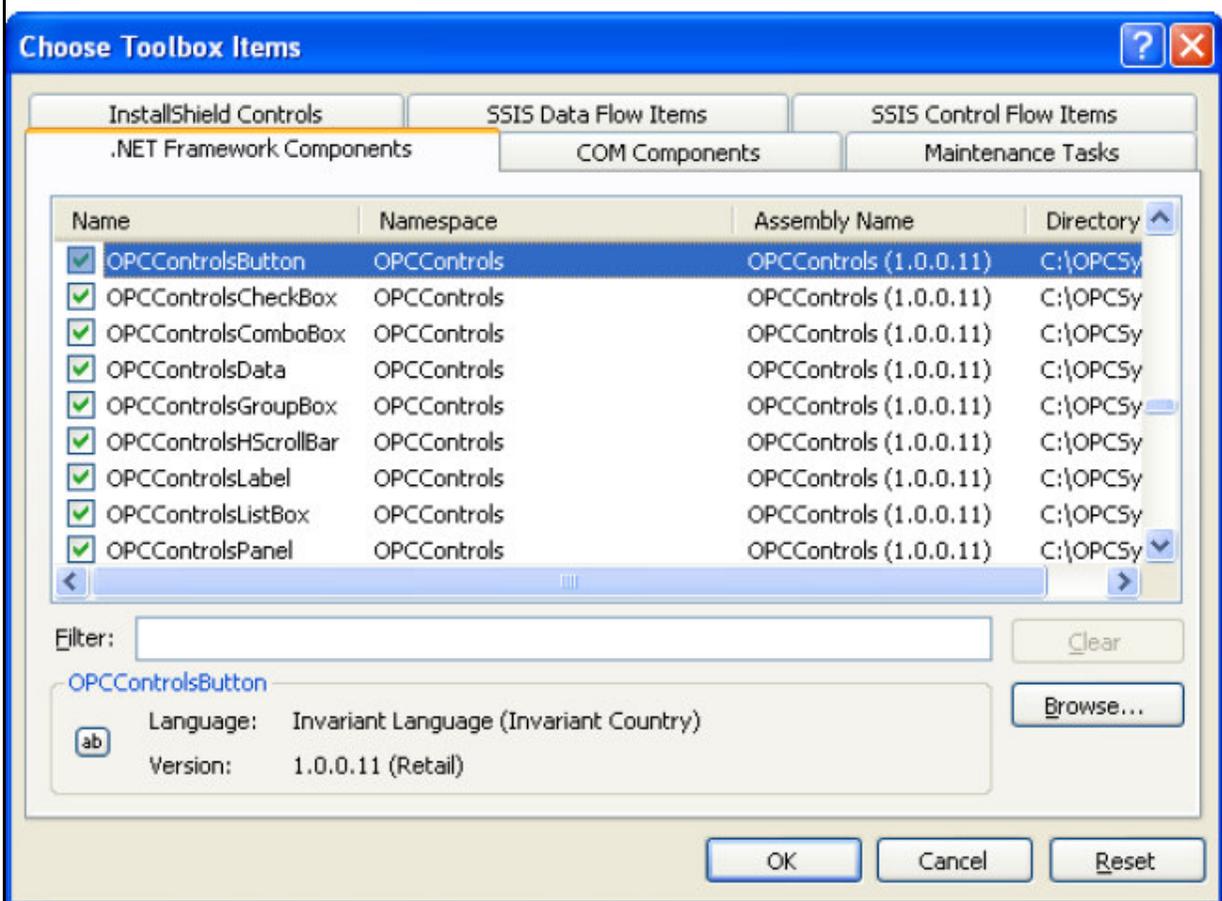
2 選擇 Windows Application 作為專案類型。



3 如果 OPCTrendControl 尚未存在於工具箱中，則請在工具箱中按下滑鼠右鍵並選取 Choose Items...。



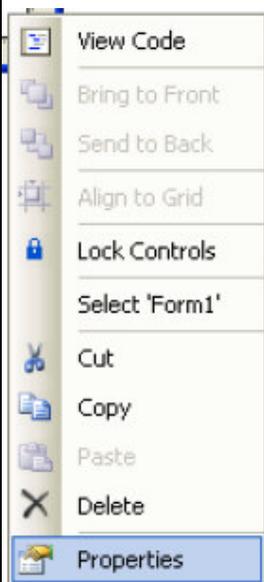
從.NET Framework Components 中勾選所有的 OPC Controls components 並按下 OK 按鈕。



4 新增 OPCControlsLabel component 到 Form 中。

A OPCControlsLabel

在 OPCControlsLabel 上按下滑鼠右鍵並選取 Properties。



點選 TextOPCSYSTEMS_Tag 屬性，並按下  按鈕來設定 OPC Systems.NET Tag 為 Ramp.Value。

TextOPCSYSTEMS_Tag 

注意：所有 Tag 名稱是有大小寫之分的。Ramp.Value 與 ramp.value 是不同的。

5 新增 OPCControlsButton 到 Form 中。

ab OPCControlsButton

設定 TextOPCSYSTEMS_Tag 為 Pump.Value。如果 Pump 這一個 Tag 不存在，則請在 Configure->Tags 中建立一個名稱為 Pump 的 Boolean Tag。並設定下圖所示之 Format 欄位。

TextOPCSYSTEMS_Tag	Pump.Value
TextOPCSYSTEMSBadQualityText	?????
TextOPCSYSTEMSFormatBooleanFalse	Pump is Off
TextOPCSYSTEMSFormatBooleanTrue	Pump is On

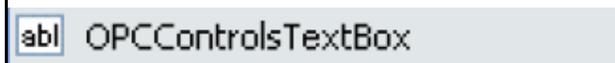
設定 BackColorOPCSYSTEMS_Tag 為 Pump.Value。

BackColorOPCSystems_Tag	Pump.Value
BackColorOPCSystemsBadQualityColor	 Yellow
BackColorOPCSystemsFalse	 Red
BackColorOPCSystemsTrue	 Lime

設定 SetValueOPCSystems_Tag 為 Pump.Value。且設定 SetValueOPCSystems 為 True。

SetValueOPCSystems	True
SetValueOPCSystems_Tag	Pump.Value
SetValueOPCSystemsDataType	LogicalDiscrete
SetValueOPCSystemsDiscreteValue	ToggleValue

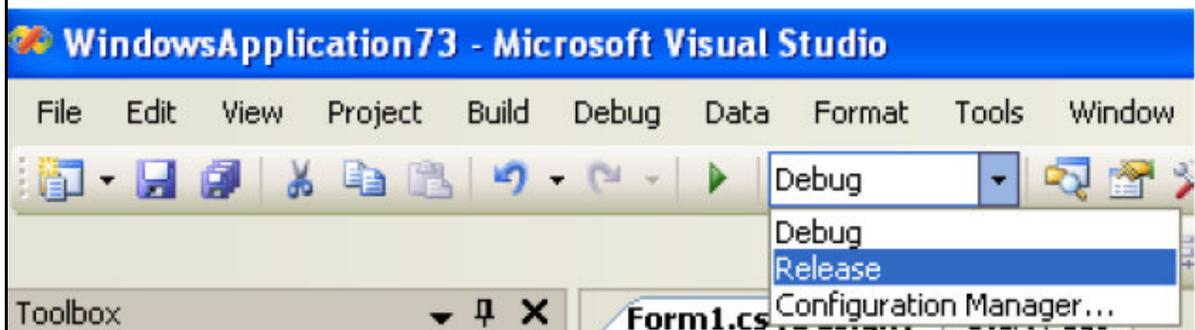
6 新增 OPCControlsTextBox 到 Form 中。



設定 TextOPCSystems_Tag 為 Pump.Value。Format 屬性為 Off 和 On。

TextOPCSystems_Tag	Pump.Value
TextOPCSystemsBadQualityText	?????
TextOPCSystemsDisableUpdateWithFocus	True
TextOPCSystemsFormatBooleanFalse	Off
TextOPCSystemsFormatBooleanTrue	On
TextOPCSystemsFormatFloat	0.000
TextOPCSystemsFormatInteger	0
TextOPCSystemsSendValueOnEnter	True

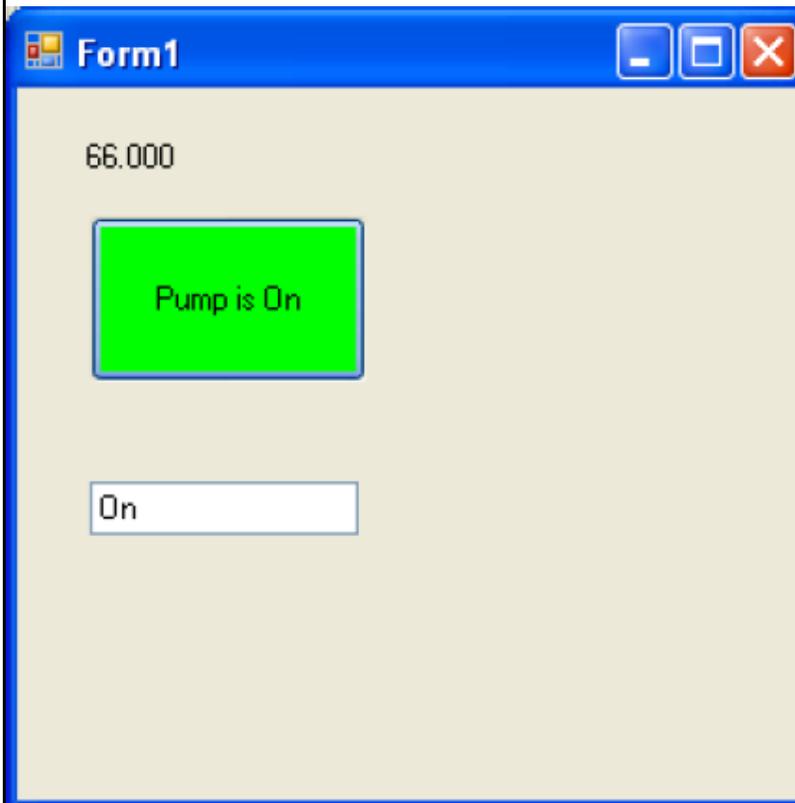
7 設定 Visual Studio 工具列中的 compile mode 為 Release。



8 從 Visual Studio 的 Build 功能表中選擇 Build 此應用程式。



9 使用檔案總管來瀏覽放置在 bin\Release 目錄中的執行檔並執行。



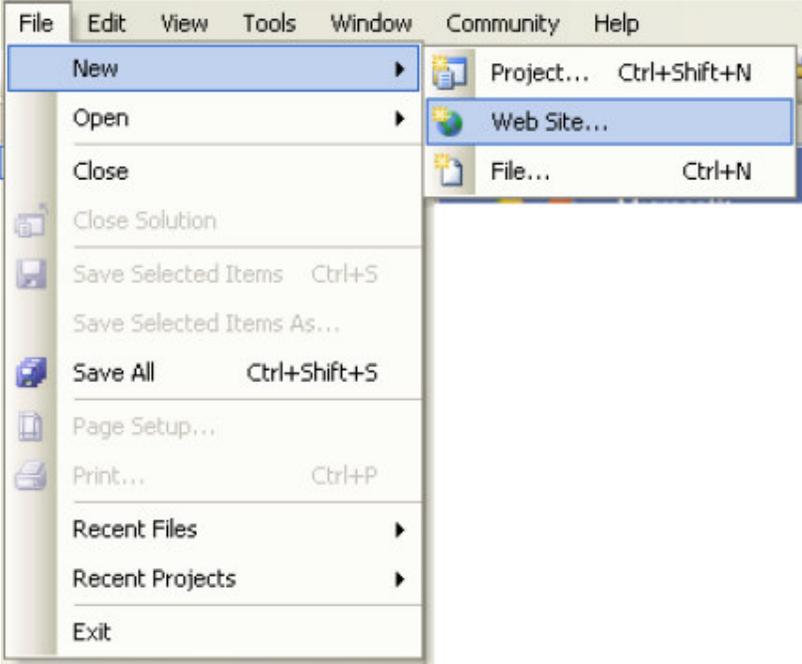
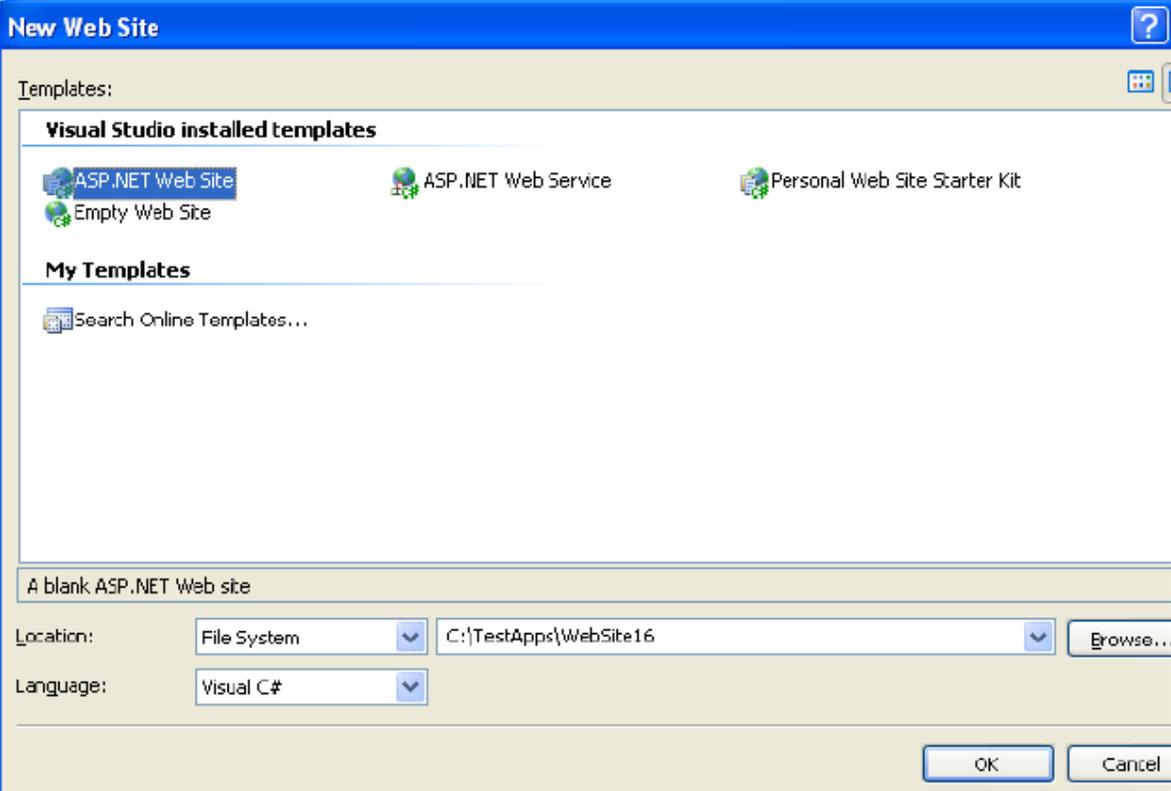
10 欲發佈此應用程式到遠端電腦中，則請確定在步驟 4、5 和 6 的 Tag 選擇中，是選擇網路上的電腦或是 IP 位址。然後複製放置在 bin\Release 目錄中的所有檔案至目的端電腦。

