

章節 5. Excel連接

目錄

5.1. 把資料放進Excel

5.1.1. 方法 1 - 使用DDEAdvise進行拖放功能

5.1.2. 方法 2 - 使用DDERequest的Excel巨集

5.2. 把資料從Excel取出

5.2.1. 方法 1 - 在Cogent DataHub設定DDEAdvise loops

5.2.2. 方法 2 - 撰寫使用DDEPoke command的Excel巨集

5.3. 範例

5.4. 網路連接Excel

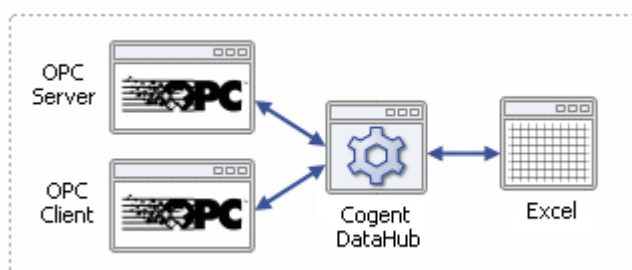
5.5. 使用Ranges

5.5.1. 把Range從Excel取出

5.5.2. 把Range放進Excel

5.5.3. 陣列的範例Excel巨集

您可以使用Cogent DataHub把資料放入Excel，並把資料從Excel寫回到DataHub。



下面各章節將解釋如何把即時資料拖拉進Excel，如何設定Cogent DataHub接收資料，以及如何使用Excel巨集以便在Excel和DataHub之間發送及接收資料。



您可以使用**OPC橋接**，從Excel連結點到OPC server或client，而無需在Excel裡變更或指派點名稱。請參閱**章節 4.4.4, “橋接到Excel”**以了解更多細節。

5.1. 把資料放進Excel

在開始之前，欲檢視任何結果，您必須確保您已把某些類型的資料送進Cogent DataHub。如果您的系統尚未建立，您可以藉由按照**章節 1.2, “以模擬資料測試”**的步驟大綱來建立一個本機資料摘要。

有兩種方法能將Cogent DataHub的資料取出並放進Excel: 藉由建立一個**DDEAdvise loop**來**自動**接收資料，或是使用來自巨集的**DDERequest**命令**讀取**資料。依照您的情況來決定使用哪一種。我們建議您熟悉這兩者。欲了解有關DDE和這些命令的更多資訊，請參閱**章節 16.3.2, “DDE通訊協定”**和**附錄 H, DDE概觀**。

5.1.1. 方法 1 - 使用DDEAdvise來進行拖放功能



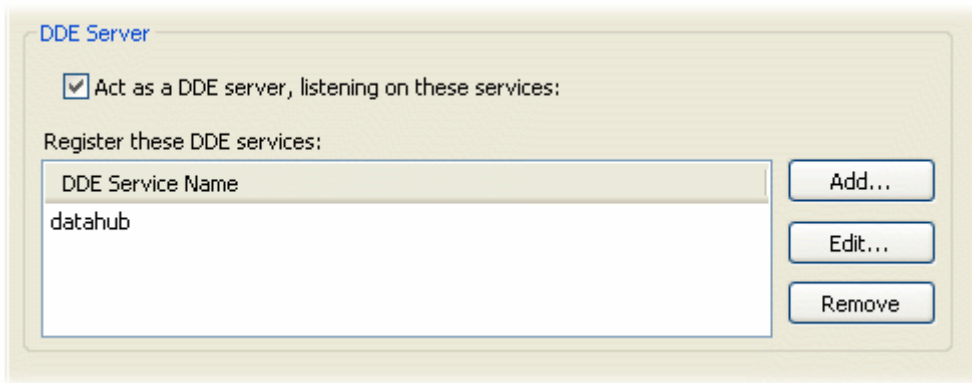
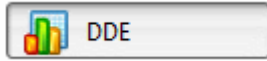
[Click here to watch a video.](#)



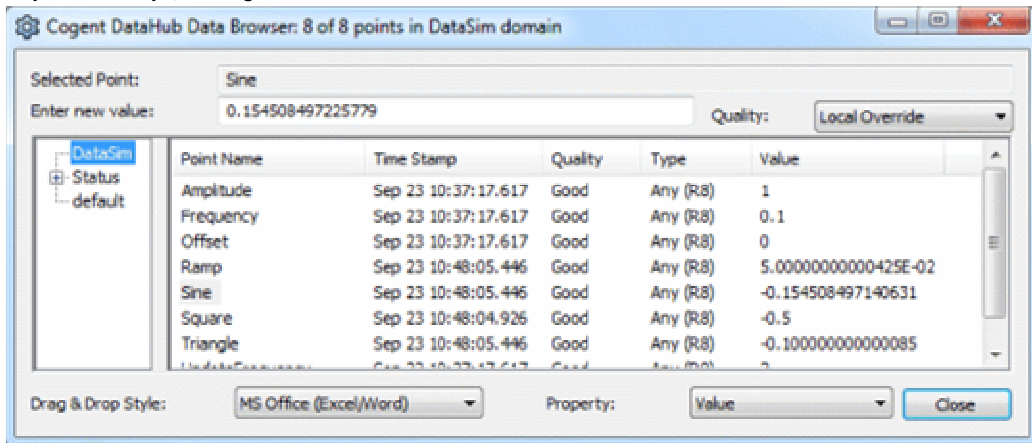
把資料放進Excel的最簡單方法就是，把點名稱從DataHub Data Browser直接拖放進Excel試算表。這會在Excel和DataHub之間自動建立一個**DDEAdvise loop**。**DDEAdvise loops**會自動更新，所以您將可以看到您試算表裡的最新資料。

