

## 0.1 TCP Library and Demo

1. 動機: 撰寫 TCP IP 的通訊程式，對於新手來說，並不友善。即使是資深的程式設計員，對於不斷的需要宣告 delegate，event，以及 Invoke，CallBack，Async 等，亦會覺得不耐煩，故已經有許多前輩寫出精采 TCP Model，可以降低入門門檻，可以不需要處理繁雜的 Callback 程序，但依然有一些問題，大部分的模組，只適用於 Console mode，運用於 Windows Form 時，依然要自己處理 Invok，因為 CallBack 等同使用另一個執行緒傳回內容，若不適當處理無法用於 Windows Form。另一個問題是部分撰寫者使用同步傳送，以減少 CallBack 的使用，但是此寫法會占用整個執行緒，使得使用者介面響應不佳，全程使用非同步傳送是較好的選擇。基於上述兩個問題，筆者已將 TCP 程序包裝成元件，且將回傳值，拉回了使用者介面執行緒。這使得使用 TCP IP 就像使用 Windows Form 中的 Button 一樣簡單。TCP IP 的另一個課題是，高於第三層的通訊協定，往往是一個 head struct，與複雜的內容 Struct，而 TCP IP 傳送的是 BYTE[]，未來也會提供將 Struct 與 byte[] 互轉的函數，使用者將不再需要了解 typeof 等技術，而可以傳送高於第三層的網路協定。

### 2. 使用於 windows form 說明

- (a) 下載 CT\_TCPIP\_Library 與 CT\_TCPIP\_Control
- (b) 開啟一個 windows Form 專案
- (c) 加入元件，於工具箱按滑鼠右鍵(圖 0.1)，選取選擇項目，開啟一個對話框(圖 0.2)，選擇瀏覽，選取 CT\_TCPIP\_Library.dll 與 CT\_TCPIP\_Control.dll 兩個檔案(圖 0.3)。

