

Cogent DataHub WebView - Dynamic Binding

使用 WebView Scripting，能夠動態繫結 DataHub 資料點到 Symbol 和控制元件，並利用這些能力，來建立樣板頁面。

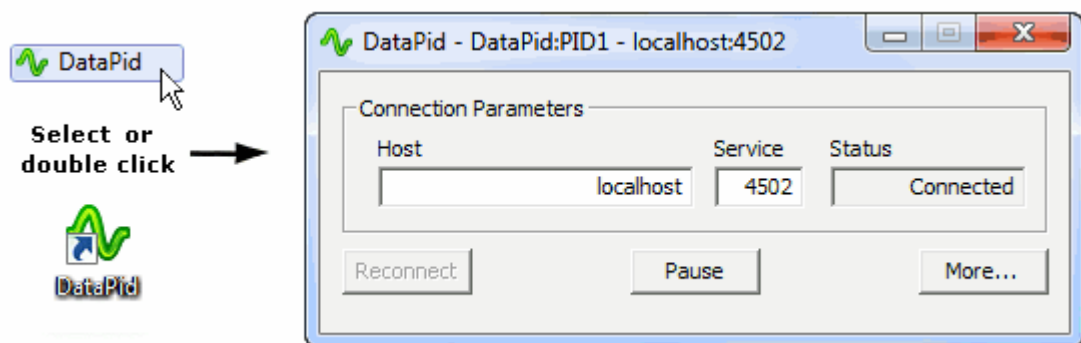
Dynamic Point Binding (動態點繫結)

動態的點繫結允許你在執行期間內修改一個關連到控制元件的 DataHub 資料點。本教程介紹兩種不同的方式來做到這一點 (簡易陣列和動態列表)，使用兩個控制元件 - ComboBox 和 ListBox。

Combo Box 控制元件

使用一個簡易陣列來將 ComboBox 繫結到一個圓形儀表元件上

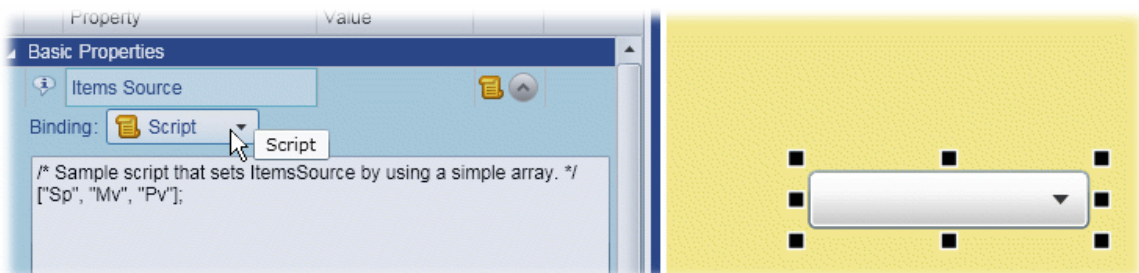
1. 從 Windows 的開始選單、命令列，或通過點擊桌面上的圖示來開啟 DataPid 程式。



只要 DataPid 啟動後，它試圖連接到一個 DataHub，並開始產生資料。

2. 開啟 DataHub WebView，並開啟一個新的頁面，在頁面中新增一個 ComboBox 控制元件。
3. 在 ComboBox 的 **Basic Properties** 基本屬性中，替 **Items Source** 選擇 **Script** 為連結類型，然後編輯預設的 script 來讀取：

```
["Sp", "Mv", "Pv"];
```