1. 在Cogent DataHub系統匣圖示上點擊右鍵並選取**Properties**。

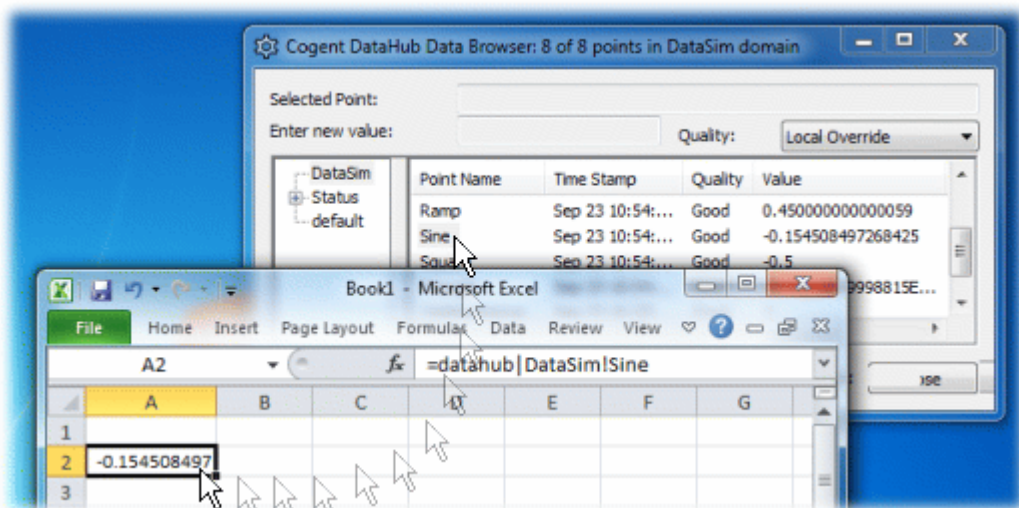
在屬性視窗中，選取**DDE**



3. 確保有勾選**Act as a DDE server**方框，以及dataHub名稱顯示在**DDE Service Name**區域。若沒有，請點擊**Add...**按鈕並新增名稱datahub。
4. 點擊**OK**以關閉屬性視窗。
5. 在桌面的Systems Tray中的Cogent DataHub圖示上點擊右鍵並從快顯功能表裡選取**View Data**以開啓資料瀏覽器。



6. 確保資料瀏覽器下方的**Drag & Drop Style**是設定為**MS-Office (Excel/Word)**。
7. 開啓Excel工作表。
8. 在資料瀏覽器，點擊一個點的標籤並拖曳進Excel工作表。



您應該可以看到資料以在Cogent DataHub更新的相同速度在工作表中更新資料。




您可以藉由使用**Shift+滑鼠左鍵**和**Ctrl+滑鼠左鍵**來選取多個點以進行拖放。



您可以使用**Property**的下拉式清單來拖放時間戳記和點的其他屬性。請參閱**資料瀏覽器**章節裡的**拖與放的樣式以及屬性**小節以獲得更多資訊。

如果您的資料顯示沒有更新，您可能需要在Excel變更您的設定。請參閱**Chapter 21**,

 **Troubleshooting**以獲得更多資訊。

 當您儲存並關閉一個連接到Cogent DataHub的試算表，然後嘗試重新開啓它，您可能會收到一個或更多訊息，這取決於您在Excel的安全設定，或是其他情況。以下是每個訊息的摘要以及做法：

該文件包含巨集，啓用?(This document contains macros. Enable them?)

點擊**Enable Macros**。

該工作簿包含連結，更新?(This workbook contains links. Update them?)

點擊**更新**。如果已經執行DataHub，所有的連結應該會自動更新。如果尚未執行

DataHub，在擁有與DataHub建立之指示loop的每個儲存格裡，會得到一個#REF!項目，接下來的訊息(見下文)可能會出現。

無法存取遠端資料，啓動DataHub?(Remote data not accessible. Start DataHub?)

點擊**NO**。在這個時候最好的辦法就是關閉工作表，手動啓動DataHub，接著重新開啓工作表。這次當您更新試算表時(見上文)，就不會得到任何#REF!項目了。如果此時您並非點擊**NO**，而是點擊**Yes**，DataHub將不會啓動，它反而會建立一個錯誤訊息，且Excel隨後會跟著當掉。

5.1.2. 方法 2 - 使用DDERequest的Excel巨集

有時候，您可能偏好手動把資料讀取至您的試算表，而不是使用**DDEAdvise** loop來連續不斷地接收新的值。您可以藉由在巨集內觸發**DDERequest**命令，讓Excel讀取Cogent DataHub裡您所要求的特定資料點。

在巨集範圍內使用**DDERequest**讓您在Excel讀取新的點值時可以完全地控制，並且讓您在同一時間內讀取多個資料點。欲執行該巨集，您可以很方便的將它連結到一個控制按鈕。這裡說明如何**新增一個控制按鈕**。


5.1.2.1. 建立一個巨集


1. 開啓一個試算表。
2. 在**Tools**功能表裡，選取**Macro**，接著再選取**Macros...**。
3. 在**Macro**對話框的**Macro Name:** 欄位中，輸入**GetInput**，並按一下**Create**按鈕。
4. 在出現的Visual Basic文字項目視窗中，編輯巨集以便讀取如下：

```

'
' GetInput Macro
'
Sub GetInput()
    mychannel = DDEInitiate("datahub", "default")
    Application.Worksheets("Sheet1").Activate
    newval = DDERequest(mychannel, "my_pointname")
    Sheet1.Cells(2, 3) = newval
    DDETerminate mychannel
End Sub

```

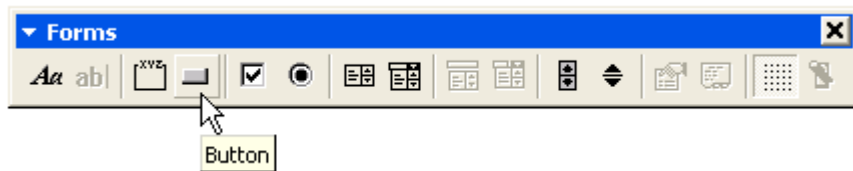
 給**my_pointname**使用您資料點在Cogent DataHub的名稱。

 我們在這個例子中使用儲存格C2。如果您需要使用另一個儲存格，您必須用欲使用之儲存格的列和欄位來置換 (2, 3)。

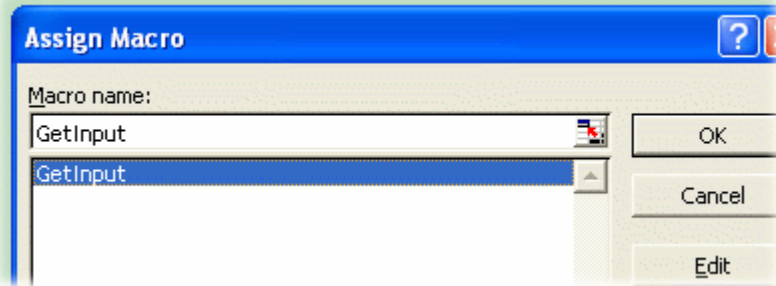
5. 儲存並關閉Visual Basic文字項目視窗。

5.1.2.2. 新增一個控制按鈕

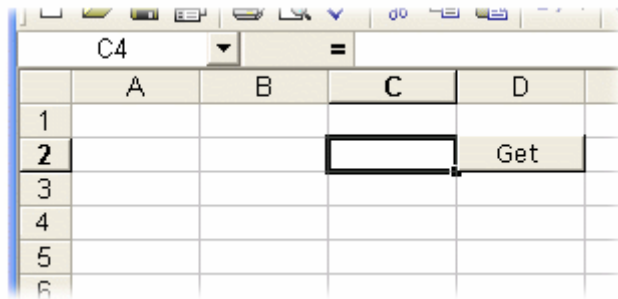
1. 點擊**View**功能表，選取**Toolbars**，然後選取**Forms**，就能啓動Forms工具列。