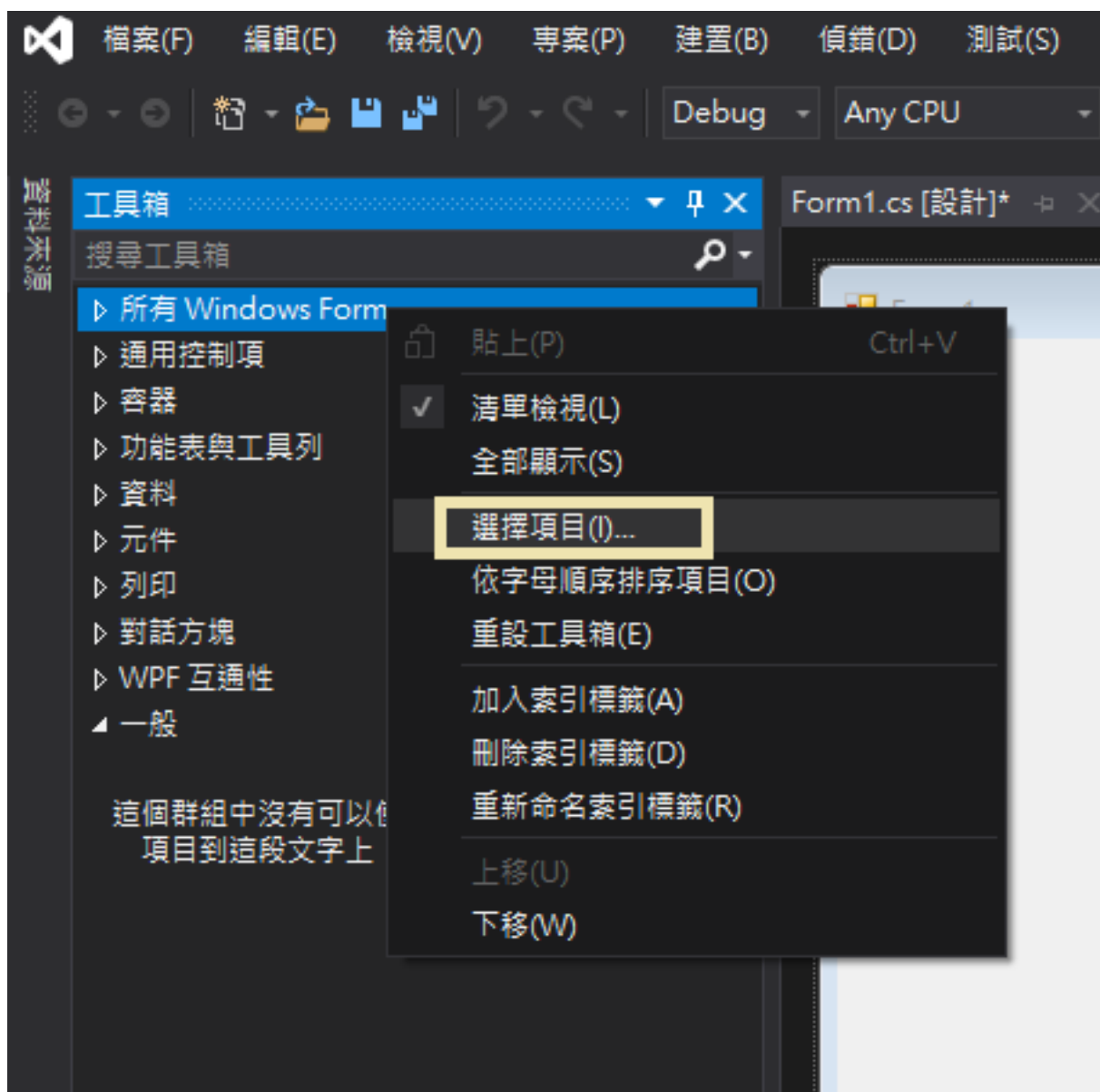


圖 0.1: 加入元件

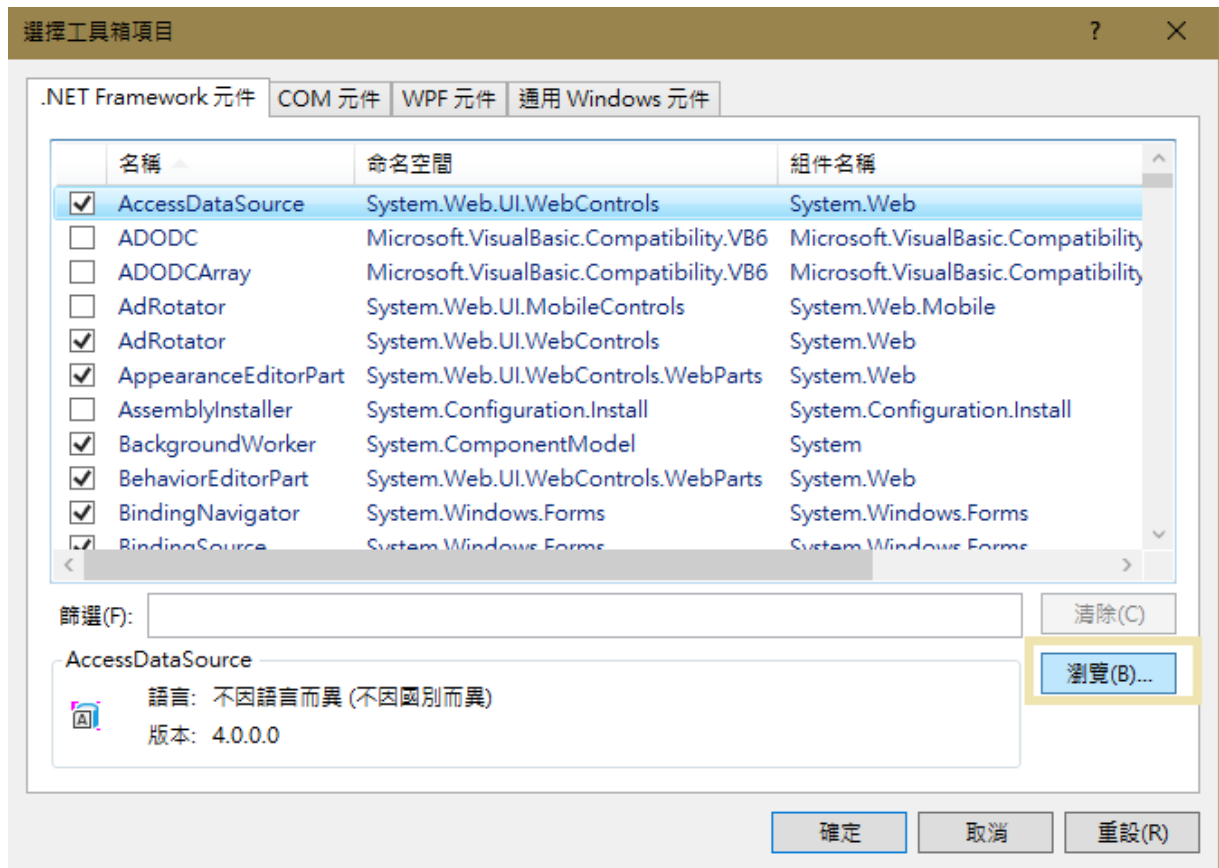


圖 0.2: 加入元件