在這列 script 內，“[“ 以及”]”字元告訴 WebView scripting 引擎，這是一個逗號分隔的字

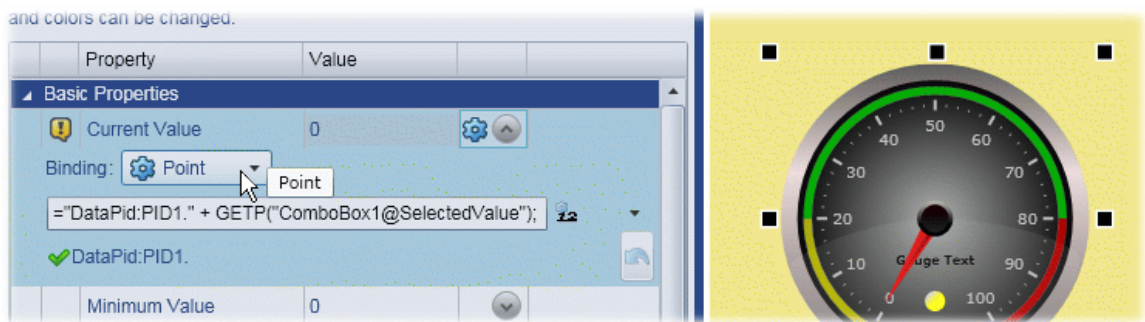
串陣列，而這其中的每一個陣列元素都應該被分配一個值到 Combo Box 控制元件中。



替代性使用陣列，使得能夠利用 script 來列出控制元件中的選項。

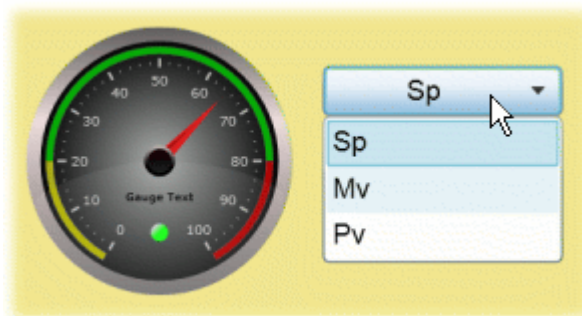
4. 新增一個圓形儀表 2(Circular Gauge 2)  控制元件到頁面中。
5. 在圓形儀表 2(Circular Gauge 2)的 **Basic Properties** 中，替 **Current Value** 選擇 Point 為繫結類型，並建立一個 script：

```
= "DataPid:PID1." + GETP("ComboBox1@SelectedValue");
```




當一系列 script 被鍵入作為點繫結，就像我們在這裡看到的，等號“=”會告訴 WebViewsript 引擎，指定的點名稱為何。在這個例子中，允許點名稱是由連接的字串所構成。GETP 函數能夠取得控制元件屬性。GETP 參數的語法是一個字串，包括控制元件的名稱，一個 @ 符號，以及一個你需要得到的參數名稱。

6. 切換到執行模式，並從 Combo Box 中選擇您要的選項，然後在圓形儀表中檢視結果。

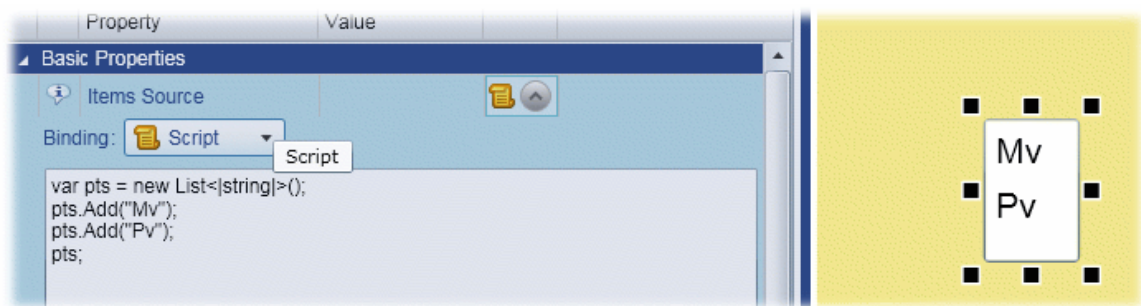


List Box 控制元件

使用 script 來將 List Box 繫結在 Trend Chart 中

1. 確定 DataPid 程式正在執行，或是從 Windows 的開始選單的命令行中，或通過點擊桌面上的圖示來啟動它。
2. 開啟 DataHub WebView，開啟一個新的頁面，並在頁面中新增一個 List Box  控制元件。
3. 在 List Box 的 Basic Properties 中，替 **Items Source** 選擇 **Script** 為繫結類型，然後編輯預設的 script 來讀取：

```
var pts = new List<|string|>();  
pts.Add("Mv");  
pts.Add("Pv");  
/* More strings can be added ... */  
pts;
```