2. 點擊按鈕圖示，並點擊儲存格D2。(我們在範例中使用該儲存格，您可以自由選擇想要的儲存格。) **Assign Macro** 視窗應會顯示如下。
3. 選取 **GetInput** 接著點選 **OK**。



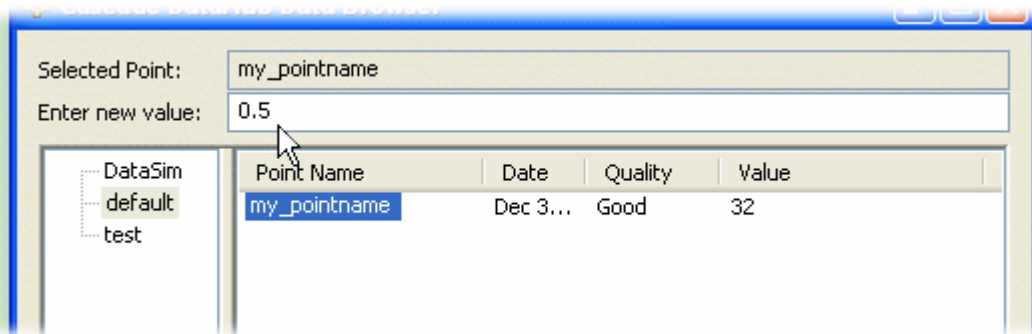
4. 把按鈕的標籤變更為"Get"。



5. 至於其外觀，您可以移動按鈕，使用控制碼調整其大小，並藉由點擊右鍵選取 **Format Control** 來變更文字的大小。
6. 儲存此試算表。

5.1.2.3. 接收資料

1. 現在您已經準備好接收資料。若尚未執行 DataHub 資料瀏覽器，請啟動它，移至預設的 Data Domain，並找尋該點的名稱。
2. 點擊該點並反白它。點的名稱應該會顯示在資料瀏覽器上方的 **Selected Point:** 欄位
3. 在 **Enter new value:** 欄位裡為該點輸入一個新值，然後按 **Enter**。



4. 移至 Excel 並點擊 **Get** 按鈕。每次您點擊此按鈕時，就能看到資料更新。

5.2. 把資料從Excel取出

有兩種方法可以把資料從Excel取出並放進Cogent DataHub:

1. 在DataHub裡的**Configure a DDEAdvise loop** 它指示Excel無論什麼時候，只要一個值變更就自動發送資料到DataHub。

每次當指定的儲存格或**range**變更時，該資料會自動地發送到DataHub。這並不允許發送任何種類的例行性檢查或資料的防護，但在某些情況下，它適合讓Excel自動發出資料。

每次發送一個點(資料item)的資料時，它也會發送所有點的資料。如果您有大量的點，就會佔用您的網路。如果您需要發送大量的儲存格資料，您可以藉由發送含有這些儲存格的儲存格位置來減少這種影響並減低CPU負載。

2. **Write a macro in Excel** 它使用DDEPoke命令把資料從Excel 'push'到DataHub。它在資料發送到DataHub時，可讓您準確定義。

5.2.1.方法 1 - 在Cogent DataHub設定DDEAdvise loops

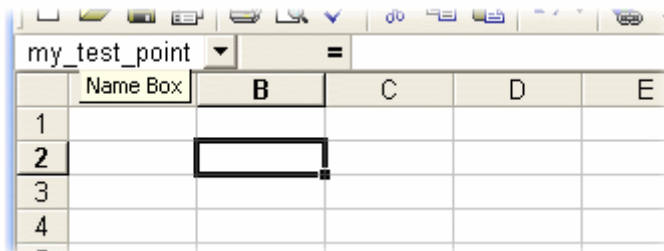


[Click here to watch a video.](#)



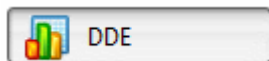
把資料從Excel取出並放到DataHub最快且最簡單的方法，就是在DataHub裡設定一個或多個DDEAdvise loops以便從Excel自動接收資料，它執行成一個DDE server。

1. 開啓一個Excel試算表。
2. 選取一個儲存格或範圍來保留您想要放進DataHub的資料。您將需要參照您的儲存格或範圍的列和欄位號碼，或是名稱。例如，儲存格B2可以稱為R2C2，或是幫他取名稱。

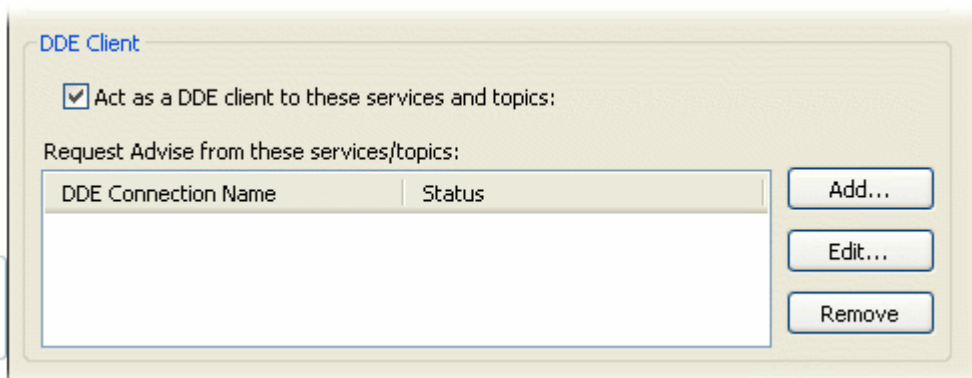


欲幫一個儲存格或範圍命名，選取該儲存格並在工作表第一個欄位上方的方框裡輸入一個獨一無二的名稱。接著儲存工作表。

3. 若尚未執行，請啓動Cogent DataHub，並開啓屬性視窗(在Windows系統匣的DataHub上點擊右鍵並選取**Properties**。)



4. 點擊DDE按鈕。

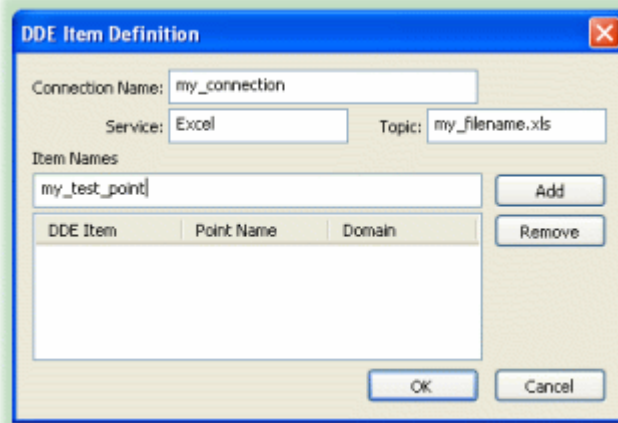


5. 確保勾選**Act as DDE client**方框。

為了達到最佳效能，確保當使用DataHub作為DDE client時，DDE server(在此情況下，即

! Excel)是執行中的狀態。當DDE server不可使用，一個DDE client在嘗試連線時會消耗掉大量的系統資源。

6. 點擊**Add**按鈕。這會開啓**DDE Item Definition**視窗，在此視窗中，您可以新增Excel作為一個新的DDE service。



7. 輸入下列資訊:

- **Connection Name:** 選取一個名稱來識別該連線，它必須是所有DDE連線中獨一無二的名稱。
- **Service:** 輸入**Excel**。
- **Topic:** 輸入工作表檔案的名稱。此名稱與Excel試算表裡，標題列破折號後面的名稱完全相同。因此，若標題列顯示，"Microsoft Excel - Book1"，那麼您的Topic就是Book1。但如果標題列顯示Microsoft Excel - Test.xls，那麼您的Topic就必須是Test.xls。

👉 如果您想要連結到非工作簿中第一頁工作表裡的儲存格或範圍，您需要把檔名放進方括弧內，其次才是工作表名稱。例如，如果您的工作表名稱是Text.xls:

工作簿裡的工作表	Service	輸入的Topic
第一頁工作表	Excel	Test.xls
未命名的工作表(例如Sheet2)	Excel	[Test.xls]Sheet2
已命名的工作表(例如StockData)	Excel	[Test.xls]StockData



- **Item Names:** 輸入列和欄位號碼，或是輸入您在Excel裡所輸入的儲存格或範圍名稱(詳見上面的步驟2)。
8. 點擊**Add**按鈕。接著新增欄位**DDE Item**、**Point Name**和**Data Domain**到與此**DDEAdvise** loop相關聯的items清單裡。您可以在試算表中繼續為其他儲存格新增點，或是點擊**OK**以關閉該對話框。

👉 **DDE Item**和DataHub裡的點相關聯。您可以藉由在名稱上連按兩下接著輸入一個新名稱，以便將**Point Name**和**Data Domain**變更到任何您想要的。

9. 點擊**OK**來關閉DDE Item Definition視窗。新的**DDEAdvise** loop會新增到清單。
10. 點擊**Apply**按鈕使變更生效。一旦您做這個動作，您應該會看到**DDEAdvise** loop連線的**Status**變更至**Connected**。
11. 在系統匣裡的DataHub圖示點擊右鍵，然後選取**View Data**就能開啓資料瀏覽器。
12. 選取**default** Data Domain，向下捲動以便檢視點的名稱。
13. 在Excel中，輸入您在步驟2裡輸入至儲存格或是範圍的號碼，並按**Enter**。您應該可以看到資料在資料瀏覽器裡作更新。

👉 雖然這是從Excel發送資料的簡易辦法，但當您需要傳輸大量的點時，它就不是最有效的方法。每當Excel使用DDEAdvise傳輸一個資料點時，它也會傳輸與任何DDEAdvise loop相關聯的每個點的目前值。

在您需要更新大量儲存格時，我們發現這是傳輸資料類型如Excel範圍的更有效辦法。在您的**DDEAdvise** loop中，定義一個包含您想要傳輸之資料的儲存格範圍。使用Excel範圍會減少電腦的負載，而且它能使您更輕鬆地設定應用程式。

-  另一個減低電腦上負載的的選項是，當傳輸一個大量點時，是撰寫一個使用**DDEPoke**的Excel巨集以便在時間基準內傳輸資料，一秒顯示一次。關於如何在Excel撰寫巨集的資訊如下。
-  當您儲存並關閉一個連接到Cogent DataHub的試算表，然後嘗試重新開啓它，您可能會收到一個或更多訊息，這取決於您在Excel的安全設定，或是其他情況。以下是每個訊息的摘要以及做法：

該文件包含巨集，啓用?(This document contains macros. Enable them?)

點擊**Enable Macros**。

該工作簿包含連結，更新?(This workbook contains links. Update them?)

點擊**更新**。如果已經執行DataHub，所有的連結應該會自動更新。如果尚未執行

DataHub，在擁有與DataHub建立之指示loop的每個儲存格裡，會得到一個#REF!項目，接下來的訊息(見下文)可能會出現。

無法存取遠端資料，啓動DataHub?(Remote data not accessible. Start DataHub?)

點擊**NO**。在這個時候最好的辦法就是關閉工作表，手動啓動DataHub，接著重新開啓工作表。這次當您更新試算表時(見上文)，就不會得到任何#REF!項目了。如果此時您並非點擊**NO**，而是點擊**Yes**，DataHub將不會啓動，它反而會建立一個錯誤訊息，且Excel隨後會跟著當機。

5.2.2. Method 2 - 撰寫使用DDEPoke命令的Excel巨集

撰寫一個Excel巨集也許是從Excel發送資料到Cogent DataHub最靈活且最有效的方法。藉由在Excel巨集裡使用**DDEPoke**命令，當資料被傳送時，您可以完全準確的控制。我們還會說明您要如何撰寫一個Excel巨集，以便在同一時間傳輸多個點(請參閱[附加指標](#)以獲得更多資訊)。

在我們的範例中，我們選擇'新增一個控制按鈕'來執行巨集，但您也可以在時間間隔內執行您的巨集，以便在您所控制的週期內自動更新。


5.2.2.1. 建立一個巨集


1. 開啓試算表。
2. 在Tools功能表中，選取**Macro**，接著選取**Macros...**。
3. 在**Macro**對話框裡的**Macro Name:** 欄位中，輸入名稱**SendOutput**，並按**Create**按鈕。
4. 在出現的Visual Basic文字項目視窗中，編輯巨集以便讀取如下：

```

'
' SendOutput Macro
'
Sub SendOutput()
    mychannel = DDEInitiate("datahub", "default")
    Application.Worksheets("Sheet1").Activate
    Call DDEPoke(mychannel, "my_pointname", Cells(4, 3))
    DDETerminate mychannel
End Sub

```

 給my_pointname使用您資料點在Cogent DataHub的名稱。

 我們在這個例子中使用儲存格C4。如果您需要使用另一個儲存格，您必須用欲使用之儲存格的列和欄位來置換(4, 3)。您也可以替一個範圍命名以便發送多個值作為陣列。

5. 儲存並關閉Visual Basic文字項目視窗。

5.2.2.2. 新增一個控制按鈕

 此說明在[Section 5.1.2.2, "Add a Control Button"](#)講解過。在此簡要的複述。

1. 點擊**View**功能表，接著選取**Toolbars**，再選取**Forms**以啓動**Forms**工具列。
2. 點擊按鈕圖示，接著點擊儲存格**D4**。(也可以選取您想要使用的儲存格。)一個**Assign Macro**視窗應會顯示。
3. 選取**SendOutput**並點擊**OK**。