11 每一個控制項都擁有許多的屬性可供設定。在許多控制項中都有 ImageIndex 屬性可以被用來依據資料的值來顯示不同的圖片。

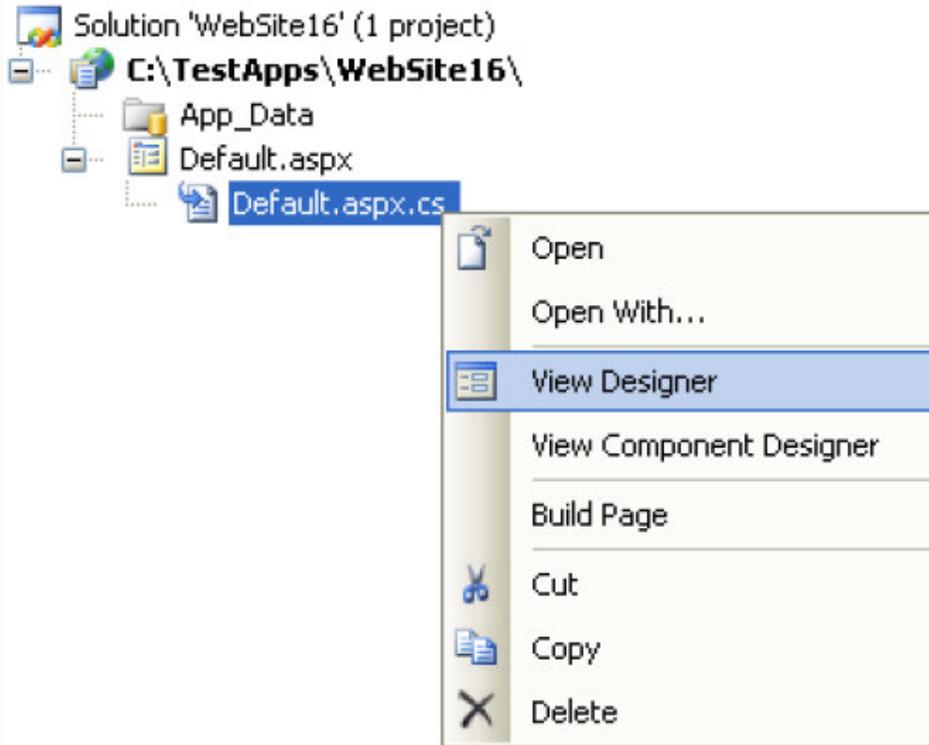
OPC Controls.NET Data component 可以被使用來利用程式碼，以相當簡單的方式來讀寫資料。請參考 VB.NET 範例程式中的 FormReadValues 和 FormWriteValues 二部份。

新增 HMI Controls 到 ASP.NET Web 應用程式

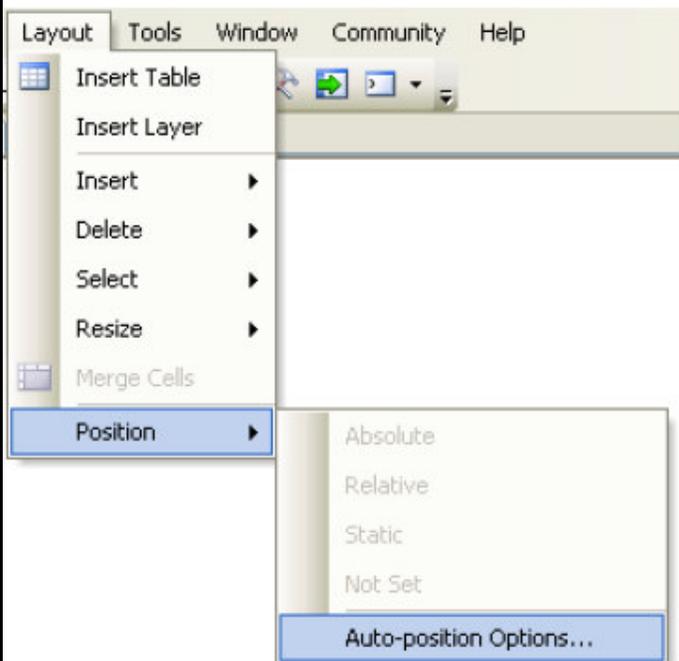
以下的步驟可以被用來新增視覺化和操作控制項到一個 C#、J#或 Visual Basic.NET Web 應用程式中。這些控制項的所有屬性都可以利用程式來讀取和修改。此例子展示不撰寫程式碼的方式來使用這些控制項。

Step	Task
1	<p>啓動 Visual Studio 2005 或 Visual Studio 2008 ，並選擇 File->New->Web Site 來建立新的 C# 、 J#或 VB ASP.NET Web 應用程式專案。</p> 
2	<p>選擇 ASP.NET Web Site 專案類型，並指定左下方所欲使用的程式語言。</p> 
3	<p>展開方案管理員中的 Default.aspx 網頁，並選取 View Designer 。</p>

注：如果你沒有看到方案管理員，則請選擇功能表 View->Solution Explorer。

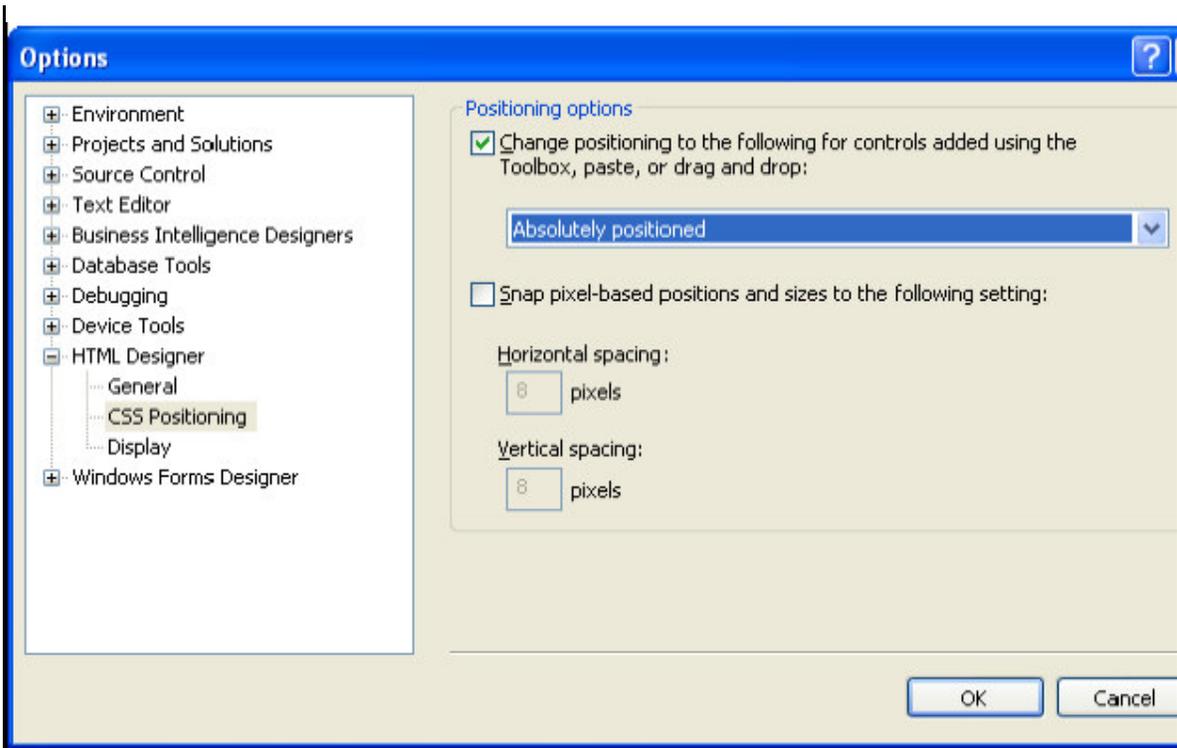


4 從 Visual Studio 功能表中選擇 Layout->Position->Auto-position 選項。

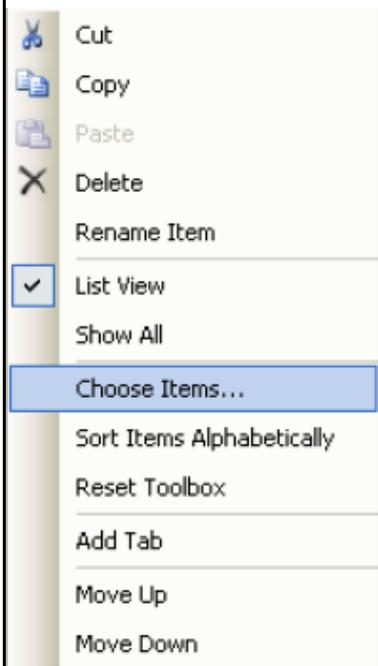


5 在 HTML Designer 中設定 CSS Positioning 為 Absolutely positioned。

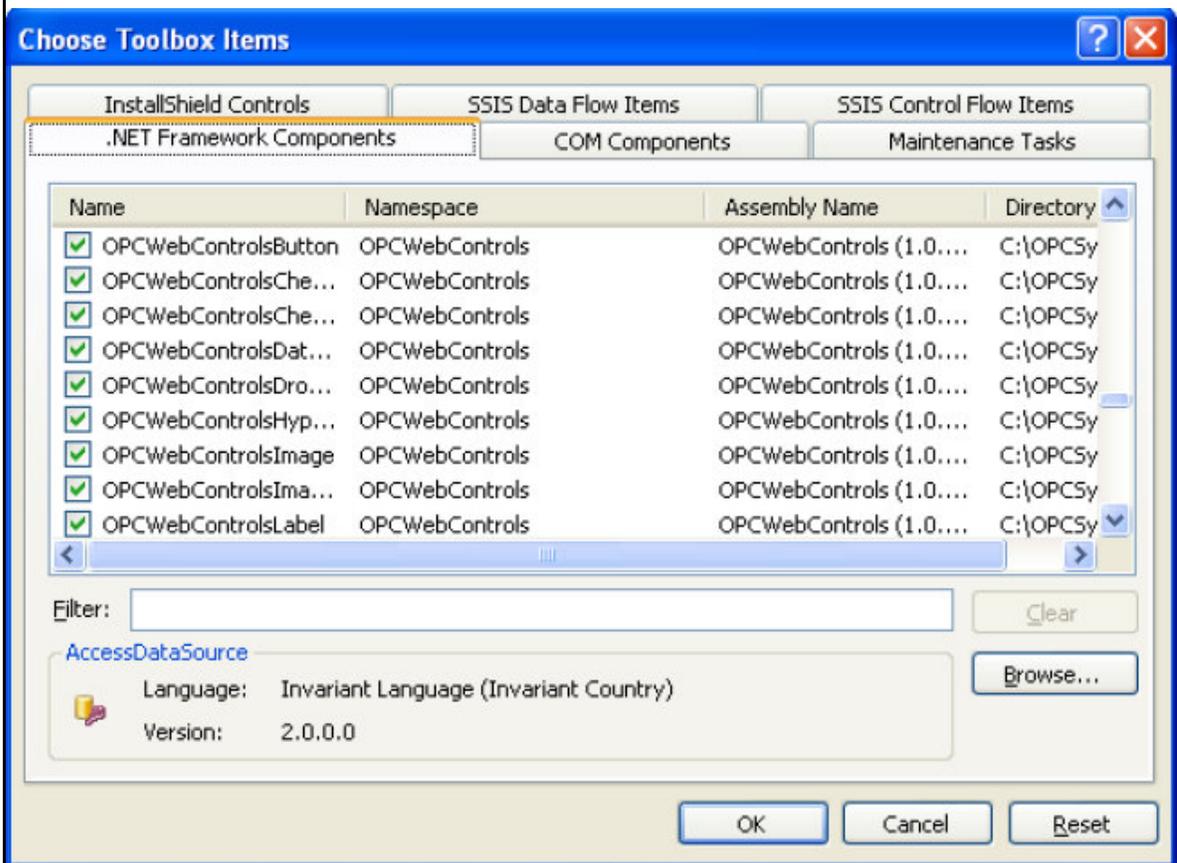
注：我們可以使用任何想要使用的定位型態。



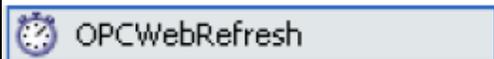
- 6 如果 OPCWebControls 和 OPCWebRefresh 尚未出現在工具箱中，則請在工具箱中按下滑鼠右鍵並選擇 Choose Items 來選取。



從 .NET Framework Components 中勾選 OPCWebControls components 和 OPCWebRefresh 並按下 OK 按鈕。



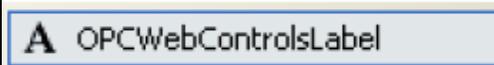
7 新增一個 OPCWebRefresh 控制項至 WebForm 中。



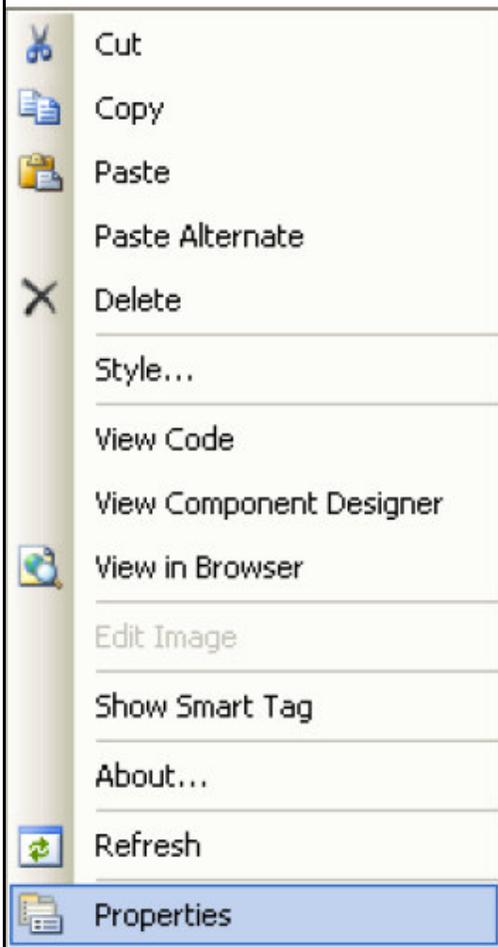
```
[OPCWebRefresh "OPCWebRefresh1"]
```

為確定專案的 web.config 檔已適當地修改為能夠執行 AJAX runtime，新增一個 OPCWebControlsLabel 到 WebForm 中，然後再刪除之。

8 新增 OPCWebControlsLabel 控制項至 WebForm 中。



在 OPCWebControlsLabel 控制項上按下滑鼠右鍵並選擇 Properties。



點選 TextOPCSystems_Tag 屬性，並按下  按鈕來設定 OPC Systems.NET Tag 為 Ramp.Value。

TextOPCSystems_Tag

注意：所有 Tag 名稱是有大小寫之分的。Ramp.Value 與 ramp.value 是不同的。

9 新增 OPCWebControlsButton 控制項至 WebForm 中。

 OPCWebControlsButton

設定 TextOPCSystems_Tag 為 Pump.Value。如果 Pump 這一個 Tag 不存在，則請在 Configure->Tags 中建立一個名稱為 Pump 的 Boolean Tag。並設定下圖所示之 Format 欄位。

TextOPCSys_tems_Tag	Pump.Value
TextOPCSys_temsBadQualityText	?????
TextOPCSys_temsFormatBooleanFalse	Pump is Off
TextOPCSys_temsFormatBooleanTrue	Pump is On

設定 BackColorOPCSys_tems_Tag 為 Pump.Value。

BackColorOPCSys_tems_Tag	Pump.Value
BackColorOPCSys_temsBadQualityColor	 Yellow
BackColorOPCSys_temsFalse	 Red
BackColorOPCSys_temsTrue	 Lime

設定 SetValueOPCSys_tems_Tag 為 Pump.Value。

SetValueOPCSys_tems_Tag	Pump.Value	
SetValueOPCSys_temsDataType	LogicalDiscrete	
SetValueOPCSys_temsDiscreteValue	ToggleValue	