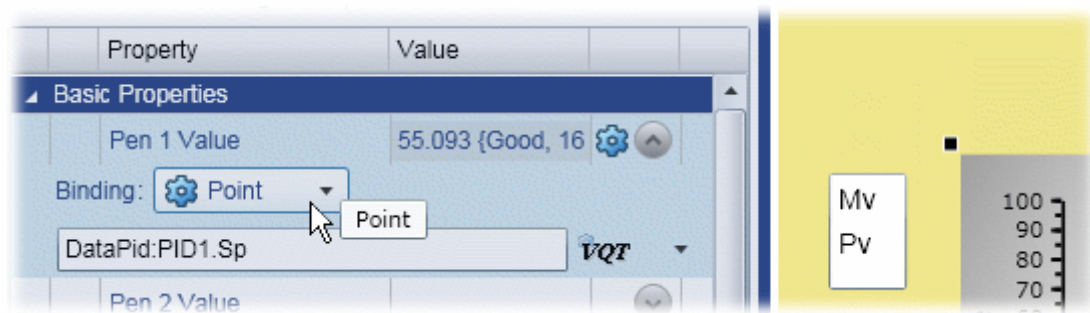


此範例 script 建立了一個字串列表，然後將兩個字串（在這種情況下為點名稱）新增至列表中。這份名單可以容納任何數量的字串。script 的最後一行呼叫 pts 變數，它將會使那些被傳遞到列表框中的字串列表作為 List Box 來源的項目。



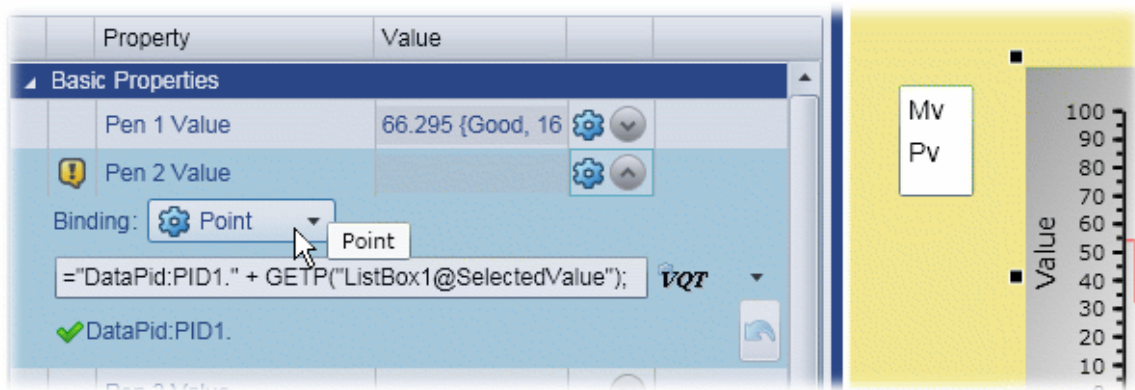
替代性使用 script，能夠使用陣列來列出項目，正如上面的 Combo Box 控制元件選項的解釋。

4. 在頁面上新增 Trend Chart (3 pens) 控制元件 。
5. 在 Trend Chart 控制元件的 Basic Properties 中，替 **Pen 1 Value** 選擇 **Point** 為繫結類型，並輸入值為 **DataPid : PID1.Sp**。

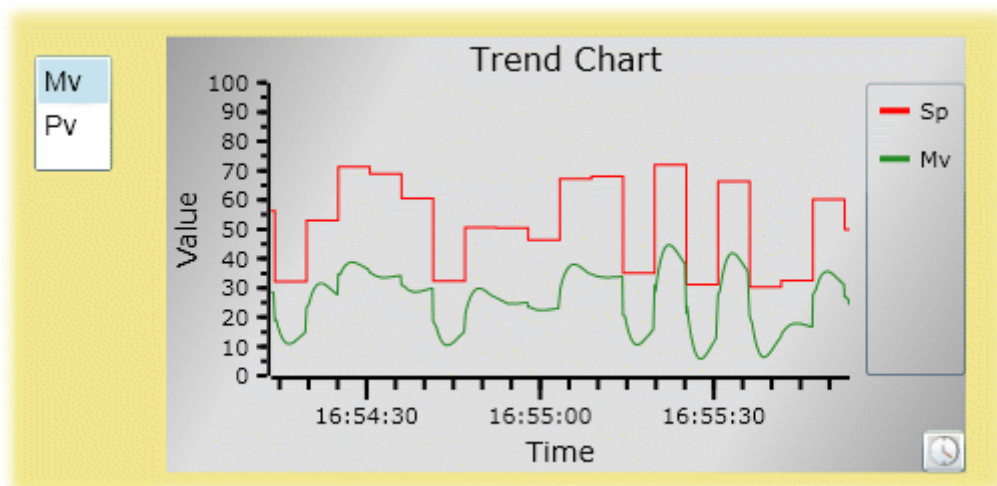


6. 為 **Pen 2 Value** 選擇 **Point** 為繫結類型，並建立一個 script：

```
= "DataPid:PID1." + GETP("ListBox1@SelectedValue");
```