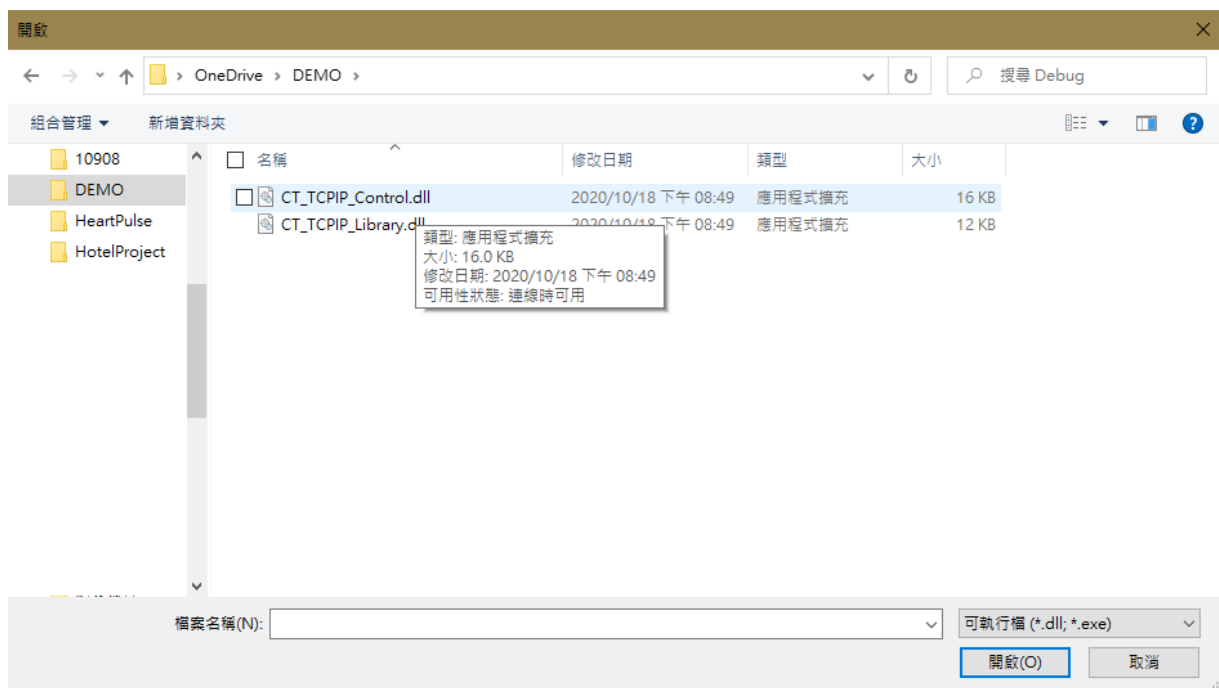


圖 0.3: 加入元件

(d) 可以在工具箱上看到兩個元件 TCP\_Server 與 TCP\_Client (圖 0.4)。