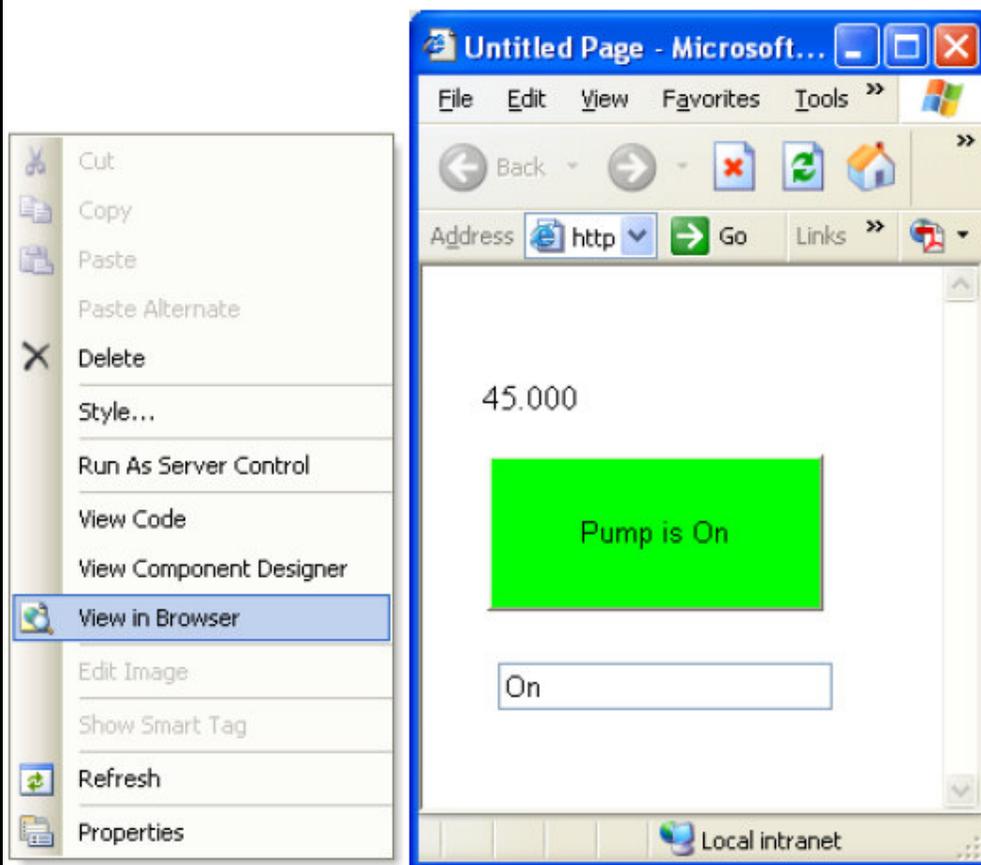
10 新增 OPCWebControlsTextBox 到 Form 中。



設定 TextOPCSys_tems_Tag 為 Pump.Value。Format 屬性為 Off 和 On。

TextOPCSys_tems_Tag	Pump.Value	
TextOPCSys_temsBadQualityText	?????	
TextOPCSys_temsDisableUpdateWithFocus	True	
TextOPCSys_temsFormatBooleanFalse	Off	
TextOPCSys_temsFormatBooleanTrue	On	
TextOPCSys_temsFormatFloat	0.000	
TextOPCSys_temsFormatInteger	0	
TextOPCSys_temsSendValueOnEnter	True	

11 在 WebForm 上按滑鼠右鍵並選擇 View in Browser。

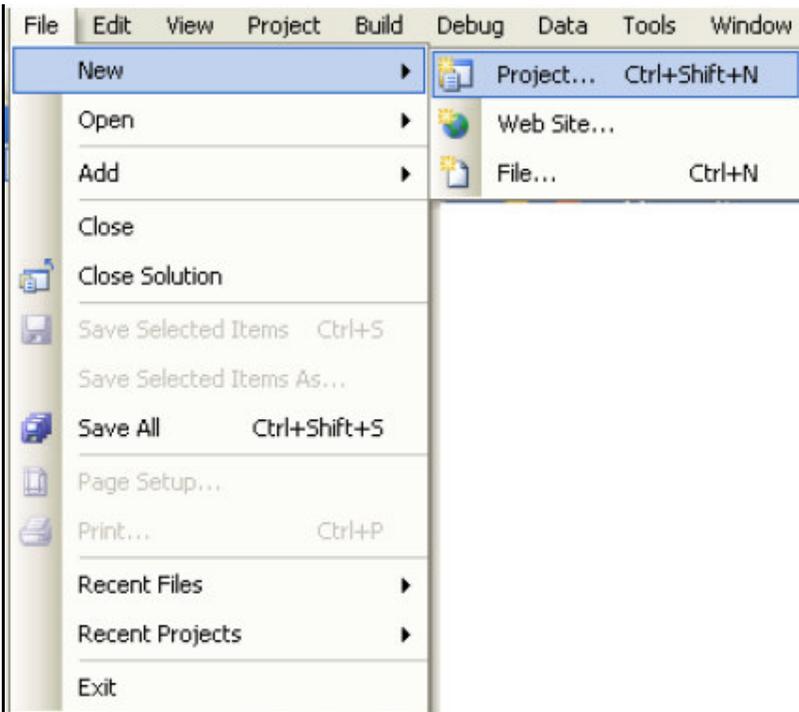


- 12 設定此應用程式為 Release 模式，修改 web.config 檔以確認 Debug 參數是設為 False，並且 Build 此應用程式。通常 Web 應用程式檔案都會被放在 Inetpub\wwwroot 目錄中。
- 參考 OPC Systems.NET help 檔中的 OPC Web Controls.NET 部份來了解特殊的考慮和開發步驟。

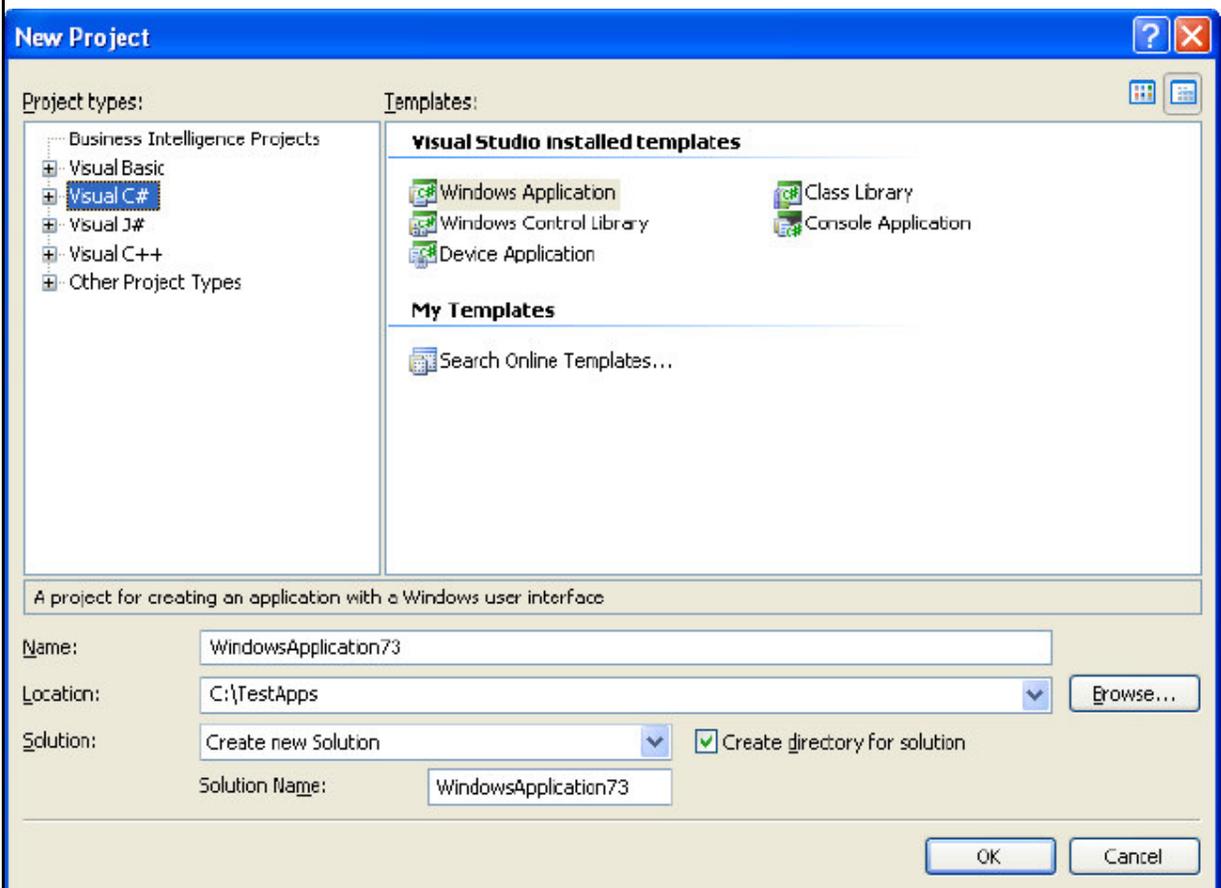
新增 Data Control 到 Visual Studio 應用程式

以下的步驟可以被用來新增即時資料讀寫功能到一個 C#、C++或 Visual Basic.NET 應用程式中。OPC Controls.NET Data component 也一樣可以被使用在 ASP.NET Web 應用程式中。請參考 VB.NET 範例程式中的 Read Values 和 Write Values 二 Form 來知道如何使用 OPC Controls.NET Data component。OPC Controls.NET Data component 的所有屬性都可以利用程式來讀取和修改。

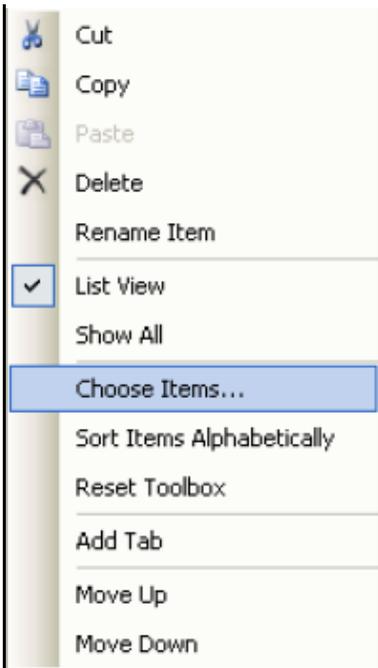
Step	Task
1	啟動 Visual Studio 2005 或 Visual Studio 2008，並選擇 File->New->Project 來建立新的 C#、C++或 Visual Basic.NET 應用程式專案。



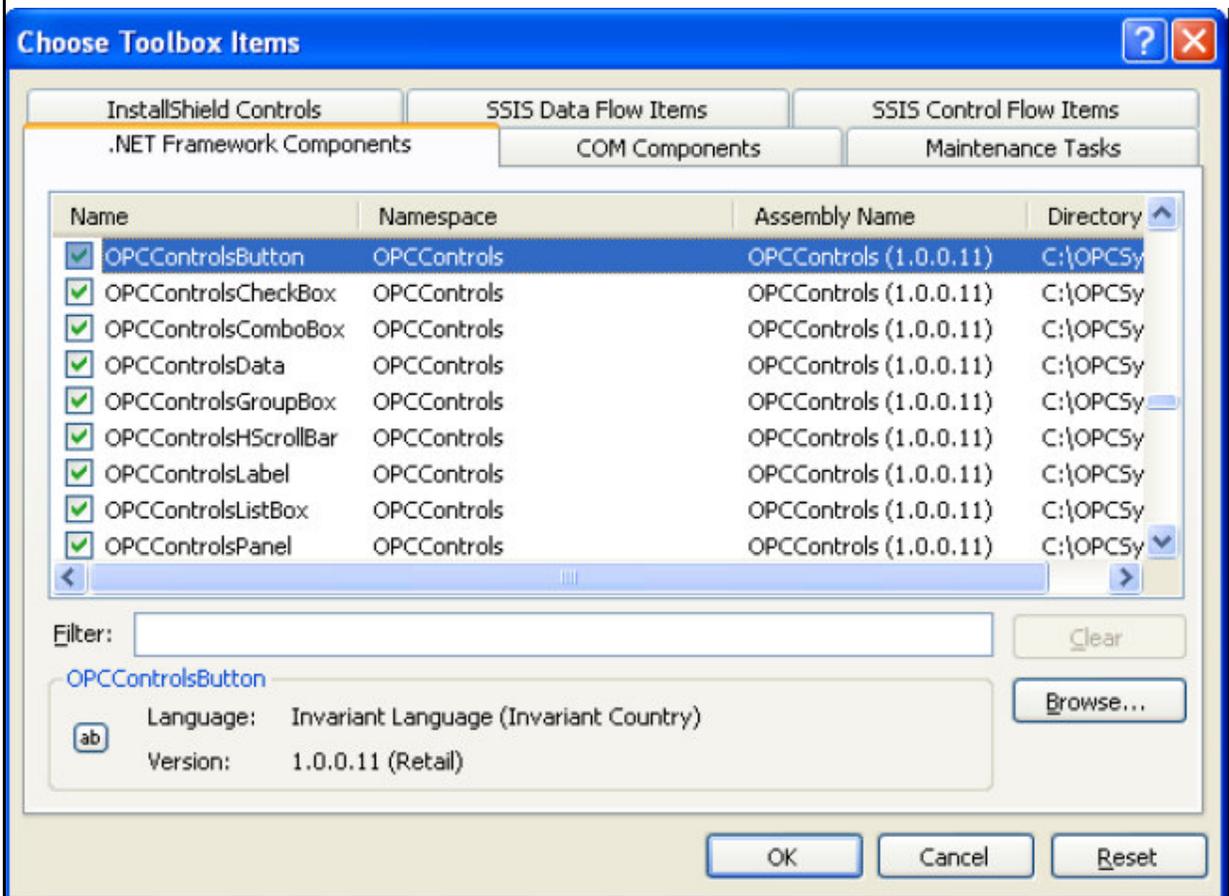
2 選擇 Windows Application 作為專案類型。



3 如果 OPCControls Data component 尚未存在於工具箱中，則請在工具箱中按下滑鼠右鍵並選取 Choose Items...。



從 .NET Framework Components 中勾選 OPCControlsData component 並按下 OK 按鈕。



4 新增 OPCControlsData component 到 Form 中。

5 新增一個 Button 控制項到 Form 中。

ab Button

變更此 Button 的名稱為 buttonAddTags。

(Name) **buttonAddTags**

變更此 Button 的 Text 為 Add Tags。

Text **Add Tags**

在 Button 控制項上連按滑鼠左鍵二次以進入程式編輯畫面，並鍵入以下程式碼。

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);
}
```

6

Form1.vb Form1.vb [Design] Start Page

(General)

(General)

Form1

- (Form1 Events)
- ButtonAddTags
- LabelPump
- LabelRamp
- OpcControlsData1

ValuesChangedAll

(Declarations)

- Disposed
- ValuesChanged
- ValuesChangedAll
- ValuesChangedWithTimeStamp