7. 切換到執行模式，並從 List Box 中選擇您要的選項，然後在 Trend Chart 中檢視結果。




Dynamic Control and Symbol Binding (動態控制元件和符號繫結)

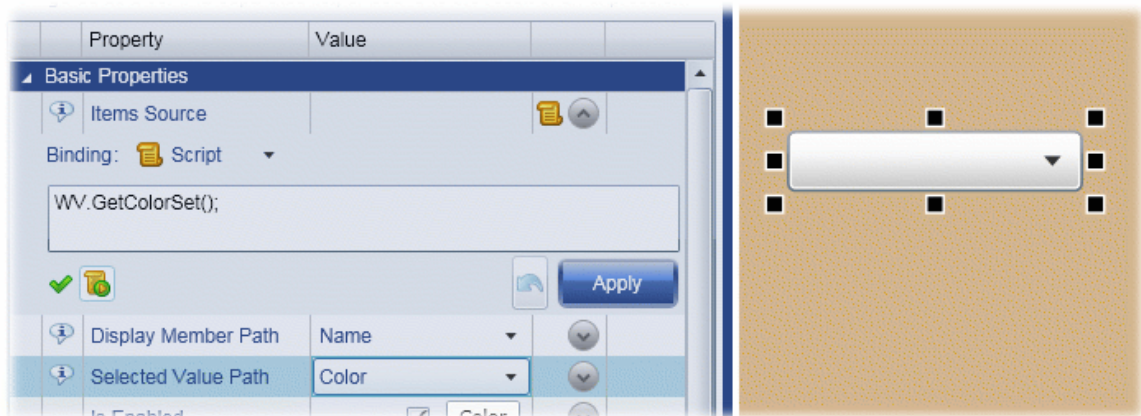
動態控制元件和符號繫結允許您在執行期間內更改在控制元件或符號上的繫結。本教程介紹如何使用一個 Combo Box 來改變 Shining Light 的顏色，以及如何呈現 Shining Light 開關符號的開與關的狀態。

控制元件繫結

將 Combo Box 繫結到燈號符號上

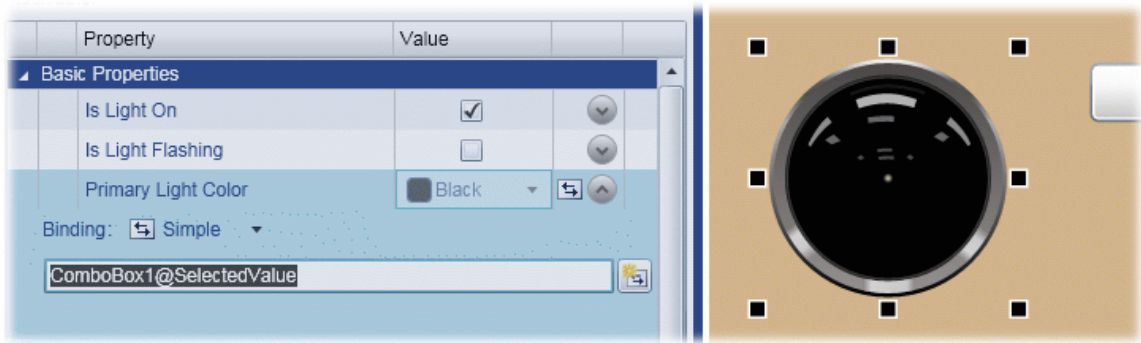
1. 開啟 DataHub WebView，開啟一個新的頁面，並在頁面中新增一個 Combo Box 控制元件 。
2. In the **Basic Properties** of the Combo Box, for the **Items Source**, select the **Script** binding type and edit the default script to read:
3. 在 Combo Box 的 **Basic Properties** 基本屬性中，替 **Items Source** 選擇 **script** 為繫結類型，然後編輯預設的 script 來讀取：

```
WV.GetColorSet();
```