4. 把按鈕上的標籤變更為"**Send**"。
5. 儲存試算表。

5.2.2.3. 發送資料

1. 現在您準備好發送資料了。若尚未執行，請啟動Cogent DataHub資料瀏覽器。移至預設Data Domain，並尋找點的名稱。
2. 在Excel的儲存格C4(或是您指定給巨集的儲存格或範圍)中輸入一個號碼，並按Enter。
3. 點擊Send按鈕。
4. 您應該能看到資料更新。

5.2.2.4. 額外的指標

- 欲減少大量資料消耗的CPU量，使用**ranges**發送資料的陣列，而不是發送每個儲存格的資料作為個別的點。
- 如果您在DDEPoke命令使用Unicode字元，您應該勾選屬性視窗裡DDE選項中的**Accept non-English characters in Excel strings (slower)**按鈕。

Accept non-English characters in Excel strings (slower)

這會讓Excel正確地發送Unicode字元的字串，雖然會比數字資料還要慢些。欲了解更多資訊，請參閱[章節 18.6](#)，**"DDE"裡面的Excel裡的非英文字元**。

- **DDEInitiate**和**DDETerminate**命令 用來開啓和關閉應用程式之間的DDE連結，它也會消耗許多CPU。當需要在頻繁的時間間隔內發送變數時，在工作階段的開啓DDE channel，並在完成之後關閉的這個方法會更有效率。這裡有兩種建議：

1. 在**DDEInitiate**和**DDETerminate**的單一命令集中發送多個點。例如：

```

' Cascade Multiple Writeback macro
'
Sub Cascade_Writeback_Many()
    mychannel = DDEInitiate("datahub", "default")
    Application.Worksheets("variables").Activate
    DDEPoke(mychannel, "pointname1", Cells(1,2))
    DDEPoke(mychannel, "pointname2", Cells(2,2))
    DDEPoke(mychannel, "pointname3", Cells(3,2))
    DDEPoke(mychannel, "pointname4", Cells(4,2))
    DDEPoke(mychannel, "pointname5", Cells(5,2))
    DDEPoke(mychannel, "pointname6", Cells(6,2))
    DDETerminate mychannel
End Sub

```


在這個範例中工作表命名為variables，包含六個我們想要發送至Cogent DataHub的變數(pointname1至pointname6)。**DDEInitiate**命令開啓channel，接著在連結關閉之前把所有六個變數發送到DataHub。

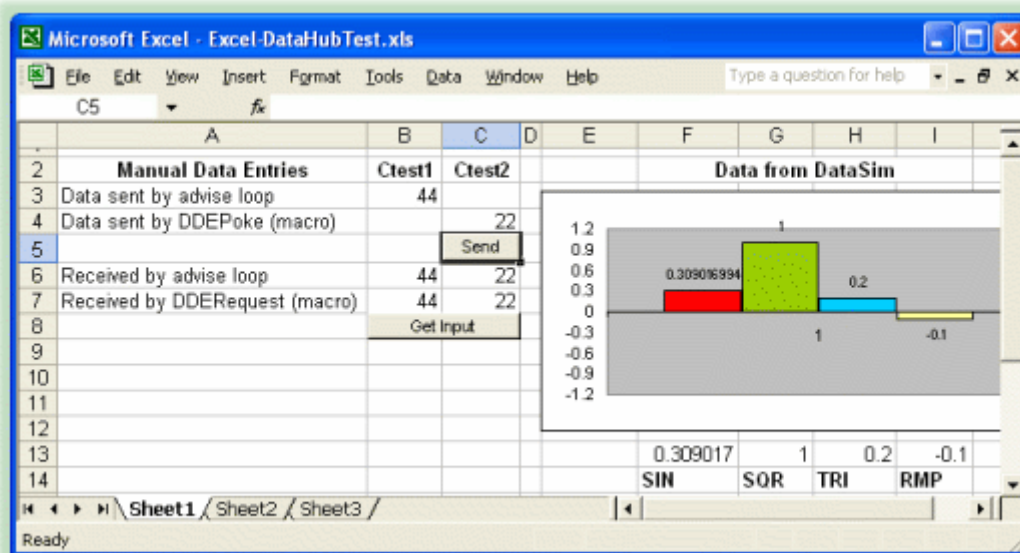
2. 為工作表建立單獨的'open'和'close'巨集，並將**DDEInitiate**和**DDETerminate**命令放進這些巨集內。這會讓連到DataHub通訊在工作表開啓時保持開放。唯一的缺點是您的資料傳輸會被中斷(見下文)。
- 如果您需要從Excel連續不斷地發送資料到Cogent DataHub，您或許會遇到使用**DDEInitiate**和**DDEPoke**的困難。當您使用**DDEInitiate**陳述式開啓一個DDE channel，並使用多個**DDEPoke**陳述式追蹤它時，DDE channel也許會在一段時間後失敗。基於這個原因，如果您需要延長DDE channel保持開放的時間，我們建議您嘗試在巨集內處理DDE錯誤。

版權所有 © 1995-2011 by Cogent Real-Time Systems, Inc.

5.3. 範例

下面是發送資料到Cogent DataHub的兩種工作表範例: DataHub執行成爲DDE client(使用advise loop), 以及使用巨集(用DDEPoke和DDERequest)。工作表也會以兩種方法從DataHub接收相同資料回來。欲了解有關DDE和這些命令的更多資訊, 請參閱附錄 H, *DDE概觀*。

 此工作表, Excel-DataHubTest.xls, 也包含在您的Cogent DataHub發佈裡。



有關手動資料項目儲存格的說明:

B3 該儲存格命名爲: Ctest1。當您使用已命名的儲存格發送資料到執行成爲DDE client的Cogent DataHub時, 每次更新任何儲存格, 其他所有以此方式建立的儲存格也會發送它們的資料到Cogent DataHub。

C4 讓一個巨集與之關聯, 如下所示。當您使用巨集發送資料到Cogent DataHub時, 在每次您發送更新值, 只有該儲存格的資料會被發送到Cogent DataHub。

巨集是:

```
Sub SendOutput()
  mychannel = DDEInitiate("datahub", "default")
  Application.Worksheets("Sheet1").Activate
  Call DDEPoke(mychannel, "Ctest2", Cells(4, 3))
  DDETerminate mychannel
End Sub
```