```

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

```

7 展開 ChartRates 屬性並設定 TimeFrame 為 60 秒。

ChartRates	60 Second Chart with 1 Second Sample Ra
SampleRate	1
TimeFrame	60
UpdateRate	1

8 選擇 Pens 屬性並點按右方  按鈕。

Pens ClassPen[] Array 

9 選擇 Local OPC Systems Service 來列出可用的 Tags。

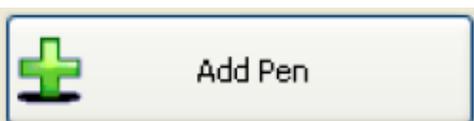


注意：如果你看不到想要用的 Tag 在清單中，則請回到 Configure->Tags，在欲使用的 Tag 參數設定中並勾選 Trend Point。

10 展開 Ramp Tag 並選取 Value。



按下 Add Pen 按鈕。



11 Ramp.Value 會出現在左下角的 pens 清單中。我們可以選擇 pen 來變更位於右方的屬性。當 Trend Window 的 YAxis.ScaleMode 屬性被設定成 PercentOfPenRanges 時，YAxisRangeHigh 和 YAxisRangeLow 屬性是非常重要的。

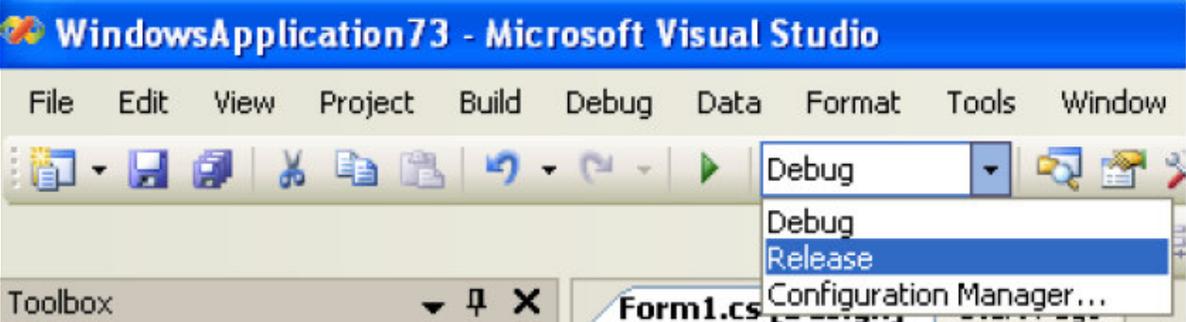
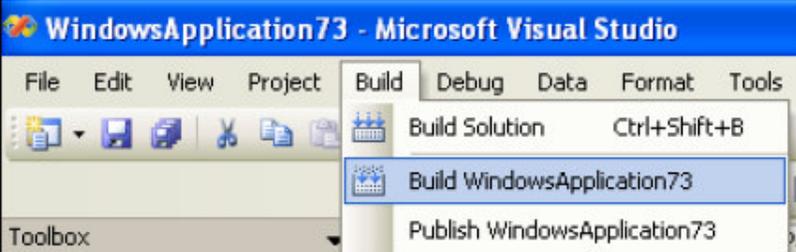
12 在 Pens 對話方盒中按下 OK 按鈕。

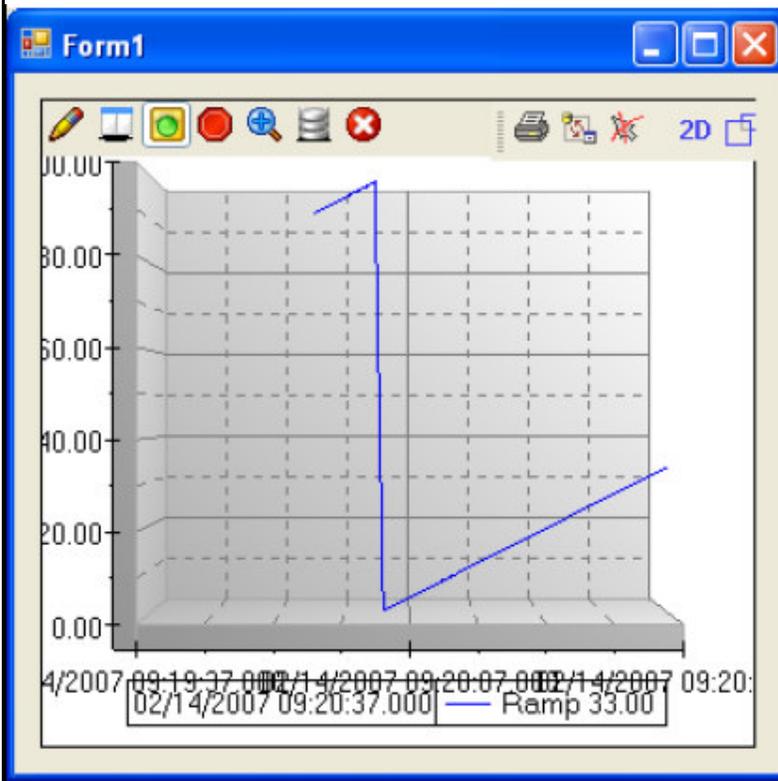
13 如果此電腦上的顯示卡支援 OpenGL rednering，則可設定 RenderMode3D 屬性為 True。



如果不確定的話，則保留此屬性為 False。

14 展開 Views 屬性並設定 Lighting 和 PerspectiveView 的為想要使用的值。

	<table border="1"> <tr> <td>Views</td> <td>Perspective With Lighting MetallicLustre</td> </tr> <tr> <td>Lighting</td> <td>MetallicLustre</td> </tr> <tr> <td>PerspectiveDepth</td> <td>20</td> </tr> <tr> <td>PerspectiveView</td> <td>Perspective</td> </tr> <tr> <td>ViewAxisDepthNumbers</td> <td>False</td> </tr> </table>	Views	Perspective With Lighting MetallicLustre	Lighting	MetallicLustre	PerspectiveDepth	20	PerspectiveView	Perspective	ViewAxisDepthNumbers	False
Views	Perspective With Lighting MetallicLustre										
Lighting	MetallicLustre										
PerspectiveDepth	20										
PerspectiveView	Perspective										
ViewAxisDepthNumbers	False										
15	<p>如果我們想要讓操作者在執行期間對 Trend Window 的設定進行保留，請設定 ConfigurationFile 屬性到一個有效的檔案路徑。</p>  <p>注意：如果你設定了此一屬性到某一個檔案，則請確實將該檔案放置於你指定的目錄中。如果我們希望在每次應用程式重啓動時都維持最初的原始設定，則讓此屬性保留為空白。</p>										
16	<p>設定 Visual Studio 工具列中的 compile mode 為 Release。</p> 										
17	<p>從 Visual Studio 的 Build 功能表中選擇 Build 此應用程式。</p> 										
18	<p>使用檔案總管來瀏覽放置在 bin\Release 目錄中的執行檔並執行。</p>										

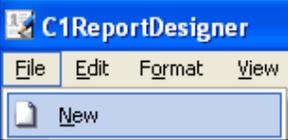
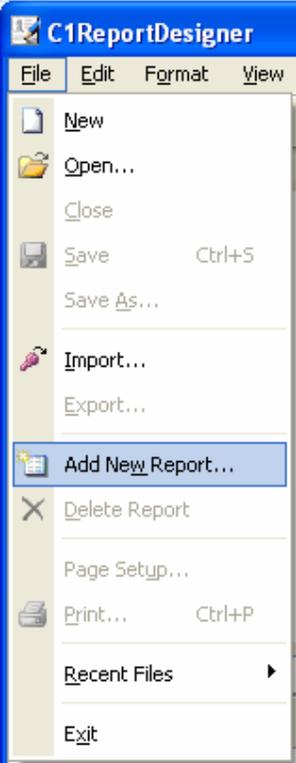


- 19 欲發佈此應用程式到遠端電腦中，則請確定在步驟 9 的 pen 選擇中，是選擇網路上的電腦或是 IP 位址。然後複製放置在 bin\Release 目錄中的所有檔案至目的端電腦。

第七章 Reports

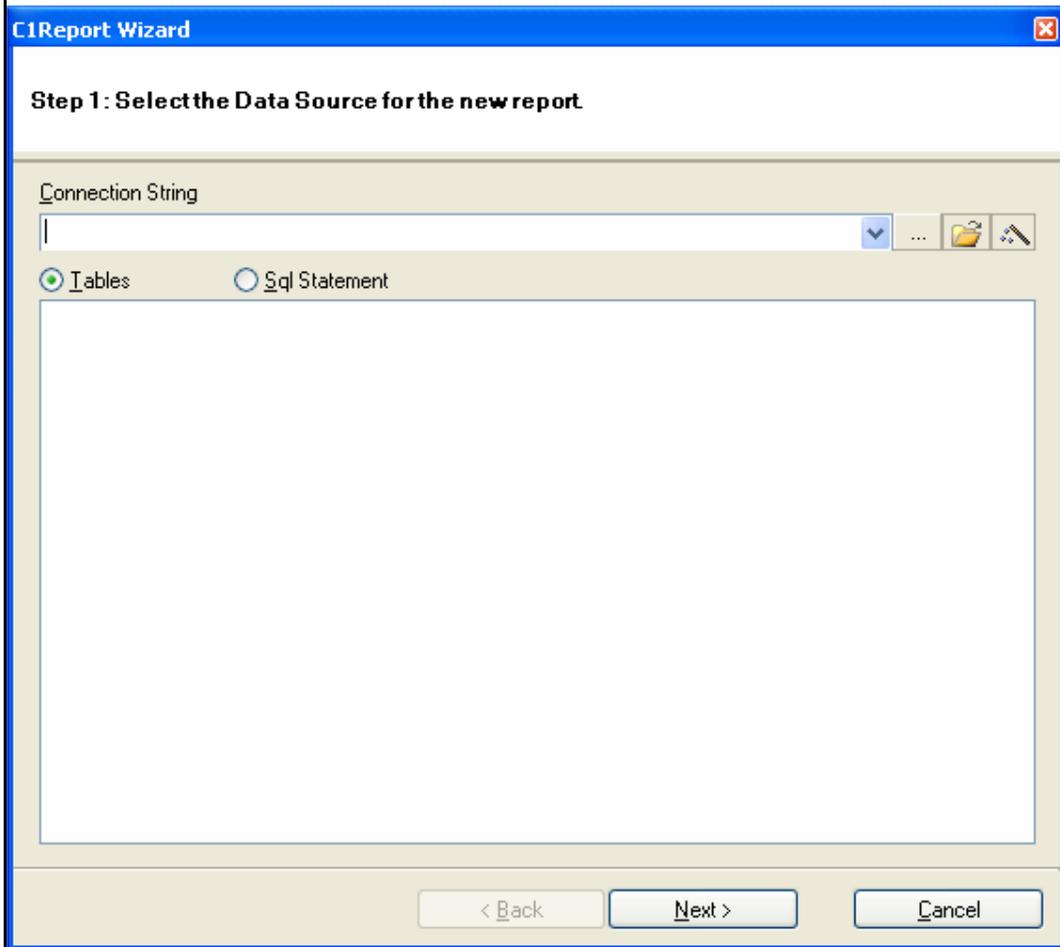
設定 Reports

Reports 可使用 Configure->Reports 來自動地產生 PDF、HTML、Word、Excel、Text 和印表機報表，或是以 Report Viewer 手動地產生。

Step	Task
1	必須先完成 Configure Data Logging 範例並讓 OPC Systems Service 執行在 Runtime 模式以記錄資料。
2	執行「開始」→「程式集」→OPCSystems.NET → Report Designer → Report Designer 以啟動 Report Designer 應用程式。
3	選擇 File → New 以清除目前 Report Designer 工作區內容。 
4	選擇 File → Add New Report。 

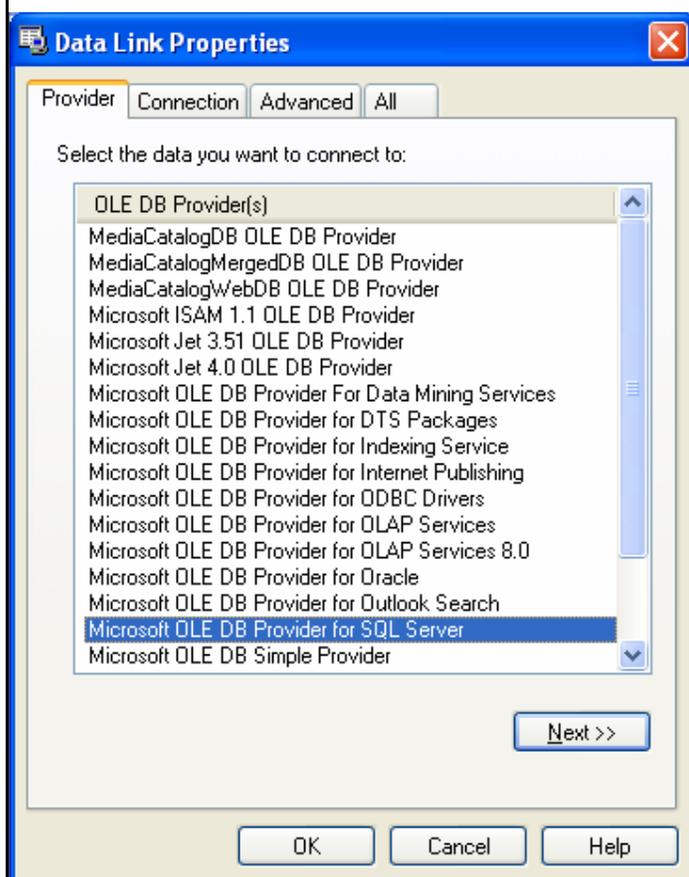
5

使用  按鈕來設定資料來源。



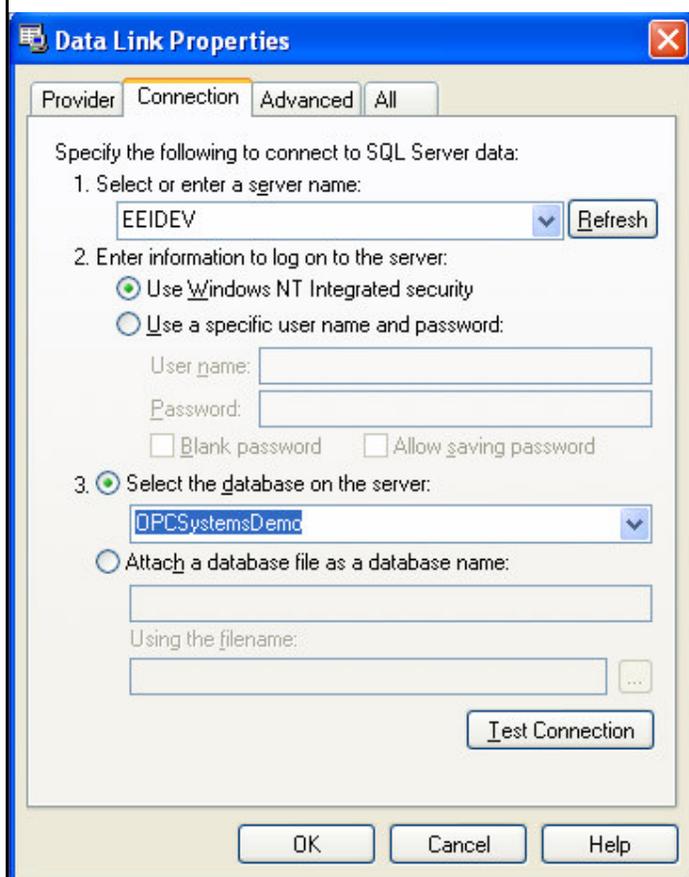
6

為你的資料庫指定適當的 Provider



7 指定 Server 和權限設定。

選取 OPCSystemsDemo 資料庫。

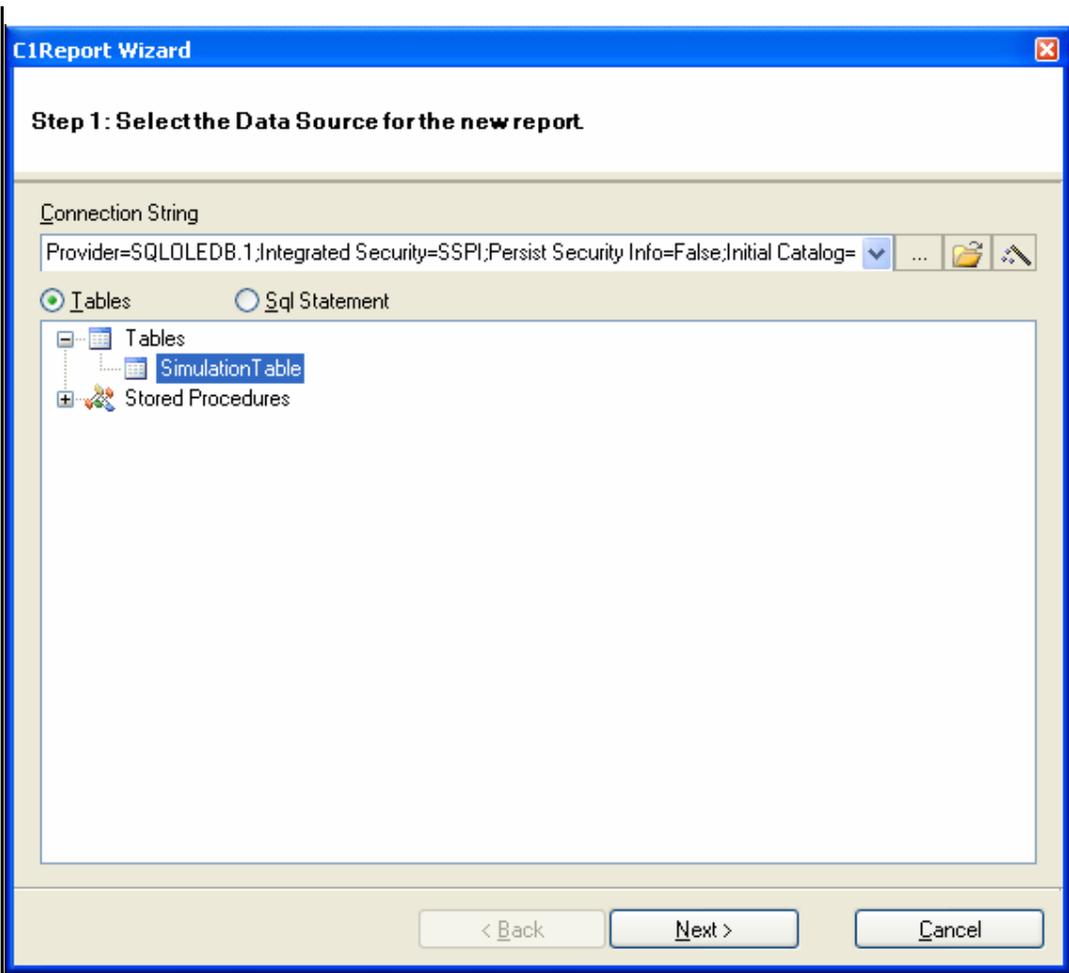


8

使用  按鈕來確認資料庫連接，如無問題則按下 OK 按鈕

9

選取 SimulationTable 並按下 Next。



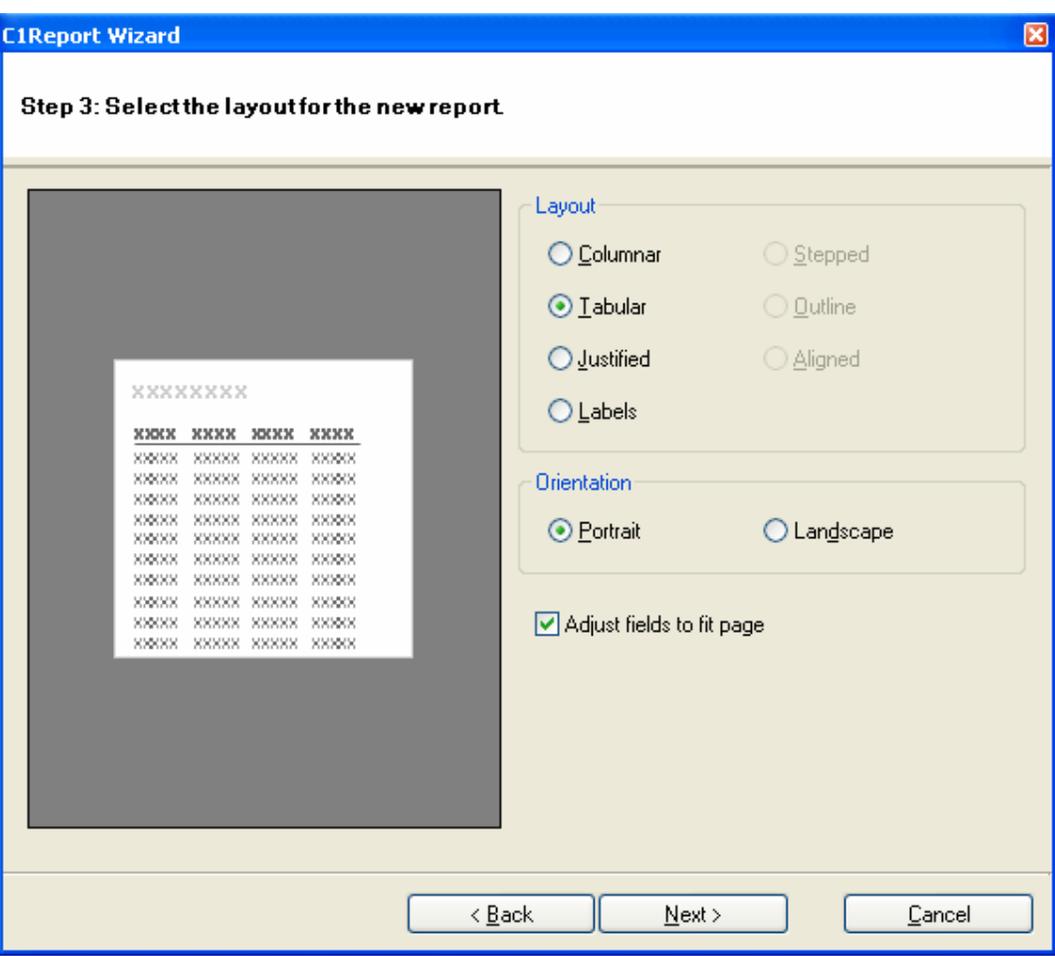
10 移動所有欲使用的欄位到 Detail 區。

The screenshot shows a software window titled "C1Report Wizard" with a close button in the top right corner. The main heading is "Step 2: Select the Fields that will be included in the new report." The interface is divided into three main sections: "Available" (a large empty list box on the left), "Groups" (a smaller empty list box on the top right), and "Detail" (a list box on the bottom right containing the text: "DateAndTime", "ms", "Ramp_Value", "Random_Value", and "Sine_Value"). Between the "Available" and "Detail" sections are two buttons: a blue button with ">>" and a grey button with "<<". At the bottom of the window are three buttons: "< Back", "Next >", and "Cancel".