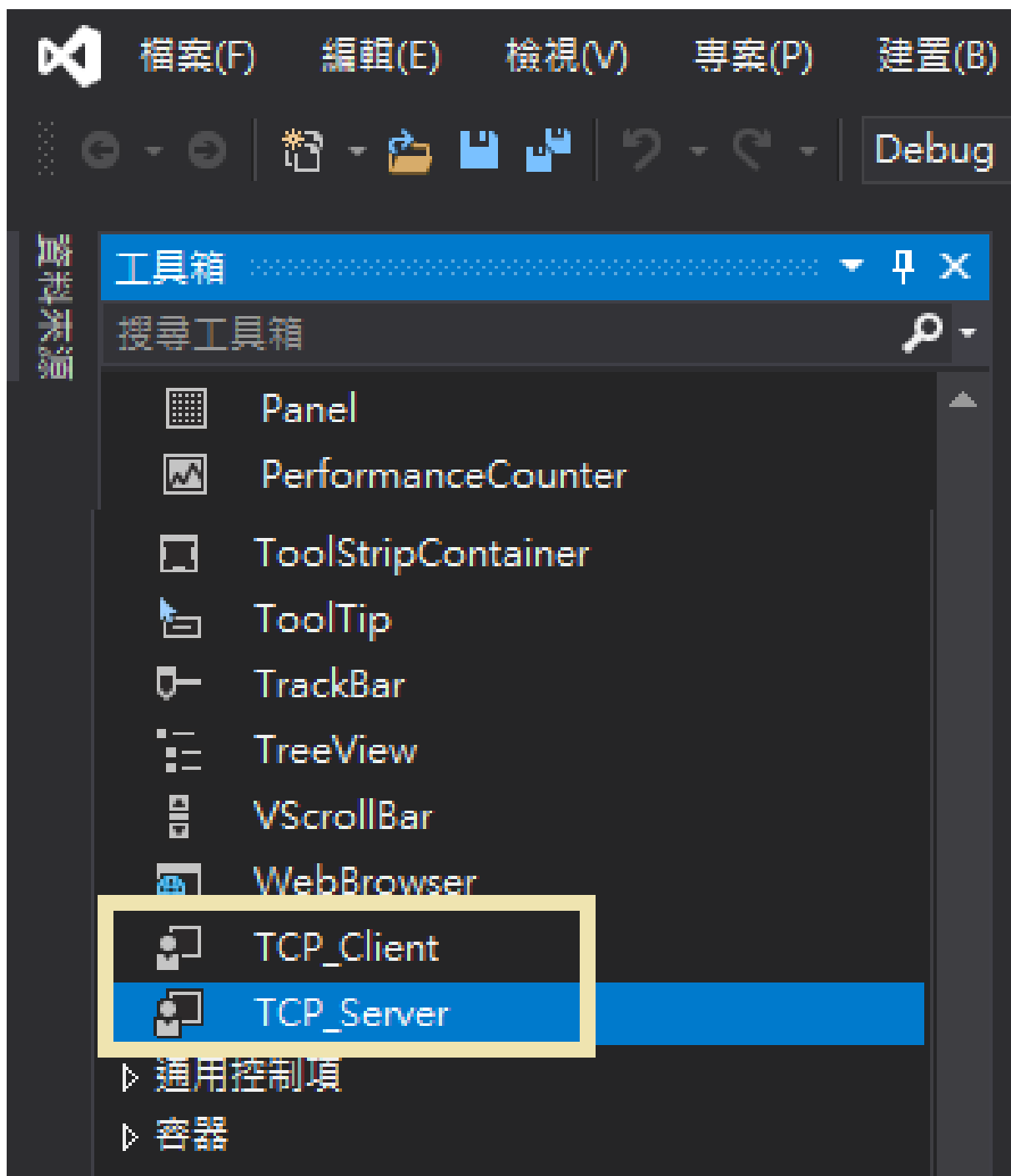


圖 0.4: 加入元件

(e) 將元件拉到 Form 上 (以 Client 為例)(圖 0.5)