C5 Send 按鈕啓動巨集與儲存格**C4**相關聯。公式是: Excel-DataHubTest.xls!SendOutput

B6 儲存格公式是: =datahub|default!Ctest1

C6 儲存格公式是: =datahub|default!Ctest2

B7和**C7** 讓一個巨集與之關聯, 如下所示。當您使用一個巨集從Cogent DataHub接收資料時, 資料只會更新一次。

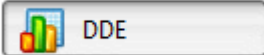
巨集是:

```
Sub GetInput()
  mychannel = DDEInitiate("datahub", "default")
  Application.Worksheets("Sheet1").Activate
  newval1 = DDERequest(mychannel, "Ctest1")
End Sub
```

```
newval2 = DDERequest(mychannel, "Ctest2")  
Sheet1.Cells(7, 2) = newval1  
Sheet1.Cells(7, 3) = newval2  
DDETerminate mychannel  
End Sub
```

B8和C8 Get按鈕啟動巨集與儲存格**B7**和**C7**相關連。公式是: Excel-DataHubTest.xls!GetInput。

試算表的**Data from DataSim**部分從四個**DataSim**點接收資料(SIN、SQR、TRI和RMP)，並在圖表中顯示它們的值。

Cogent DataHub DDE屬性  在該範例中設定如下:

- **Act as DDE client to these services and topics:** 已勾選。
 - **Connection Name:** ExcelTest
 - **Service:** Excel
 - **Topic:** Excel-DataHubTest.xls
 - **DDE Item:** Ctest1
 - **Point Name:** Ctest1
 - **Data Domain:** default
- **Act as DDE server listening on these services:** 已勾選。
 - **DDE Service Name:** datahub

5.4. 網路連接Excel

您可以使用Cogent DataHub，利用DataHub鏡像功能以便即時網路連接Excel。**Mirroring**是兩個或更多Cogent DataHubs連結透過網路或Internet來維護相同資料集的方法。

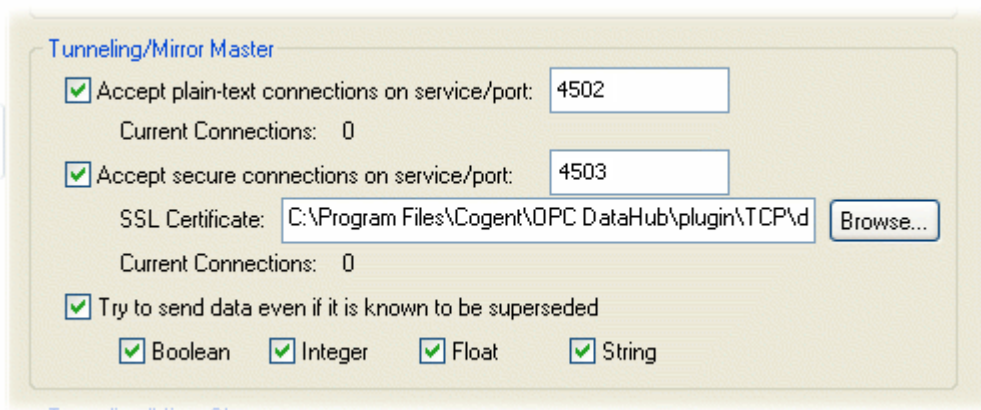
 Mirroring和tunnelling是一樣的，如**章節 2**，**OPC Tunnelling**所述。鏡像也可用來連接執行在Linux的Cascade DataHub。請參閱用於Linux和QNX的Cascade DataHub手冊的鏡像資料章節以了解更多細節。

欲網路連接Excel，在每個節點上您都需要將Excel連接到Cogent DataHub。接著鏡像連接是在每個DataHub之間設定。對於每個鏡像連接，您必須指定一DataHub為master，其它的為slave。這會決定由哪一邊啟動溝通。一旦建立通訊，資料就是完全相同的。通常，我們建議由server上的DataHub或是最不可能關機的機器執行成為master，而slave在client機器上。在hub-and-spoke連線排列方式裡，DataHub可以是連接到多個masters的slave，以便在單一DataHub裡連接所有資料。

設定DataHub作為tunnel/mirror master

1. 在Cogent DataHub系統匣圖示上點擊右鍵，並選取**Properties**。

2. 在屬性視窗中，選取**Tunnel/Mirror** 。



3. 在**Tunnelling Master**區段裡，您可以設定純文字或安全tunnelling。確保至少勾選其中一個。如果您想要變更任何其它的預設，請參閱**章節 18.4, "Tunnel/Mirror"**以了解更多資訊。

 欲進行傳輸量最佳化，取消勾選**Try to send data even if it is known to be superseded**選項。這會讓DataHub在client通知原始變更之前丟棄已經變更的失效點值。最新的值會一直被傳送。

4. 點擊**OK**以關閉屬性視窗。

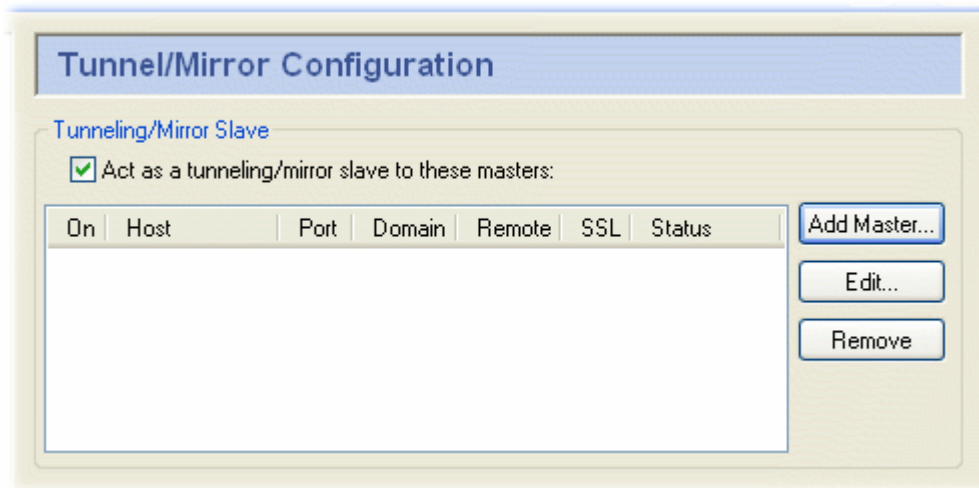
您現在可以設定slave DataHub。

設定Cogent DataHub作為tunnel/mirror slave

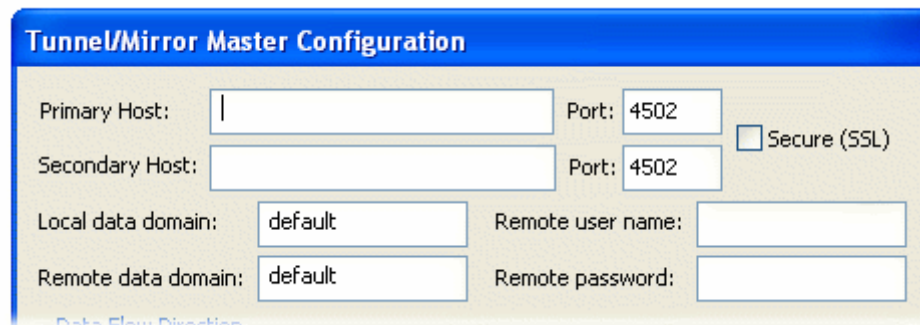
slave DataHub行為完全就像master DataHub，除了slave會在一開始建立tunnelling和在網路中斷後重新連線之外。