11	點選預設的 Tabular 報表佈局，並按下 Next。
----	------------------------------

C1Report Wizard

Step 3: Select the layout for the new report



Layout

Columnar Stepped

Tabular Outline

Justified Aligned

Labels

Orientation

Portrait Landscape

Adjust fields to fit page

< Back Next > Cancel

12 選擇樣版型態。報表顏色、字型 and 圖形都可以再改變。

C1Report Wizard

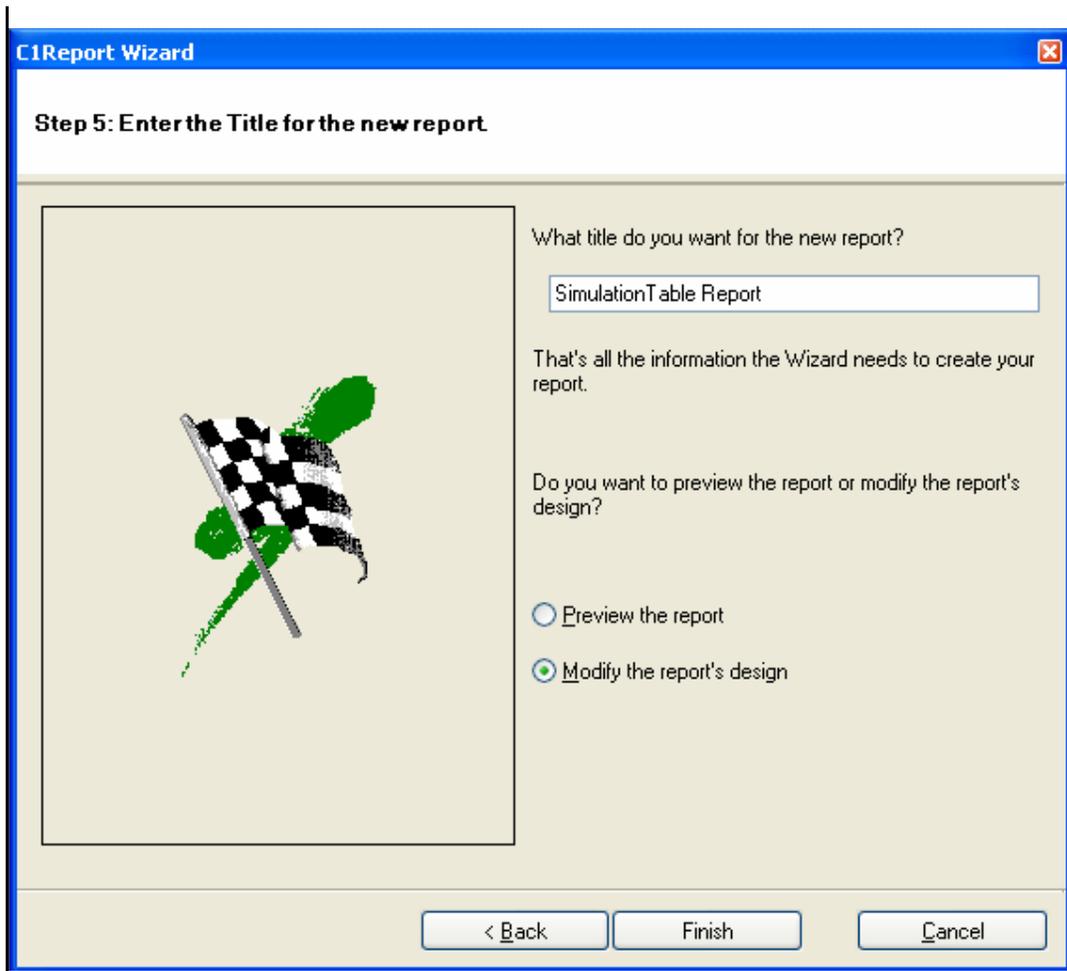
Step 4: Select the style for the new report

The preview area shows a report layout with a title bar, a label, and a detail section. The style list includes:

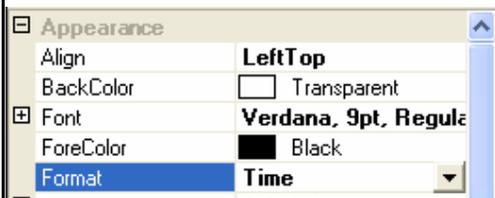
- [Bold]
- [Casual]
- [Compact]
- [Corporate]
- [Formal]
- [Soft Gray]
- Verdana**
- WebReport

Buttons: < Back, Next >, Cancel

13 點選 Modify the reports' design，並按下 Finish 按鈕。



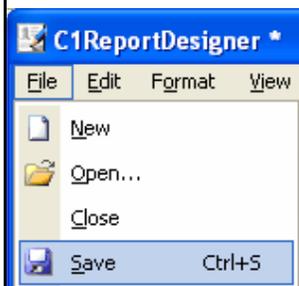
14 點選右方報表 Detail 內之 DateAndTime 欄位，並在左下角的屬性窗格設定 Format 屬性為 Time。



15 將右方報表細節內之 Sine_Value 欄位拉寬一些。

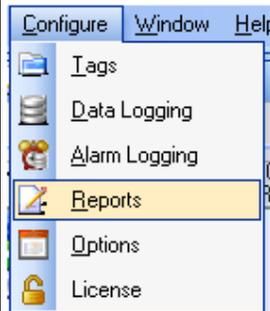
SimulationTable Report				
DateAndTime	ms	Ramp_Value	Random_Value	Sine_Value
DateAndTime	ms	Ramp_Value	Random_Value	Sine_Value
Now()				"Page " & [Page] & " of " & [Pages]

16 選擇 File->Save，以 ReportDemo 為檔名來儲存，此將建立一 XML 檔。

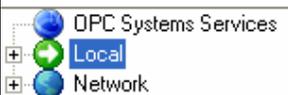


17 關閉 Report Designer，並執行 Configure OPC Systems 應用程式。

18 選擇 Configure->Reports。



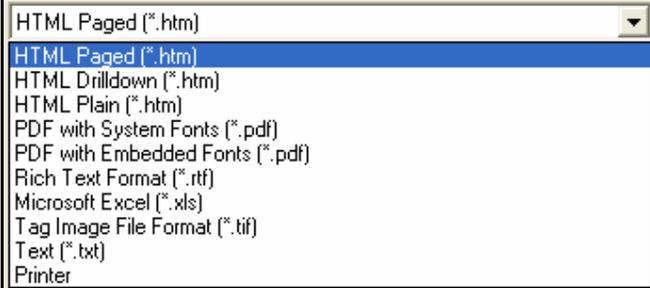
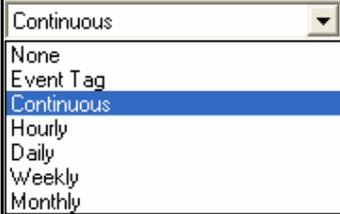
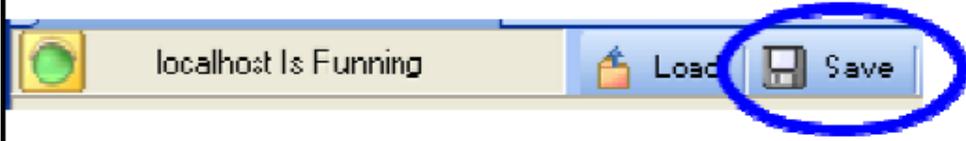
19 選取 Local OPC Systems Service。



20 在右上角 Report Name 欄位中鍵入 Simulation 為報表名稱。



21 按下 Browse 按鈕找到剛才以 Report Designer 建立的 XML 檔。

	
22	<p>從列示框中選出剛才建立的報表。</p> 
23	<p>選取想要的輸出型態。</p> 
24	<p>指定輸出目錄。</p> 
25	<p>設定 Execution Type 為 Continuous。</p> 
26	<p>設定 Execution Rate 為 60 秒。</p>
27	<p>指定 Filter 為 None。如有大量的資料，則可能有必要設定查詢條件來過濾資料。</p>
28	<p>選擇  按鈕以加入 Report。</p>
29	<p>按下上方工具列中的 Save 按鈕來儲存剛才之設定。</p> 
30	<p>以 DemoReport.Reports 為檔名儲存到 C:\OPCSystemsDemo 中。</p> <p>使用 Configure->Options 來設定成當 OPC Systemes Service 啟動時，自動載入此</p>

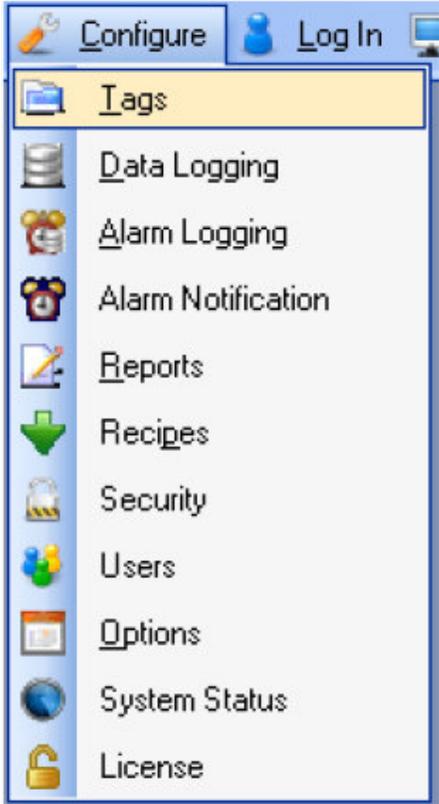
設定檔。

當你啓動 **Service** 在 **Runtime** 模式，報表將每分鐘產生一次。

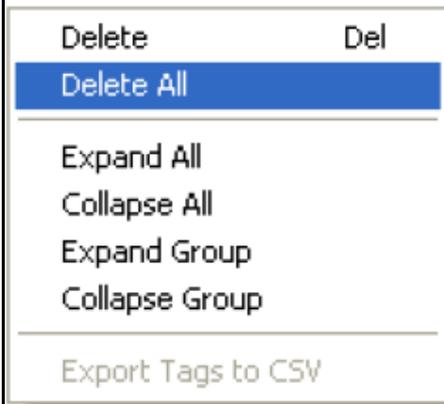
我們也可以指定輸出爲 PDF 檔格式，並設定以 E-mail 來發送報表的選項。

第八章 設定 Recipes

Recipes 是用來傳送值從資料庫到 OPC Systems.NET Tags。如果 Tags 的資料來源是設成 OPC Item，則此值就會被寫到 OPC Item 所定義的控制器位址。最常見的應用是以 PLC 上的一個變數來觸發 recipe 的執行。PLC 將等待確認碼或是錯誤碼被寫入 PLC，以指示該 recipe 執行是否完成。如果有錯誤，PLC 可以再次嘗試觸發傳送動作。

Step	Task
1	啓動 Configure OPC Systems 應用程式。
2	選擇 Configure->Tags 新增一些 Tags 來被使用在 Recipe 的執行。 
3	選擇 Local OPC Systems Service。 
4	如果你正在設定一已存在的 Tag 設定檔，則請進行下列動作來儲存它。

在 Tags 窗格中按下滑鼠右鍵並選擇 Delete All 來刪除所有的 Tag。



5 以指定的型態新增下列的 Tag：

Boolean 型態的 **Trigger**

Integer 型態的 **Value01**

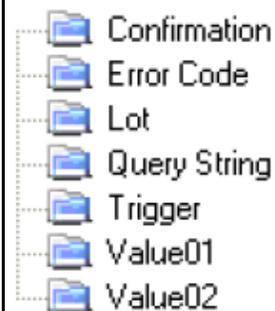
Integer 型態的 **Value02**

值為 "A" 的字串型態 **Lot**

Boolean 型態的 **Confirmation**

Integer 型態的 **Error Code**

資料來源為 Calculation 的字串型態 **Query String**，其內容為 "WHERE Lot=" & "" & [Lot.Value] & ""



6 按下上方工具列的 Save 按鈕，並使用 Recipe Tags 為檔案名稱。



7 使用 Microsoft SQL Server、Oracle、Access 或 MySQL 來建立下列的 Table。

建立一個名稱為 OPCSystemsRecipes 的 Database。

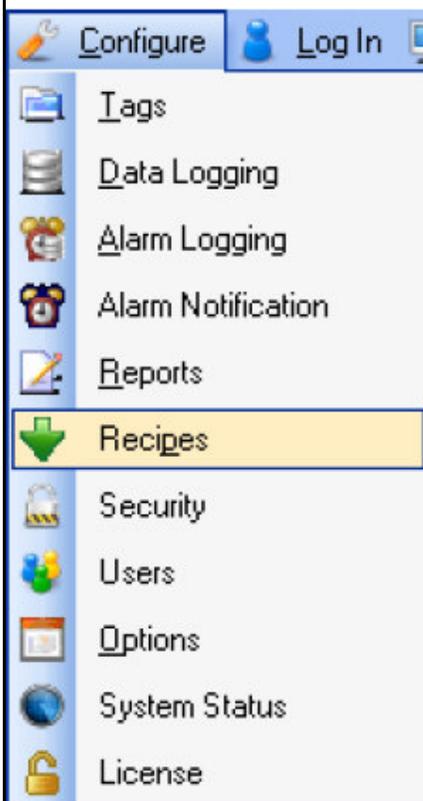
建立一個名稱爲 RecipeValues 且如下欄位結構的 Table。

Column Name	Data Type	Allow Nulls
Lot	char(10)	<input checked="" type="checkbox"/>
Value01	int	<input checked="" type="checkbox"/>
Value02	int	<input checked="" type="checkbox"/>

鍵入下列的值到 RecipeValues。

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

8 在 Configure OPC Systems 程式中選擇 Configure->Recipes。



9 選擇 Local OPC Systems Service。

NetworkNode Select localhost, Network Node, or IP Address

10 鍵入 Recipe01 為 Recipe 名稱。