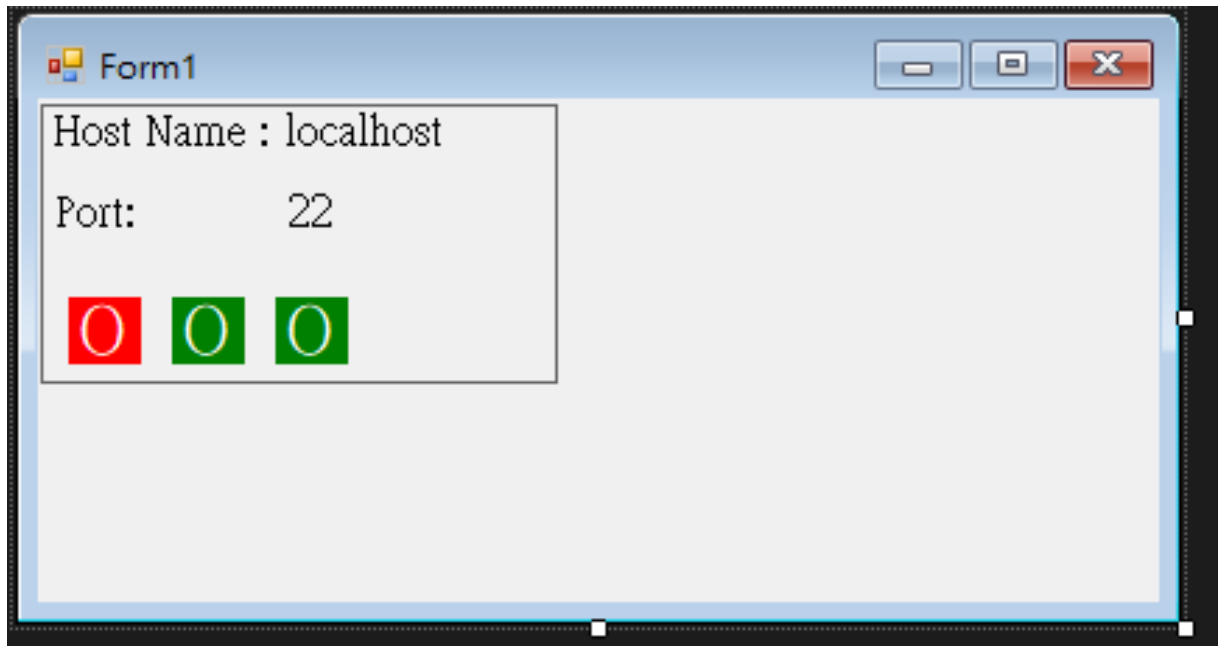


圖 0.5: 加入元件

(f) 加入要連接對象的 IP 與 PORT (圖 0.6)

```
private void Form1_Load(object sender, EventArgs e)
{
    tcP_Client1.HostName = "127.0.0.1";
    tcP_Client1.Port = 10022;
    tcP_Client1.Connect();
}
```