1. 在桌面的System Tray中的Cogent DataHub圖示點擊右鍵並選取 **Properties**。

2. 在屬性視窗中，選取**Tunnel/Mirror** 。



3. 勾選**Act as a tunnelling/mirror slave to these masters**方框
4. 點擊**Add Master...** 來指定該slave的master。 **Tunnel/Mirror Master Configuration**會開啓如下:



5. 輸入下列資訊:
 - **Primary Host:** 執行tunnelling master DataHub的電腦名稱或IP位址。
 - **Port:** 該連接埠的連接埠號碼或service名稱。您應該使用預設的連接埠號碼(4502)除非您已經變更master DataHub的輸入項目。
 - **Secondary Host:** 讓您可以選擇擁有一個備用主機和service/port號碼。在啓動或是連線中斷之後，DataHub會先搜尋主要主機，接著才是次要主機，在主要和次要之間交替直到建立連線。如果沒有指定次要主機，就只會在主要主機上嘗試連線。
 - **Local data domain:** 您計畫從中接收資料的Data Domain。
 - **Remote data domain:** 您計畫從中接收資料的master DataHub Data Domain。點的名稱會從遠端Data Domain中(在master DataHub)對應到本機Data Domain(在該DataHub)，反之亦然。



除非您有好的理由需要讓這些不同，否則，爲了簡單化，我們建議兩個DataHub都使用相同的Data Domain。




以下列出的是在Cogent的server上執行的DataHub，您可以連接它以進行測試。下面是您需要輸入的參數:

- **Primary Host:** developers.cogentrts.com
- **Port:** 4502
- **Local data domain:** test
- **Remote data domain:** test


6. 您現在有鏡像連接方式的多種選擇。

- a. **Data Flow Direction:** 讓您決定資料流的方式。slave和master之間的預設值是雙向資料流，但您可以藉由選取個別選項來有效地建立一個唯讀或唯寫的連線。

 欲最佳化傳輸量，勾選**Read-only: Receive data from the Master, but do not send**選項。只有當您真的想要一個唯讀連線時才勾選該選項。如果您不要求讀寫雙向的存取，唯讀的tunnel速度會更快。

- b. **When the connection is initiated:** 當slave第一次連接到master時，決定點的值如何被指定。這有三種可能性: slave取得來自master的所有值、slave發送其所有值到master、或master和slave同步它們的資料集、逐一地，根據每個點的最新值(預設)。

- c. **When the connection is lost:** 決定"未連線"時，要在哪裡顯示資料品質—在master、在slave、或是兩者皆非。

 如果您已經將**When the connection is initiated**設定為**Synchronize based on time stamp**(見上文)，那麼這個選項必須設定為**Do not modify the data quality here or on the Master**以便取得正確的資料同步。

- d. **Connection Properties** 有下列選項:

- **Replace incoming timestamp...** 該選項讓您在時間戳記裡使用本機時間。如果資料來源無法建立時間戳記，或是您不信任資料來源的時間，那這個選項就非常有用。
- **Transmit point changes in binary** 該選項提供使用者x86 CPUs，這是能加快資料傳輸率的方法。選取該選項可以提升最大傳輸量至高達50%。

 欲了解更多資訊，請參閱**章節 17.1**，**“二進位模式的Tunnel/Mirror (TCP) 連接”**。

- **Target is a Cogent Embedded Toolkit server** 該選項讓此slave連接到內嵌工具組server，而不是連接到另一DataHub。
- **Heartbeat** 該選項使用在此指定的每個milliseconds秒數來發送一個活動訊號訊息到master，藉此驗證已連線。
- **Timeout** 該選項指定活動訊息的逾時期間。如果slave DataHub在逾時之內沒有從master接收到回應，它會中斷該連線。您必須把逾時時間設定為至少是活動訊息時間的

兩倍。



欲將此減緩網路的設定最佳化，請參閱**章節 17.2**，“**慢速網路的 Tunnel/Mirror (TCP)連接**”。

- **Retry** 在嘗試重新連接到已中斷的連線之前，指定一個等候的milliseconds秒數量。
7. 點擊**OK**以關閉**Tunnel/Mirror Master**視窗。屬性視窗裡**Tunnelling Slave**表格欄位現在應該已填滿。
 8. 點擊屬性視窗的**Apply**按鈕。如果正在執行master DataHub，該DataHub會建立tunnelling連線，**Status**狀態也會顯示**Connected**。您可以用**Data Browser**檢視資料，或是使用**Connection Viewer**來檢視連線。

如果master Cogent DataHub有被正確地設定，您應該會看到Data Domain的所有master DataHub資料。

版權所有 © 1995-2011 by Cogent Real-Time Systems, Inc.

5.5. 使用Ranges

Cogent DataHub可以發送和接收包含在Excel試算表裡整個範圍的資料。這個資料會被視為一個array，而一個二維範圍的儲存格作為列和資料行。此陣列可以如所需的一樣大小(在點的大小限制範圍之內)，或是小至一個儲存格大小 – 至少一行或是一個資料行大小。

資料格式

Excel傳輸陣列資料作為tab-and-newline的值分隔符號文字字串。每個列裡面的值會被標籤隔開，而且每個列用換行字元分隔。該字串不包含任何有關試算表裡陣列來源範圍的資訊。

5.5.1. 把Range從Excel取出

有兩種從Excel傳輸到Cogent DataHub的方法，一是傳輸一個儲存格位置，另一為傳輸陣列資料。這些完全符合用於個別點的資料機制: **DDEPoke**和**DDEAdvise**。

與巨集一起使用DDEPoke

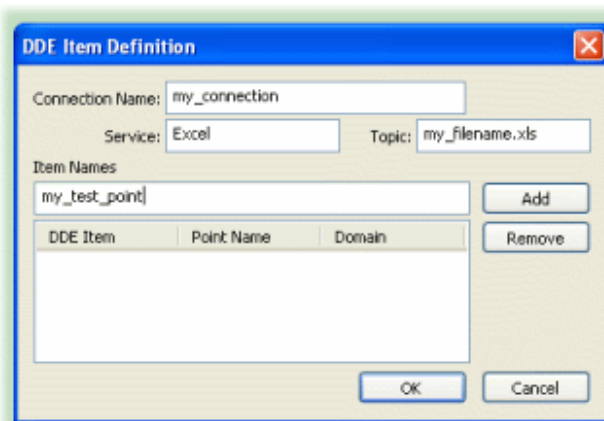
DDEPoke命令可以由Excel發行以便在Excel內根據觸發程序發送資料至Cogent DataHub。為了使它運作，Cogent DataHub需要設定以**執行成為DDE server**並至少登錄一個service名稱。接著，Excel巨集可以發送一個DDEPoke到該service，同時包含Cogent DataHub資料網域名稱(DDE topic)、一個點名稱(DDE item)和一個值。如果值是在Range類型內，那麼Excel會自動格式化此值作為一個tab-and-newline的分隔字串。