Recipe Name

勾選 Recipe Active。

Recipe Active 

設定 Recipe Type 為 SingleRecord。

Recipe Type

設定 Set to Execute Recipe 為 Trigger.Value。

勾選啓用 Enable Confirmation Tag 且設為 Confirmation.Value。

勾選啓用 Enable Error Tag 且設為 Error Code.Value。

Recipe Name

Common
 Tags
 Database

Recipe Active 

Activate Recipe With Tag

Recipe Type

Execution Type

Tag To Execute Recipe

Enable Confirmation Tag 

Enable Error Tag, 0 = No Error, 1-5 See Help File 

Timeout Seconds

Float Deadband

Write All Values Without Feedback

11 選擇 Tags 頁籤。



我們可以指定 Database Table 中的欄位到 OPC Systems.NET 的 Tag，讓 recipe 執行時能夠寫入。如果 OPC Systems.NET Tag 的資料來源是定義成 OPC Item，它將自動地寫值到 OPC Item。

也可以選擇性地使用 DirectOPC 來直接連接 OPC Server 上的 OPC Items，而不是 OPC Systems.NET 的 Tag。

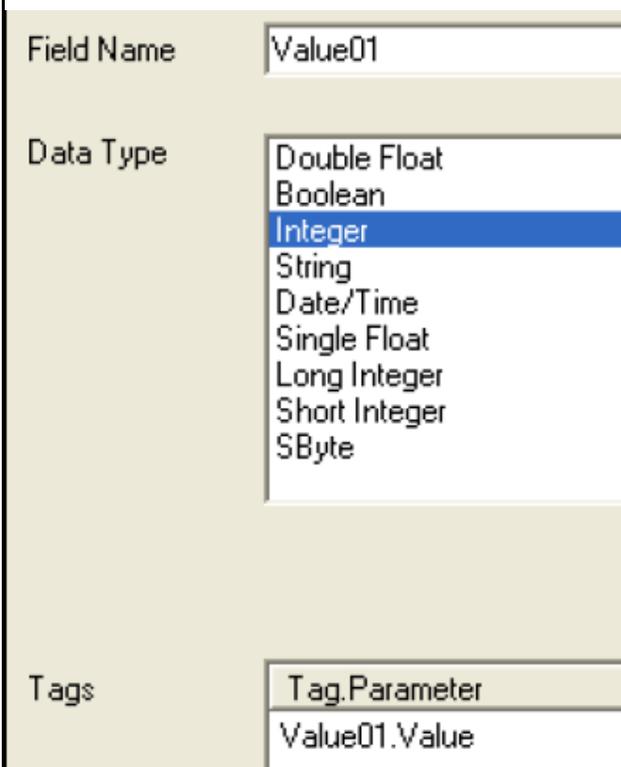
按下 Add 按鈕。



選擇 Tag 和 Parameter 為 Value01.Value。

改變 Field Name 為 Value01。

變更 Data Type 為 Integer。

A screenshot of a configuration dialog box with a light beige background. It contains three main sections: 1. "Field Name" with a text input field containing "Value01". 2. "Data Type" with a dropdown menu that is open, showing a list of data types: "Double Float", "Boolean", "Integer" (highlighted in blue), "String", "Date/Time", "Single Float", "Long Integer", "Short Integer", and "SByte". 3. "Tags" with a list box containing "Tag.Parameter" and "Value01.Value".

按下 OK 按鈕。

再按下 Add 按鈕。



選擇 Tag 和 Parameter 為 Value02.Value。

改變 Field Name 為 Value02。

變更 Data Type 為 Integer。

Field Name	Value02
Data Type	Double Float Boolean Integer String Date/Time Single Float Long Integer Short Integer SByte
Tags	Tag.Parameter Value02.Value

按下 OK 按鈕。

12 選擇 Database 頁籤。



定義適當的資料庫連接。

勾選啓用 Set Query String with Tag 並設為 Query String.Value。

Common  Tags  Database

Provider:

Server:

Set Server Name with Tag
 

Database:

Set Database Name with Tag
 

Table:

Set Table Name with Tag
 

Use WinNT Authentication

User Name:

Password:

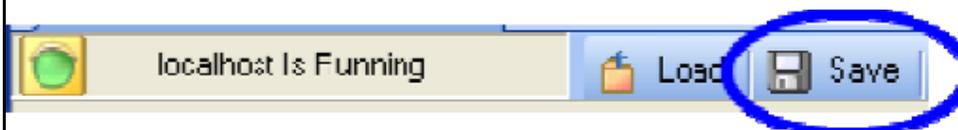
Query String: 

Set Query String with Tag 
 

13 按下左下角的 Add 按鈕來新增 Recipe。

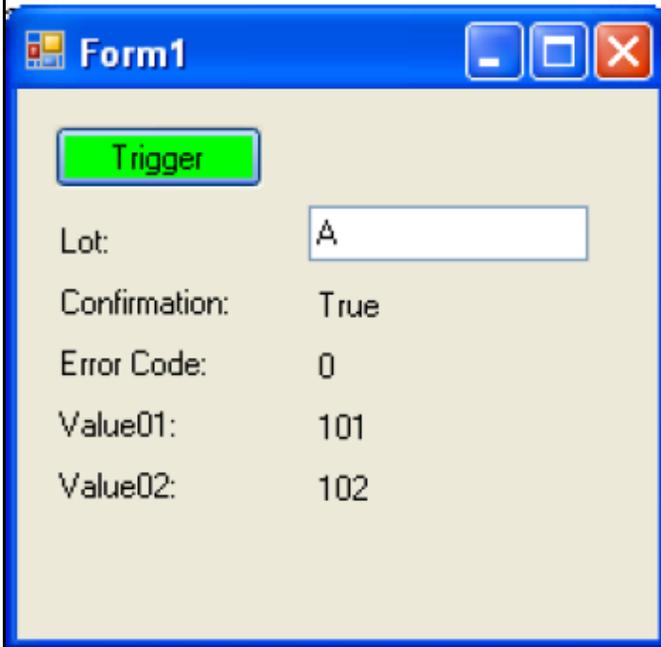


14 按下上方工具列中的 Save 按鈕。



以 RecipeExample.Recipes 為檔名，儲存在 C:\OPCSytemsDemo 資料夾中。

15 在此一時點



16 如果我們在 Configure->Options 中已啓用 Error Logging，則當我們執行 Recipe 時，將會有 Recipe 執行的 log 產生在 Error Log 同樣的目錄中。

Queued Recipe 型態的 recipe 在 recipe 執行成功時，將刪除 returned record。

MultipleRecord Recipe 型態的 recipe 僅有二欄位，一個是用來設定 Tag Name，另一個是寫入 Tag 的值。使用此種型態，我們可以指定什麼 Tag 要寫到資料庫。

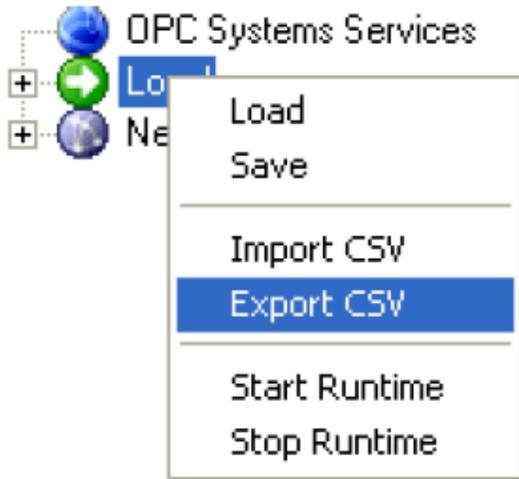
當 recipe 被執行，僅當資料庫中值與目前 Tag 的值不同才會進行寫入動作。浮點數值使用定義在 Common 頁籤中的 Float Deadband 屬性，來指定什麼是允許的範圍並決定值是相同或不同。

Timeout 屬性是指定 recipe 執行後，等待所有值從 Tags 回覆的時間量。如果 Tag 的資料來源是 OPC Item，則當一個或多個值無法從 OPC Server 讀回時，此 recipe 的執行將以 Timeout 爲錯誤碼而停止執行。

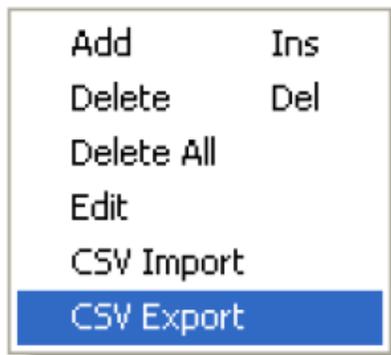
Write All Values Without Feedback 屬性一經設定，recipe 的執行將不會等待值的回覆，而直接將所有資料庫中的值寫入指定的 Tag 中。此種設定通常是使用在 Execution Type 設定成 Continuous，且爲快速執行爲目的時。

Recipe CSV 匯出和匯入

所有 Recipe 可以被匯出成爲 CSV 檔，包含所有 recipe 欄位名稱到個別的次檔案。其做法是在 Local service 上按滑鼠右鍵並選擇 Export CSV。



我們也可在 Tags 欄位清單上按滑鼠右鍵並選取 CSV Export 來僅匯出 recipe 群組中的個別欄位。

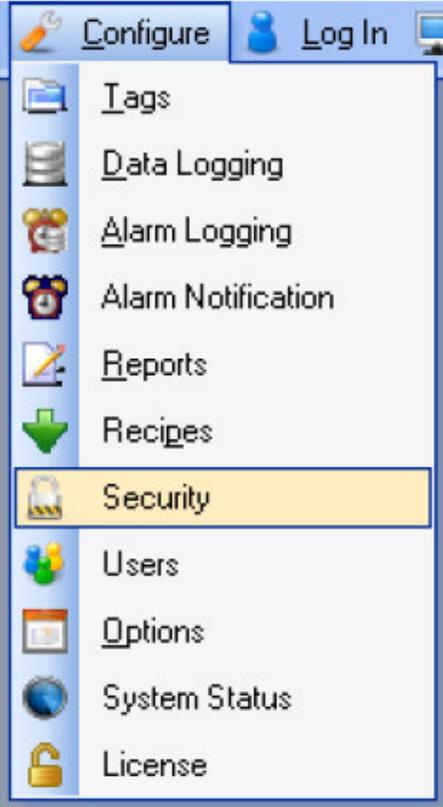


Recipe 程式開發界面

使用 OPC Systems component 在 Visual Studio 應用程式中，可程式化修改 Recipe 群組。參考 VB.NET example 範例程式中的 Configure Recipes Form 和 Configure CSV Form 來了解如何新增和修改 recipe 群組。請參照 OPC Systems Component help 檔來得知所有 method 的語法。

第九章 設定 Security

建立 Security 機制可以用來限制對於設定參數的變更，以及即時和歷史資料的存取。

Step	Task
1	啓動 Configure OPC Systems 應用程式。
2	選擇 Configure → Security。 
3	選擇 Local OPC Systems Service。 