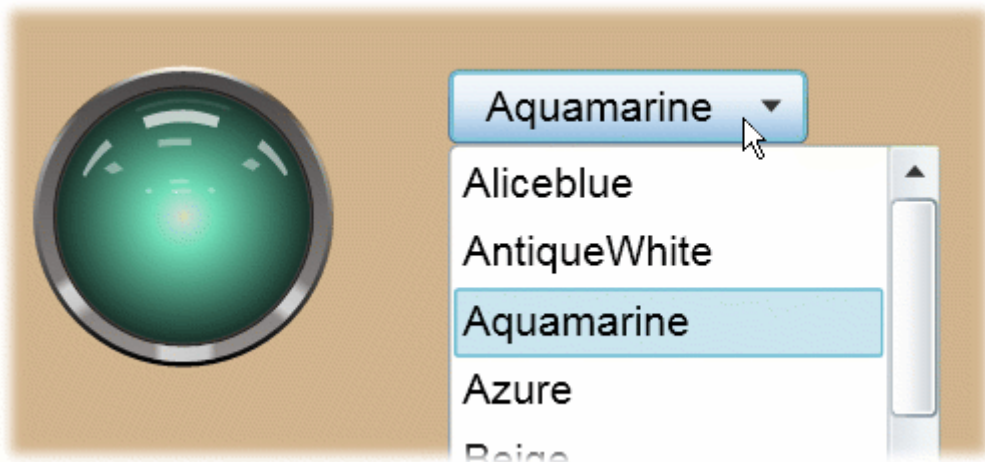


4. 在 **Display Member Path** 屬性中，選擇 **Name**。
5. 在 **Selected Value Path** 屬性中，選擇 **Color**。
6. 在頁面上新增一個 **Shining Light** 控制元件 。
7. 在 Shining Light 的 **Basic Properties** 中，替 **Primary Light Color** 選擇 **Simple** 的繫結類型，並輸入值：


```
ComboBox1@SelectedValue
```