圖 0.6: 加入元件

(g) 於屬性視窗，點選事件，找尋到 ReviceTriger Event(圖 0.7)，並點兩下使其自動產生 Eeventt 程式碼(圖 0.8)。

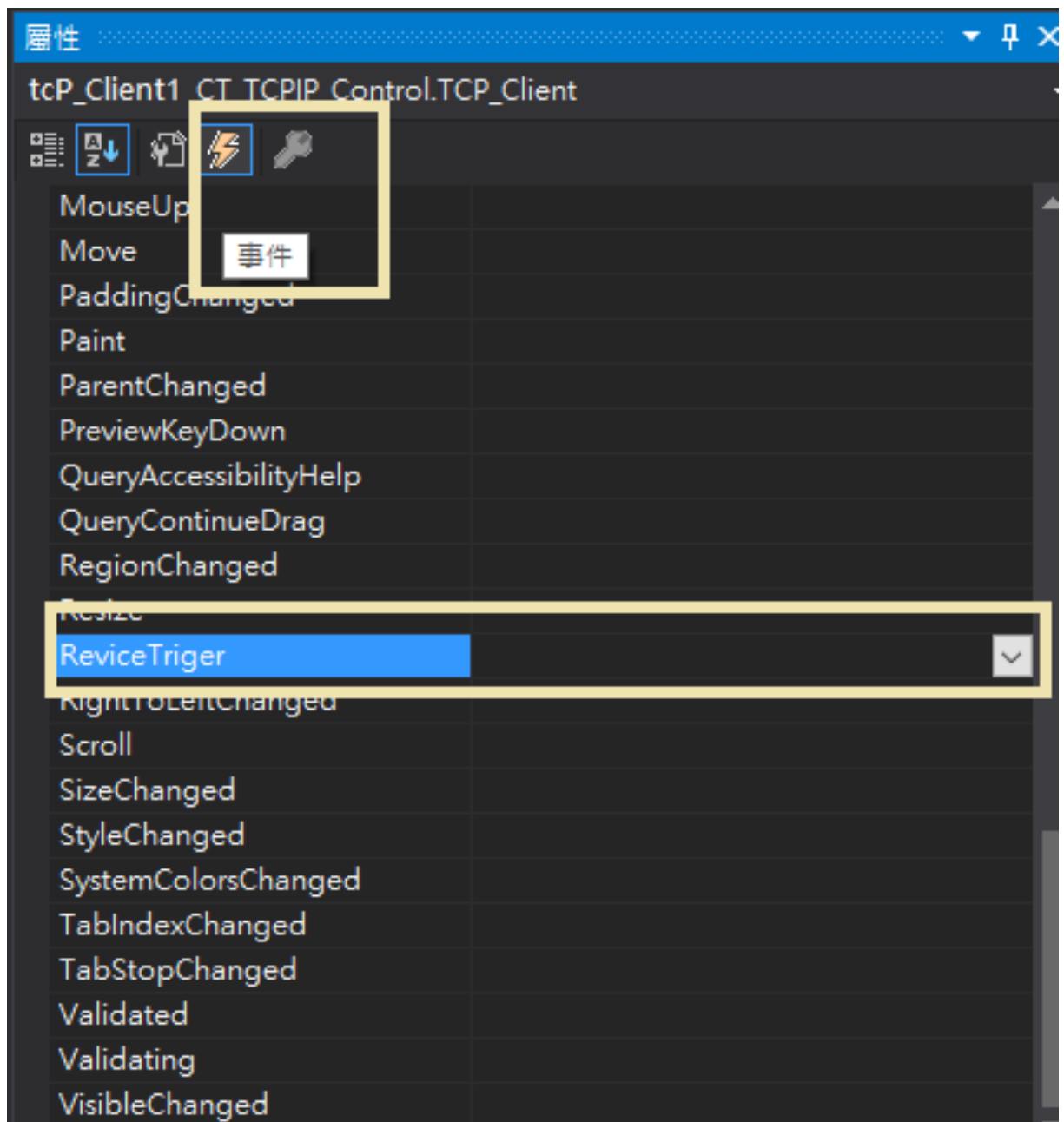


圖 0.7: 加入元件

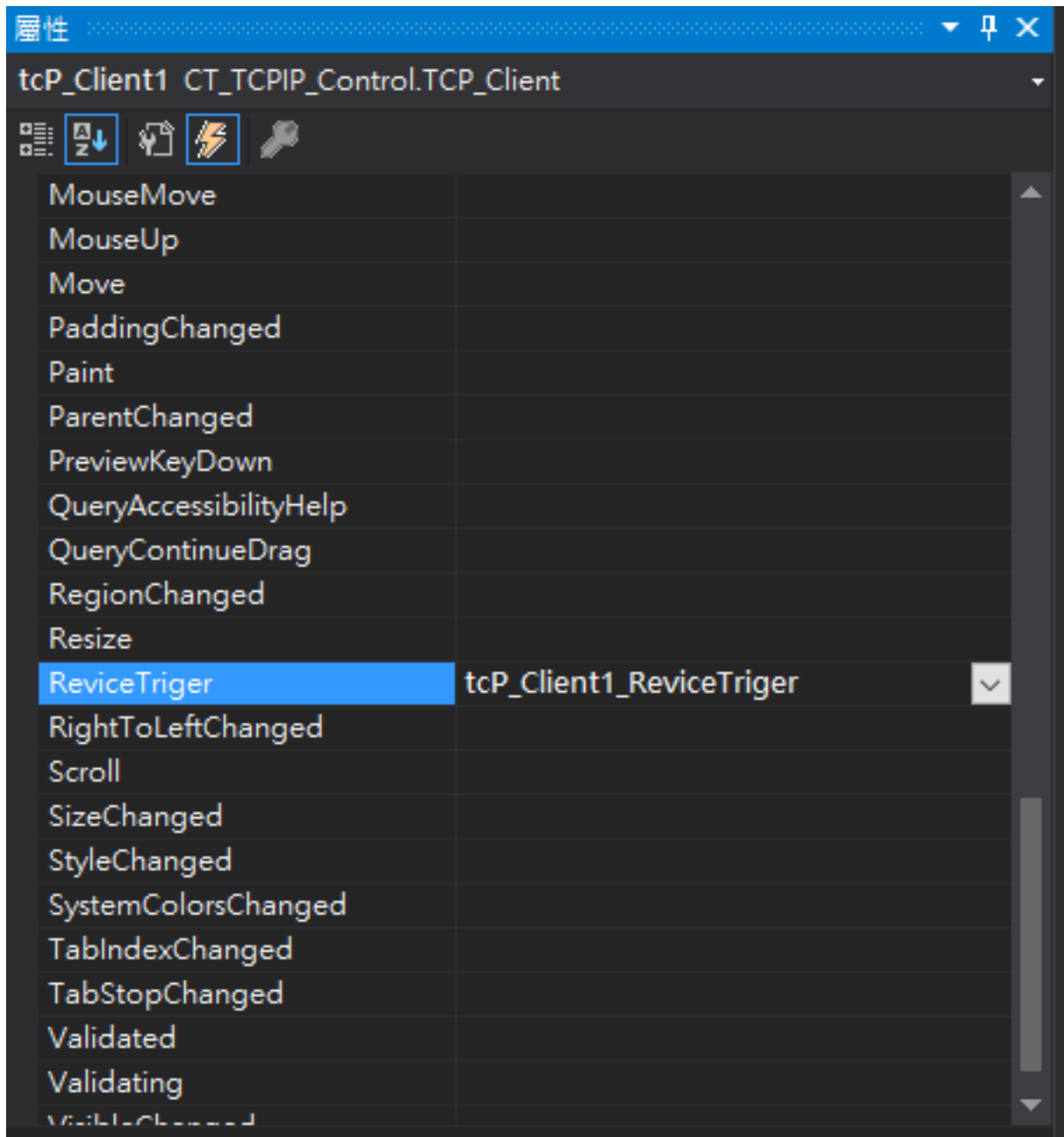


圖 0.8: 加入元件

(h) 於函數 `tcP_Client1_ReviceTriger` 內加入你要處理的程式 (圖 0.9)。

```

1 個參考
private void tcP_Client1_ReviceTriger(object sender, EventArgs e)
{
    //DO .....
}

```

圖 0.9: 加入元件