4	Default Security Group 包含所有的 security 預設設定。當 Client 應用程式連接上 Service 但未登入時，則套用此設定。勾選 Enable All Features 將啓用所有功能的存取。我們可以新增多個 Security 群組，且讓它們各有自己的 security 設定。 新增 Security 群組之後，再以 Configure->Users 來指定 Security Users 到適當的 Security 群組。Default Security Group 是不能指定給 User 的。



Common 屬性是指整個 OPC Systems Service 所共有且可以被限制的功能。



Tags 屬性是指可以用來操作即時 Tag 資料庫各種方法。



Read Tags 屬性提供啓用或停用對 Tags 的即時資料存取。我們可先停用所有 Tags 的讀取，然後再開放某些 Tags 的讀取限制。或者相反地，先可先開放所有 Tags 的讀取，然後再停用某些 Tags 的讀取。此 Read Tags 屬性不僅僅可影響從像是 OPC Controls.NET 和 OPC Web Controls.NET 的 HMI 控制項的存取，也限制即時 Trending 和 Data Logging 的運作。



Write Tags 屬性是用來限制 Client 程式對 Tags 的寫入動作。



Trends 屬性是用來限制某些特定名稱的取用。



Trend RealTime 屬性是用來幫忙限制即時 Trending 的存取。



Trend History 屬性是用來限制 Client 程式對歷史資料的重播。



Data Log 屬性是用來限制對 data logging 設定參數的存取和修改。要限制資料的儲存，請使用 Read Tags 屬性。



Alarm 屬性是用來限制已定義的 Alarm Group Names 之取得。



Alarm Ack 屬性是用來限制去 Acknowledge 基於 Alarm Priority 和 Alarm Groups 所產生的 alarm 之能力。



Alarm RealTime 屬性是用來限制基於 Alarm Priority 和 Alarm Groups 所產生的現行 alarm 之存取能力。此屬性也限制了 Alarm Logging 功能。



Alarm History 屬性是用來限制從基於 Alarm Priority 和 Alarm Groups 所儲存的歷史 alarm 之存取能力。



Alarm Log 屬性是用來限制對 alarm logging 設定參數的存取和修改。要限制 alarm 的儲存，請使用 Alarm RealTime 屬性。



Alarm Notification 屬性是用來限制對 alarm notification 設定參數的存取和修改。



Reports 屬性是用來限制對 report 設定參數的存取和修改。



Recipes 屬性是用來限制對 recipe 設定參數的存取和修改。



Security 屬性是用來限制對 security 設定參數的存取和修改。



注意：在限制存取和修改 Default Group 中的 Security 之前，請先新增加一個 Security 群組，其包含有存取和修改所有 security 的設定，並指定 User 到這一個 Security 群組中。



Options 1、2 和 3 屬性是用來限制對 Configure->Options 設定參數的存取和修改。

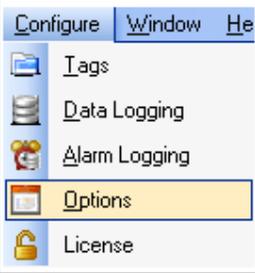
6	 <p>一旦定義了新的 Security 群組，記得為每一個 User 選擇並指定使用者名稱、密碼和 Security 群組。</p>
7	<p>在上方的工具列中按下 Save 按鈕，並指定一個名稱，來儲存 Security 和 User 設定。</p>  <p>在 Configure->Options 中設定在系統啟動時自動載入 Default Security。</p>
8	<p>在 Configure->Options 之中，請注意以下二屬性：</p> <p>Security User Name for Service 和 Security Password for Service。</p> <p>這些屬性將可允許 OPC Systems Service 被啟動執行在指定的使用者帳號，來給予遠端 Client 權限，以讀取 Calculation Tags 和 Data Logging，以及 Alarm Logging 的即時 Alarm。當遠端 OPC Systems Service 上的這些功能已被 Security 限制時，此設定即顯得相當重要。</p>
9	 <p>參考 VB.NET 的 Example 程式來了解如何以 Visual Studio 應用程式，程式化登入多個 client component。程式碼位於 VB.NET 的 Example 程式的 FormMain 中。</p> <p>OPC Controls.NET 的 LogIn 和 LogOff 方法是在 OPCControlsLogIn 控制項中。它控制使用者存取所有使用在 Visual Studio 應用程式中的 OPC Controls.NET component。ShowUserLogIn 方法顯示 Log In 對話方盒以作為 security 存取登入之用。以 CurrentUser 函式，我們可以得知目前的使用者。</p> <p>OPC Trend.NET component 程式界面擁有 LogIn 和 LogOff 方法，且也在工具列中提供 LogIn 圖示供登入之用。</p> <p>OPC Alarm.NET component 程式界面擁有 LogIn 和 LogOff 方法，且也在工具列中提供 LogIn 圖示供登入之用。</p> <p>OPC Web Controls.NET OPCWebRefresh 包含有 LogIn 和 LogOff 方法來控制特定的 Web 網頁。</p> <p>OPC Web Trend 控制項擁有 LogIn 和 LogOff 方法。</p>

	<p>OPC Web Alarm 控制項擁有 LogIn 和 LogOff 方法。</p> <p>OPC Systems Component 使用在各種設定的程式界面均提供有 LogIn 和 LogOff 方法。</p>
10	<p>Configure OPC Systems 程式中提供一個 Log In 功能表選項，以授予存取權限可進入受到 security 保護的 OPC Systems Service。例如 CSV Import 和 Export 的功能也是受到 security 權限的保護。</p>

第十章 預設設定

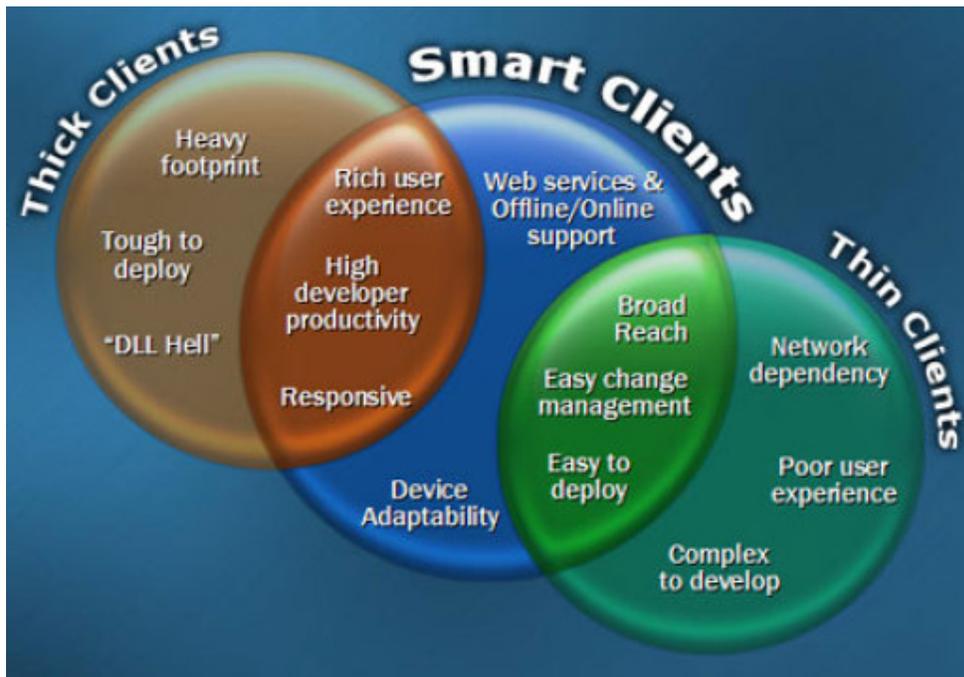
設定 Options

OPC Systems.NET Service 在每次啓動時，會自動地載入在 Configure->Options 中指定的預設設定檔。也有一些其他可調整的參數，包括啓動 OPC Server Watchdog 和重新載入即時趨勢與警報資料。以 Configure->Options 設定完成之後的設定，會自動產生 OPCSystems.options 檔。其放置的目錄是 OPC Systems.NET 系統目錄（預設是 C:\Program Files\Open Automation Software\OPC Systems.NET\）。

Step	Task
1	啓動 Configure OPC Systems 應用程式。
2	選擇 Configure → Options。 
3	選擇 Local OPC Systems Service。 
4	勾選 Load Default Tag Configuration On Service Start。 按下  按鈕找到 C:\OPCSystemsDemo\DemoTags.tags 檔案並開啓。
5	勾選 Load Default Data Logging Configuration On Service Start。 按下  按鈕找到 C:\OPCSystemsDemo\DemoLogging.DataLog 檔案並開啓。