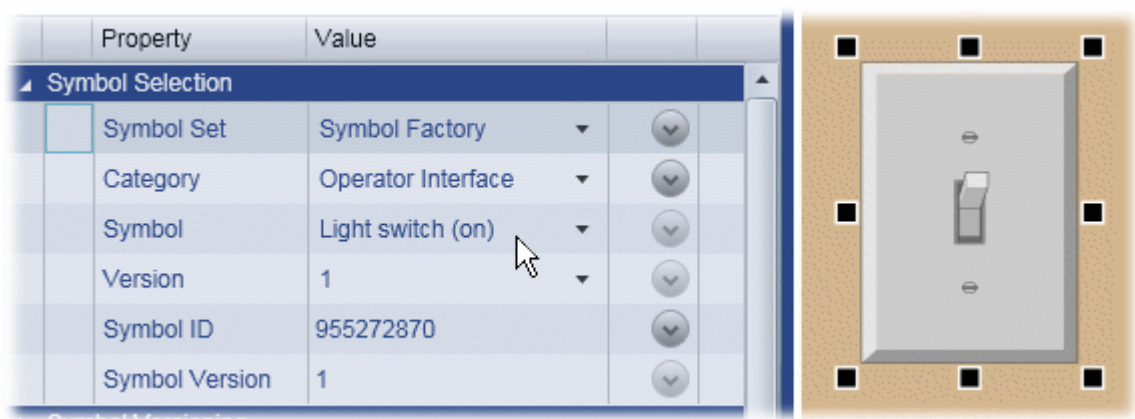


8. 切換到執行模式，並從 List Box 中選擇您要的選項，然後在執行頁面中檢視結果。

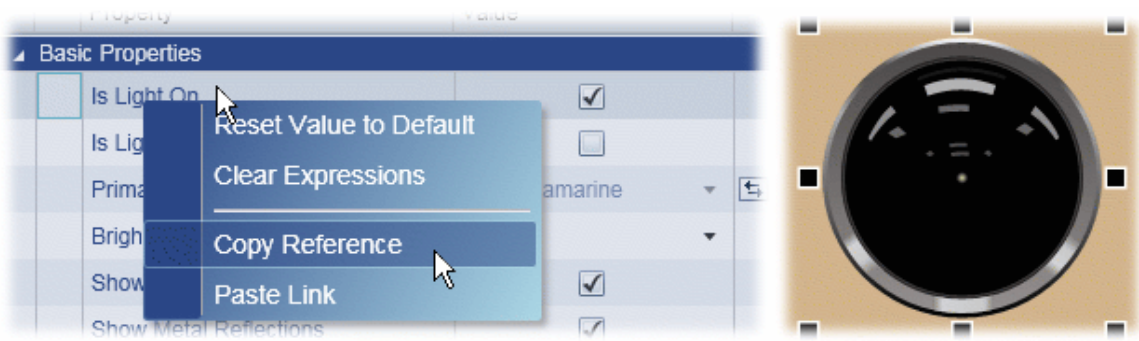


將更改燈號的開關符號連結在模擬開啟和關閉上

1. 使用與上述相同的頁面，新增一個符號控制元件 。
2. 在 **Symbol Set** 中，替 **Symbol Selection** 選擇 **Symbol Factory**，然後替 **Category** 選擇 **Operator Interface**，接者是替 **Symbol** 選擇 **Light switch (on)**。

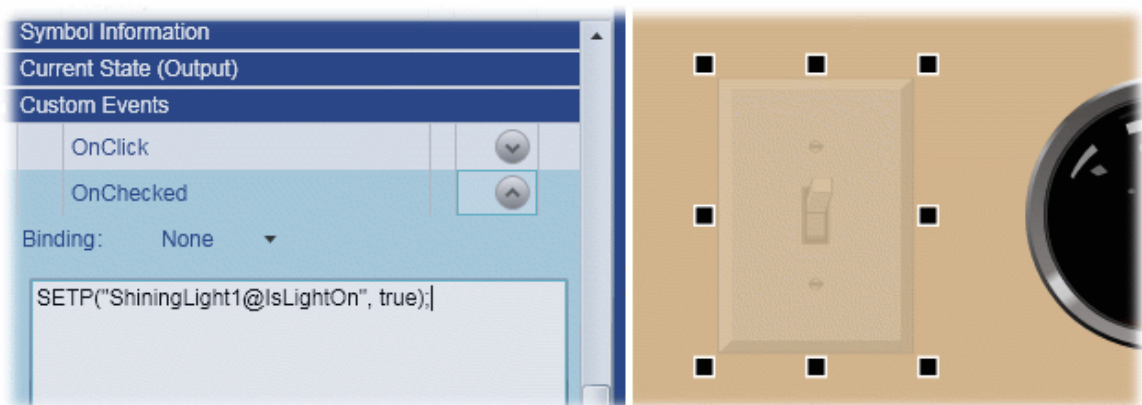


3. 選擇 **ShiningLight** 控制元件，然後在 **Basic Properties** 中，對 **Is Light On** 點擊滑鼠右鍵，然後複製參考。



4. 選擇符號 **light switch**，然後在 **Custom Events** 中替 **OnChecked Event** 輸入：

```
SETP("ShiningLight1@IsLightOn", true);
```



對於這一步，以及下一步，字串 `ShiningLight1@IsLightOn` 是你從 `Shining Light` 控制元件複製的。你只要直接貼上它。

5. 在 **Custom Events** 中，替 **OnUnChecked Event** 輸入：

```
SETP("ShiningLight1@IsLightOn", false);
```

6. 切換到執行模式，然後按一下 **light switch** 符號來開啟和關閉 `Shining Light`。



此動作正常，但如果在燈號熄滅時，`light switch` 仍處於 `ON` 的位置，就有點奇怪了。接

下來我們會解決這個問題。

7. 選取符號 light switch，然後在 **Custom Events** 中，替 **OnChecked Event**，再新增一行：

```
SETP("ShiningLight1@IsLightOn", true);  
SETP("@SymbolID", 955272870);
```

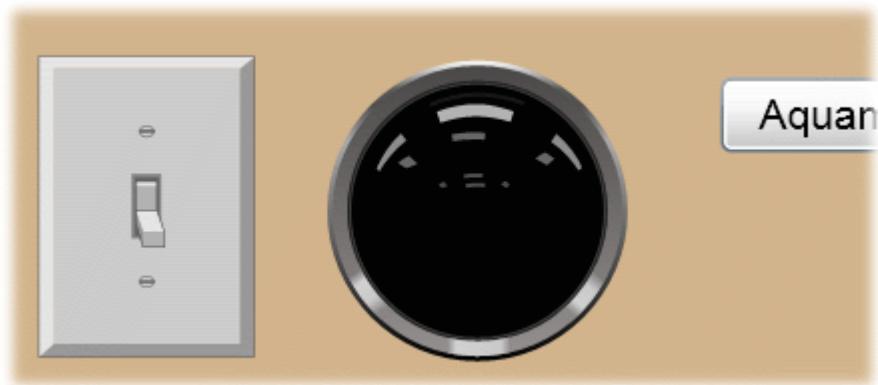


對於這一步，以及下一個步驟，字串@SymbolID 是參照符號本身的。

8. 在 **Custom Events**，替 **OnUnChecked Event** 再新增一行：

```
SETP("ShiningLight1@IsLightOn", false);  
SETP("@SymbolID", 1685125442);
```

9. 切換到執行模式，然後按一下 light switch 符號來開啟和關閉 Shining Light。



現在，Light switch (on)以及 Light switch (off)的符號將會隨著燈號的開啟以及關閉的改變作變化了。

建立一個樣版頁面

本教程介紹了如何建立一個頁面，在此頁面中，您可以點擊一個按鈕來切換顯示區內的資料來源。

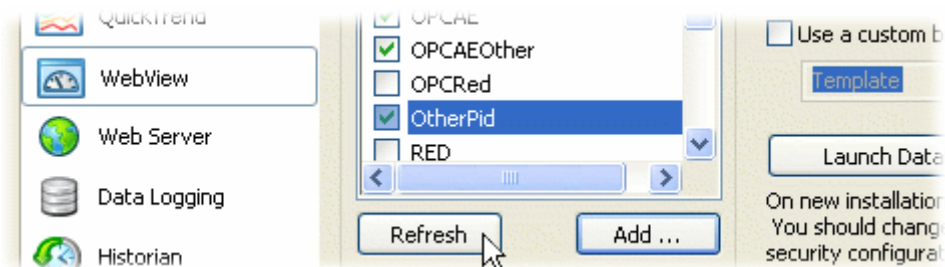
建立一個一致的資料來源

1. 在 DataPid 執行的過程中，開啟另一個 DataPid 執行個體。在第二個 DataPid 執行個體中，點擊 **More...** 按鈕來開啟 DataPid 的 **Configurable Options**。
2. 將 **Domain** 修改為 **OtherPid**，然後點擊 **Apply Changes** 按鈕。然後按下 **Reconnect** 按鈕。




如果你仔細觀察 DataHub 資料瀏覽器，你應該看到一個新的資料領域 **OtherPid** 的資料變化值。現在，我們需要新增 DataHub WebView 中的資料領域。

3. 選擇 DataHub 屬性視窗中的 **WebView** 選項，在 WebView 中可見的 **Data Domains Visible to WebView** 的這個部分，點擊 **Refresh** 按鈕。

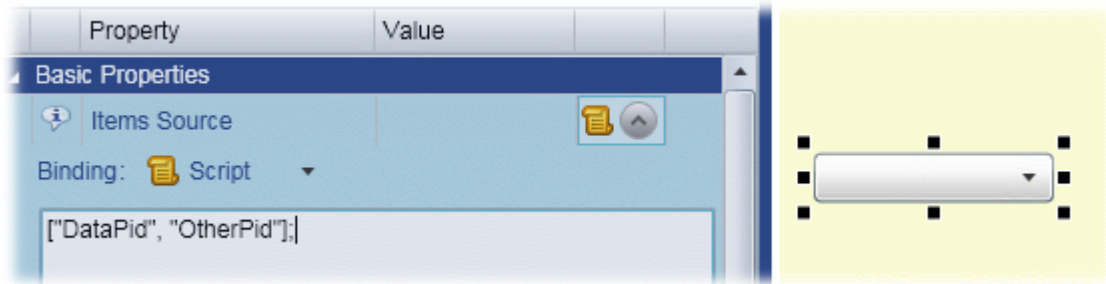


4. 領域名稱 **OtherPid** 應該會出現在列表中。勾選該複選框來使我們能夠在 DataHub WebView 中看到 **OtherPid** 領域，也讓我們能夠存取該領域的資料。現在我們有兩個包含不同資料點的相同資料集來展示我們的樣版頁面。