6	勾選 Load Default Alarm Logging Configuration On Service Start。 按下  按鈕找到 C:\OPCSystemsDemo\DemoAlarmLogging.AlarmLog 檔案並開啓。
7	勾選 Load Default Report Configuration On Service Start。

	<p>按下  按鈕找到 C:\OPCSysytemsDemo\DemoReport.Reports 檔案並開啓。</p>
8	<p>如果需要的話，再勾選其他如 Default Alarm Notification Configuration、Default Recipe Configuration 和 Default Security Configuration，並找到和開啓相對應的設定檔。</p>
9	<p>勾選 Log Errors。預設的 Error Log path 是在 OPC Systems.NET 系統目錄（預設為 C:\Program Files\Open Automation Software\OPC Systems.NET\）下的 Log 目錄。</p> <p>我們可以重新指定到我們希望存放的目錄。</p>
10	<p>啓動 Transaction Logging 的時機，是僅當我們希望看到 OPC Server Items 在 Good Quality 下所收到的 raw value 時。所有 Bad Quality 的 OPC Server Items 只會出現在 Error Log。</p>
11	<p>勾選 Auto Runtime On Service Start。</p> <p><input checked="" type="checkbox"/> Auto Runtime On Service Start </p>
12	<p>建議設定 OPC Server Watchdog 頻率為 60 秒。</p>
13	<p>如果我們從遠端 OPC Systems Service 進行 data logging，或是儲存資料到遠端的資料庫系統，則請啓用 Store Data Logging Buffer to Disk on Network Loss or Database Engine Failure 功能。此舉可在網路暫時斷線時，將資料暫存在硬碟中，以空出記憶體空間，並在網路連線恢復時再傳送，避免資料遺失。</p> <p>啓用此功能之後，我們可以限制最大的硬碟儲存時間以防止因長時間的網路失效而造成硬碟暴滿。</p>
14	<p>如果我們已在 Configure->Tags 中的 Tag 設定有 Time On and Counts 的屬性，則我們可勾選啓用 Retain Times and Counts to File，並指定一個檔案來儲存設定，使得系統重新啓動時能夠重新取得此數值。</p>
15	<p>按下 Apply Changes 按鈕以儲存變更。</p> <p></p>

第十一章 Smart Client 開發



以 100% managed component 開發的應用程式可以被執行在遠端系統上，但不必安裝該應用程式在那遠端系統上。這使得應用程式的更新十足地簡單，且比 Web 應用程式提供了更佳的使用者經驗。 .NET Framework 提供了內建安全性，若無適當的授權則限制遠端程式的執行。

Visual Studio 2005 和 2008 的 Smart Client 程式開發精靈使得 Smart Client 程式開發變得非常容易。此精靈允許我們設定版本資訊，來讓遠端系統知道是否有更新版存在。此開發方式要求 Client 系統要安裝有 .NET Framework 2.0，如果是使用 Visual Studio 2005 開發；或者是安裝有 .NET Framework 3.5，如果是使用 Visual Studio 2008 開發。

COM component 或過時的 ActiveX controls 都不與 Smart Client 相容。所有 OPC Systems.NET 中的 .NET component 都是 Smart Client 相容。只要一些簡單的步驟，就可以佈署程式為 Smart Client 程式。以下是引導我們完整 Smart Client 執行的例子。

Smart Client 範例

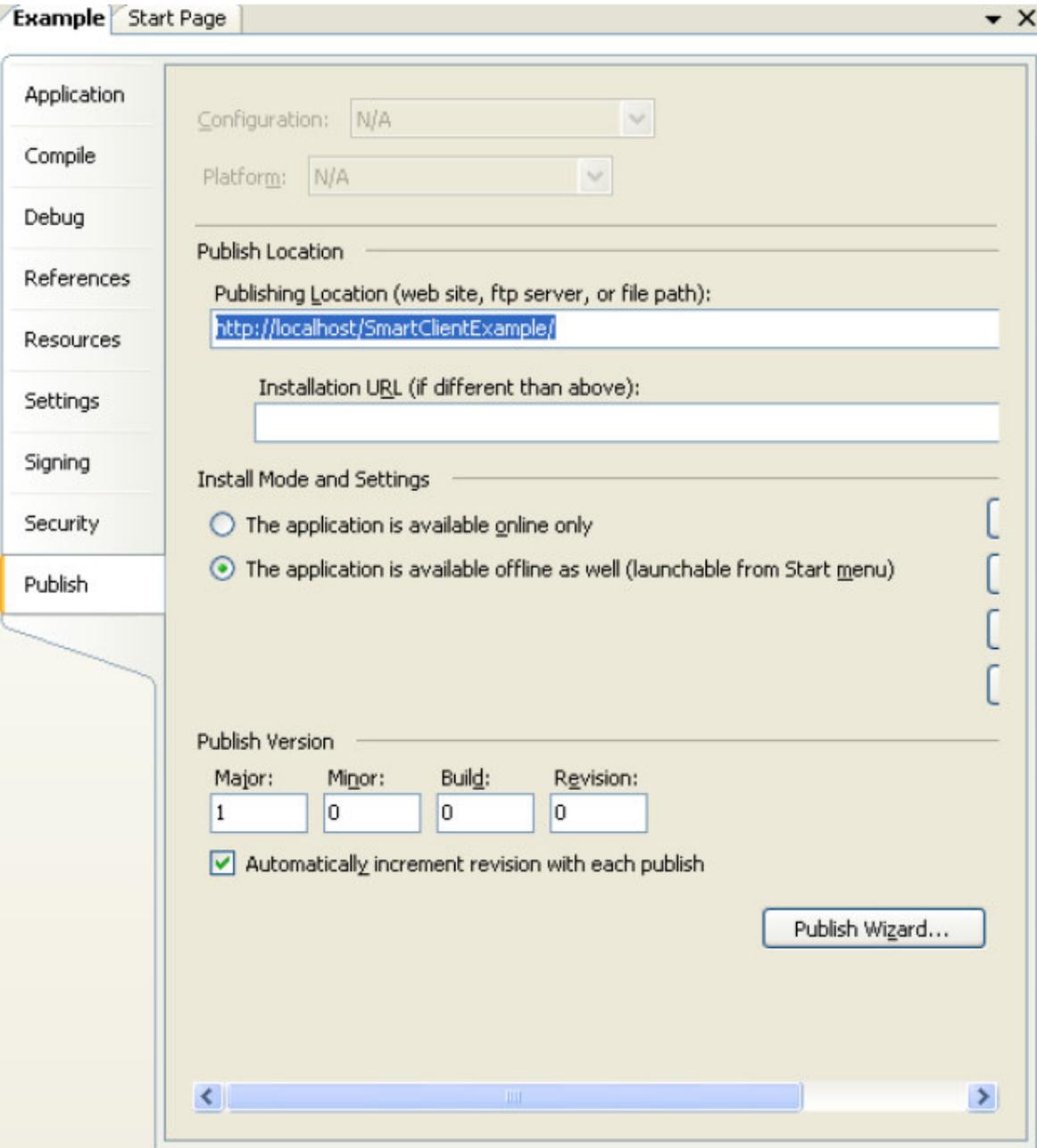
與 OPC Systems.NET 一併安裝進去的範例程式就是一個 Smart Client 相容的應用程式。此應用程式展示遠端連接至 OPC Systems.NET Service。這些應用程式包含 OPC Controls.NET、OPC Trend.NET、OPC Alarm.NET 和 OPC Systems configuration component。

當建立我們自己的 Smart Client 應用程式時，請記得包含想要連接的遠端 OPC Systems Service 的電腦名稱或是 IP 位址。此包括 OPC Controls Tag 屬性、OPC Trend pen Tag 名稱、OPC Alarm window network nodes，和使用 OPC Systems 程式界面建立 routines。

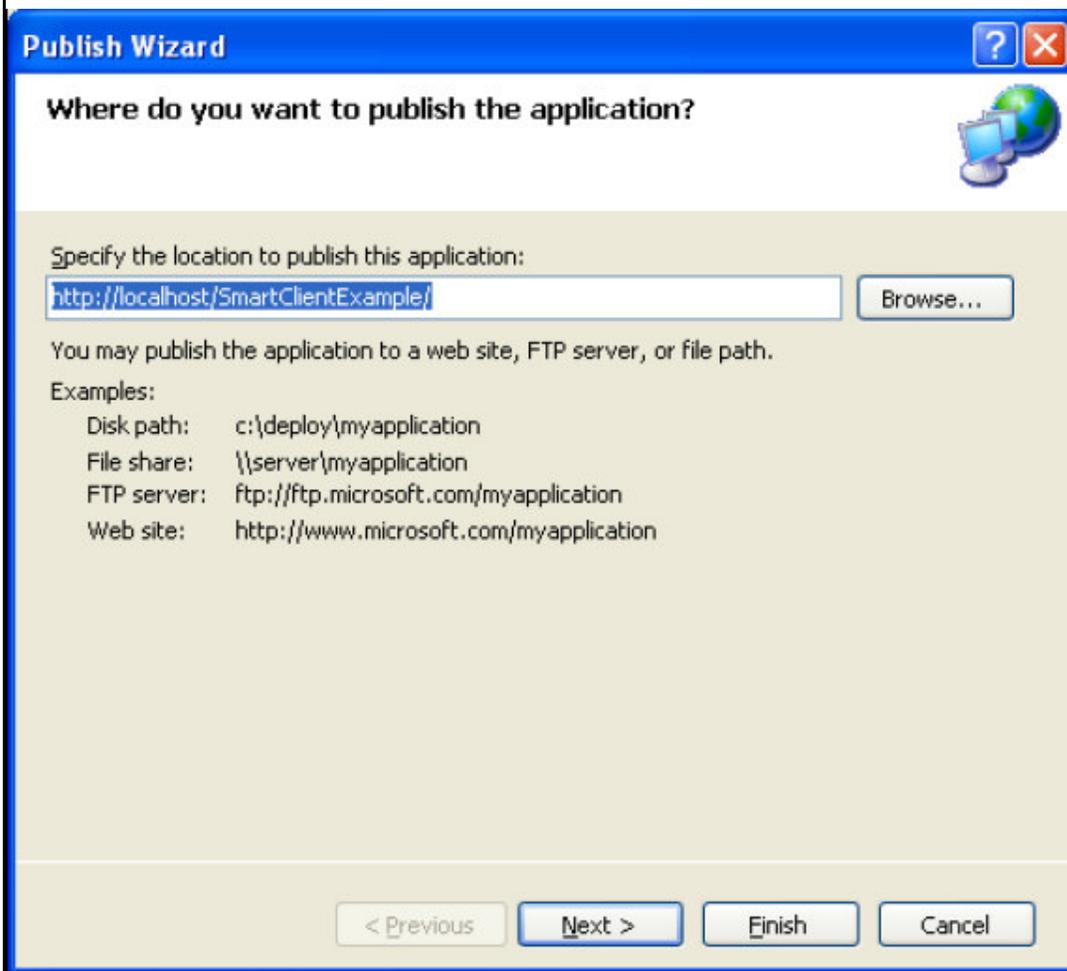
OPC Systems.NET 使用 TCP port 58723，所以請記得在防火牆的設定中開放此 port。

Smart Client Click Once 開發

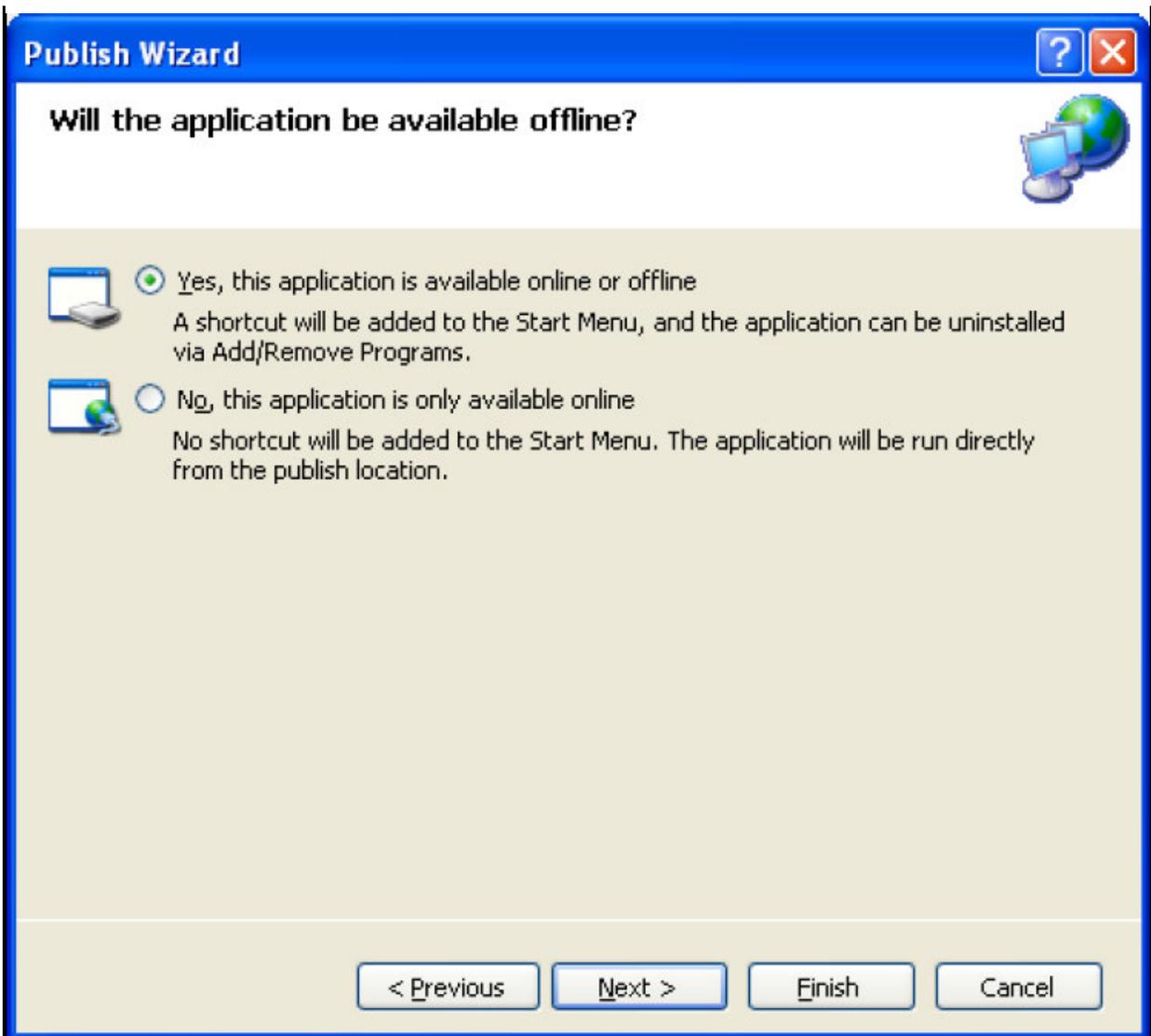
佈署以 Visual Studio 2005 搭配 .NET Framework 2.0 開發的程式是相當容易的。

Step	Task
1	確定 IIS 已安裝且執行在欲進行佈署的電腦系統中。
2	<p>在 Visual Studio 2005 或 2008 開發環境中選擇專案屬性。然後選擇 Publish 頁籤來查看 Click Once deployment 部份。</p> 
3	設定 Publish Location 為 local 或遠端執行有 IIS 的電腦系統。
4	設定起始 Publish Version，並保留為自動增加版本的預設設定，或是停用此功能並設成手動，讓我們在每次更新應用程式再去設定版本。

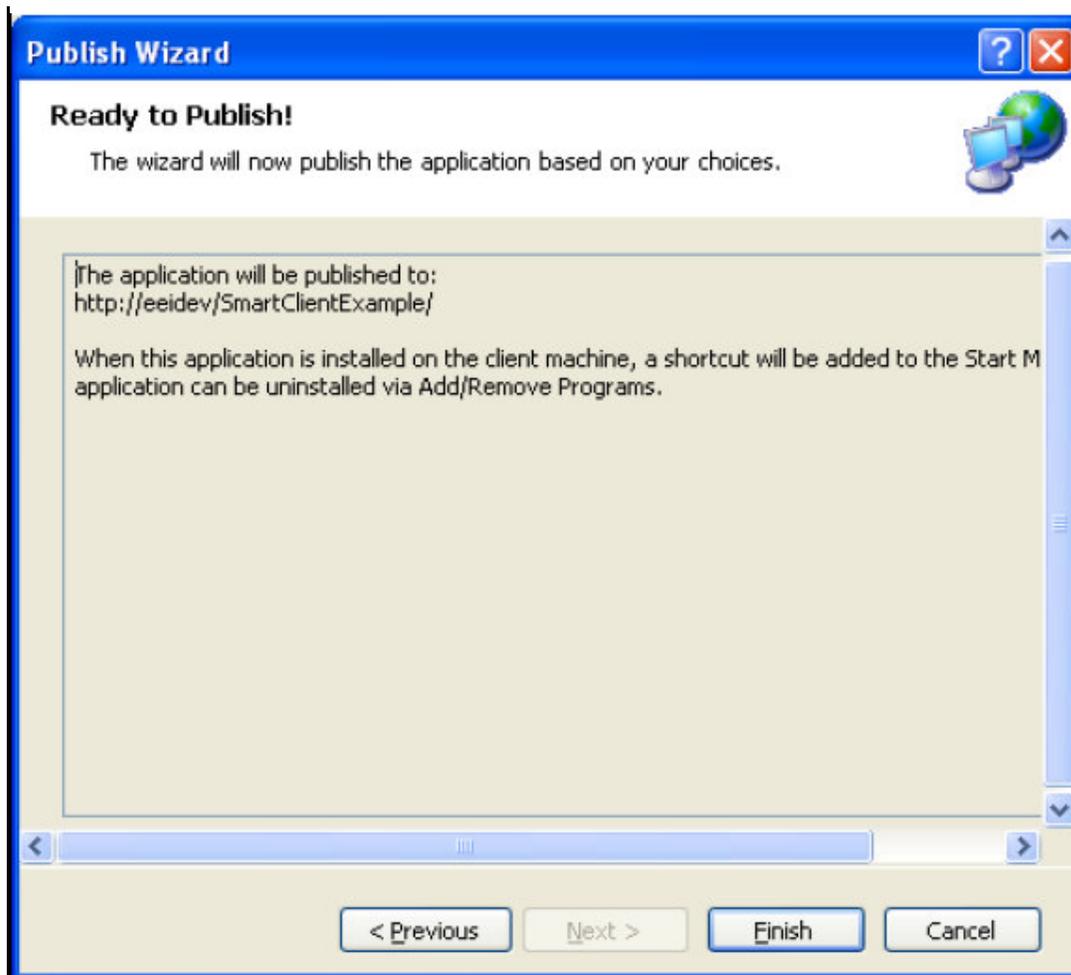
5 按下 Publish Wizard 按鈕來啟動佈署應用程式的步驟。



6 決定此被執行在遠端電腦系統的應用程式是否在與佈署主機斷線時能否執行。



7 確認佈署虛擬目錄是正確的，並按下 Finish 按鈕。



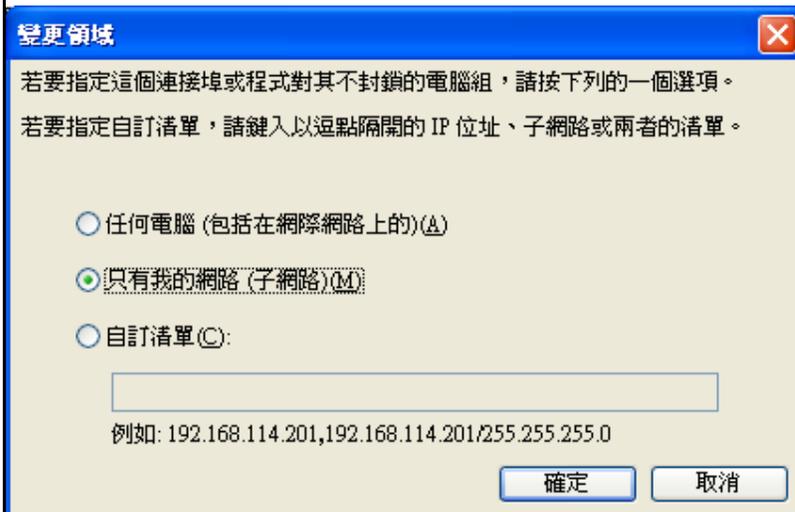
8 在所有將會使用的電腦並啟動有防火牆的系統中，開放 port 58723。

以下是 Windows XP SP2 中的範例。

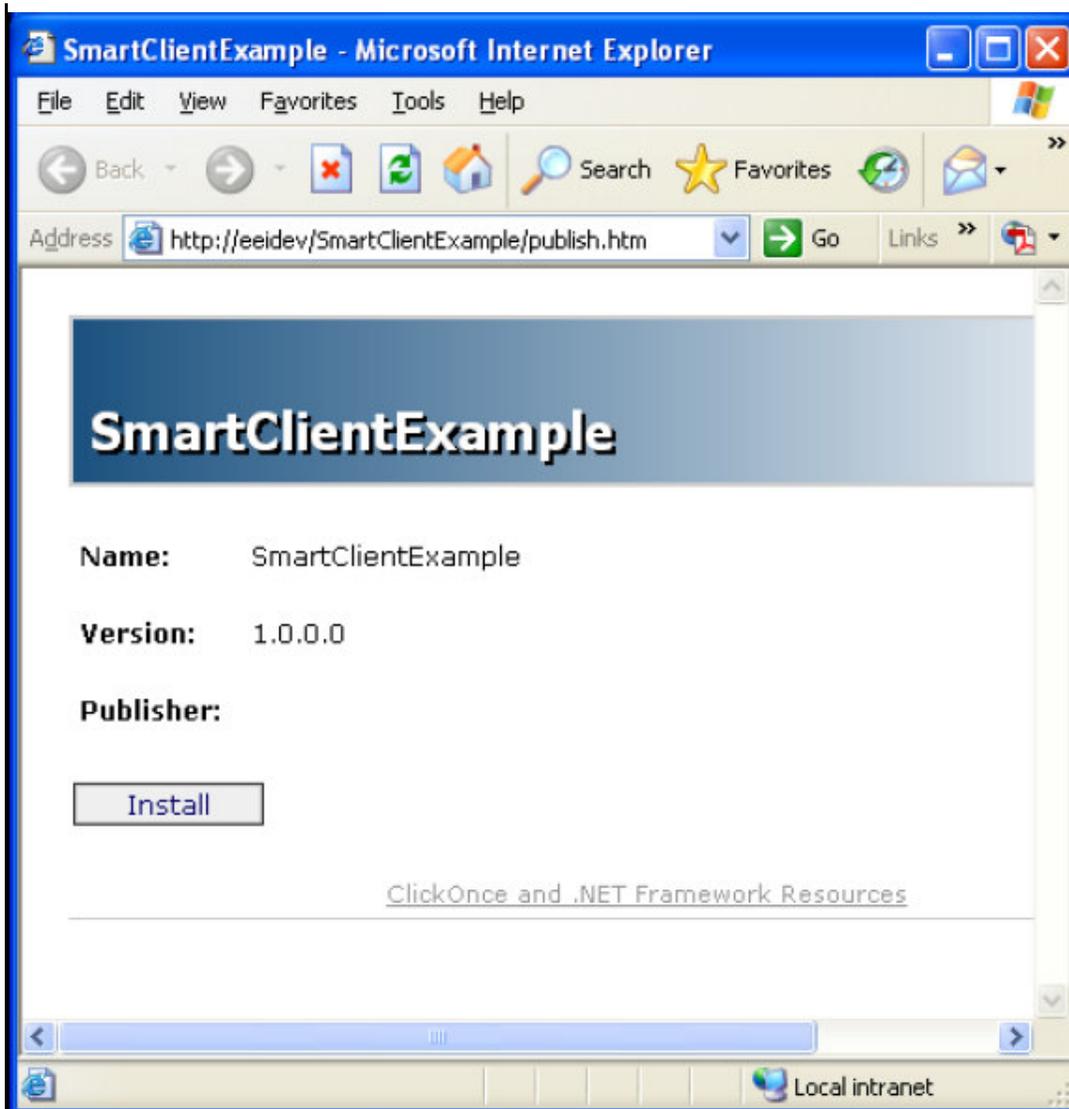
進入「控制台」並開啓「Windows 防火牆」。在「例外」頁籤中，按下「新增連接埠…」按鈕。輸入一個名稱和指定 Port Number 爲 58723。



按下「變更領域」按鈕來設定有效的電腦網路領域。



- 9 從任意遠端 Client 系統中，使用 Web browser 來鍵入佈署主機的 URL 網址。網頁成功出現後，按下網頁中的 Install 按鈕。



使用 Smart Client 開發方式，我們可以相當容易地佈署更新後的程式到開發的伺服器即可，所有遠端的使用者在他們下次執行他們自己系統中的此一程式，只要系統與佈署伺服器連線，即可自動獲得更新後的程式。

一旦此應用程式被安裝成 offline/online 佈署，則每次該程式被啟動執行時，它就會自行連接上佈署伺服器來檢查是否有更新的版本。如果有，則會提示使用者是否要安裝更新後的版本。