建立樣版頁面

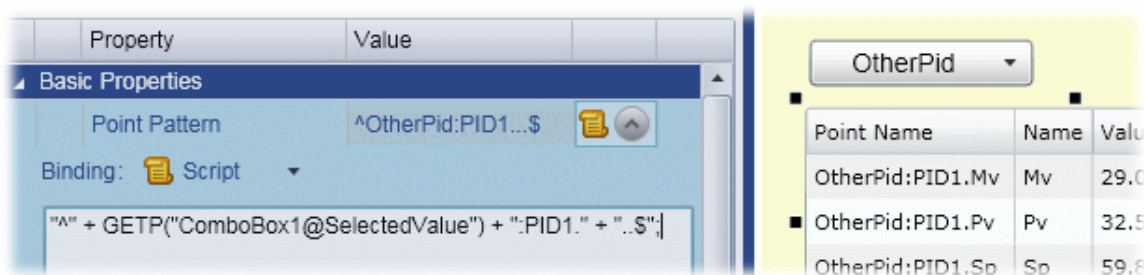
1. 啟動 DataHub WebView，開啟新的頁面，然後在頁面中新增一個 Combo Box 控制元件 
2. 在 Combo Box 的 **Basic Properties** 中，替 **Items Source** 選擇 **script** 為連結類型，然後編輯預設的 script 來讀取：

```
["DataPid", "OtherPid"];
```



3. 在頁面上新增資料表控制元件 。
4. 在資料表的控制元件的 **Basic Properties**，替 **Point Pattern**，選擇 **script** 為繫結類型，然後編輯預設的 **script** 來讀取：

```
"^" + GETP("ComboBox1@SelectedValue") + ":PID1." + "..$";
```



^這個符號允許任何的前綴字元做組合，而 GETP 則表示從 ComboBox 內取出數值。而..\$ 字串允許在其所需要的 PID1. 字串後面的任何後綴字元做組合。

5. 為了使資料表更容易閱讀，在 **Table Columns** 屬性中，您可以顯示資料表中後面的欄位：**Point Name**，**Display Name**，**Value** 和 **Quality**。
6. 切換到執行模式，並將 ComboBox 內的資料領域從 **DataPid** 改為 **OtherPid**，來檢視兩種不同資料集的資料點。

在頁面上新增 Trend Chart

1. 在頁面上新增 Trend Chart(3 pens)控制元件 。
2. 在 Trend Chart 的 **Basic Properties** 中，替 **Pen 1 Value** 選擇 **Point** 為繫結類型，並輸入：

```
=GETP("ComboBox1@SelectedValue") + ":PID1.Sp";
```



當一列 script 被鍵入作為點繫結，就像我們在這裡看到的，等號"="會告訴 WebViewscript 引擎，指定的點名稱為何。在這個例子中，允許點名稱是由連接的字串所構成。GETP 函數能夠取得控制元件屬性。GETP 參數的語法是一個字串，包括控制元件的名稱，一個 @ 符號，以及一個你需要得到的參數名稱。

3. 為了提供給資料點 Sp 的趨勢線一個更好的形狀，在 Pen1 Properties 中，勾選 **Pen 1 Is Square** 以及 **Pen 1 Auto Extend** 選項。
4. 回到 **Basic Properties**，替 **Pen 2 Value** 選擇 **Point** 為繫結類型，然後輸入：

```
=GETP("ComboBox1@SelectedValue") + ":PID1.Mv";
```

然後替 **Pen 3 Value**，選擇 **Point** 為繫結類型，然後輸入：

```
=GETP("ComboBox1@SelectedValue") + ":PID1.Pv";
```

現在你應該可以看見所有三筆資料的趨勢線了。

5. 切換到執行模式，並再次將 ComboBox 內的資料領域，從 **DataPid** 改為 **OtherPid**。現在，兩種不同的點資料集在資料表和 Trend Chart 之間交替著。

跟隨這個例子，你可以建立一個頁面來顯示多個相同的資料集在任一組控制元件中。利用只需點擊一個按鈕就可在資料集之間切換的能力。