範例: 請參閱下面**Excel macro coding examples**裡的PutData函式的定義。

使用DDE Advise Loop

當使用**DDE advise loop**從Excel發送資料到Cogent DataHub時，Excel執行成為DDE server，DataHub執行成為client。欲建立一個advise loop:

1. 開啟Cogent DataHub屬性視窗(在Windows系統匣的DataHub圖示上點擊右鍵，然後選取**Properties**)。
2. 點擊**DDE**按鈕。
3. 確保已勾選**Act as DDE client**方框。
4. 點擊**Add**按鈕。這會開啓**DDE Item Definition**視窗。



5. 輸入下列資訊:
 - **Connection Name:** 選取一個名稱以識別該連接，它必須是所有DDE連接中唯一的名稱。
 - **Service:** 輸入**Excel**(是否大小寫不重要)。
 - **Topic:** 輸入工作表檔案的名稱，包含.xls副檔名，例如: *my_filename.xls*。
 - **Item Names:** 這會在Excel儲存格和儲存格位址之間建立一個對應，以及Cogent DataHub點名稱。您也可以指定一個單一儲存格，在r1c1格式中指定一個單一儲存格，在r1c1:r2c2格式中指定一個儲存格位址、指定一個儲存格名稱或是指定一個儲存格位址名稱作為DDE Item名稱。例如:

- r2c5 - accesses the cell E2 (second row, fifth column)
- r3c3:r5c9 - accesses the range C3:I5
- MyRange - accesses the cell or range that is named MyRange

6. 點擊Add按鈕。DDE Item、Point Name和Data Domain欄位會自動的填入一些值。



勾選Point Name和Data Domain欄位的名稱。如果這其中沒有您所需要的，請連按兩下滑鼠以選取，並進行變更。

7. 點擊OK以關閉DDE Item Definition視窗。屬性視窗裡的DDE Connection Name和Status欄位應該也會被填滿。
8. 點擊OK以關閉屬性視窗。
9. 在試算表裡已定義的儲存格位置中輸入一些值。您可以看到資料瀏覽器裡的陣列相對的變更。

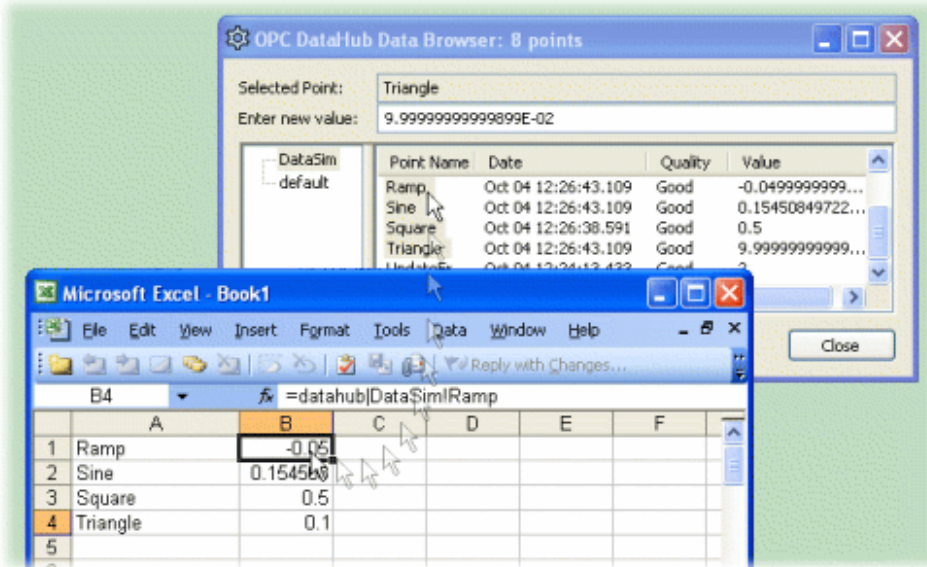
5.5.2. 把Range放進Excel

有兩種方法可以將資料拖放進Excel以建立一個range，使用DDE advise loops，或是您也可以使用DDE Request和巨集。

將一個點的群組拖放進Excel

下列說明如何在DataHub裡收集點的群組，並拖曳進Excel，在Excel裡每個資料點都佔用一個唯一的儲存格。

1. 在Cogent DataHub和DataSim執行時，開啓資料瀏覽器。
2. 在資料瀏覽器上選取點的群組。
3. 將點的名稱拖曳至Excel。



您應該會在儲存格裡進行的資料更新。



您可以使用Property下拉式清單來拖放點的名稱、時間戳記以及點的其他屬性。請參閱Data Browser章節裡的拖與放的樣式和屬性以了解更多資訊。

將一陣列拖放進Excel

下面範例示範如何在DataHub陣列裡取出一個點值，並且讓陣列裡的每個值佔用Excel裡每個獨一無二的儲存格。

為了示範，我們首先將結合上面的兩個程序以便在DataHub中建立一個陣列。

製作一個陣列

1. 選取Excel中的範圍，如在上圖將一個點的群組拖放進Excel中建立的範圍，並在左上角的名稱方框中輸入FirstRange。

	A	B	C
1	Ramp	-0.15	
2	Sine	0.404508	
3	Square	0.5	
4	Triangle	0.3	
5			

- 在Cogent DataHub屬性視窗中，選取DDE選項並確保已勾選Act as DDE client方框。接著點擊Add按鈕。
- 在DDE Item Definition視窗中輸入下面資訊：
 - **Connection Name:** 輸入Ranges。
 - **Service:** 輸入Excel。
 - **Topic:** 輸入Book1，或是輸入您的工作表檔案，包含.xls副檔名。
 - **Item Names:** 輸入FirstRange。
- 點擊Add按鈕。欄位DDE Item和Point Name應該為FirstRange，而且Data Domain應該為default。
- 點擊OK以關閉DDE Item Definition視窗，也在屬性視窗中點擊OK以關閉視窗。
- 開啓資料瀏覽器，並移至default Data Domain。您應該會看到點FirstRange，其值如下：

Point Name	Date	Quality	Value
FirstRange	Oct ...	Good	Ramp□0.35□Sine□-0.404508497□Square□0.5□Tri...

單一方框將值分隔成一排，而兩個方框會分隔行列。該陣列已經做好放進Excel的準備了。

拖放陣列

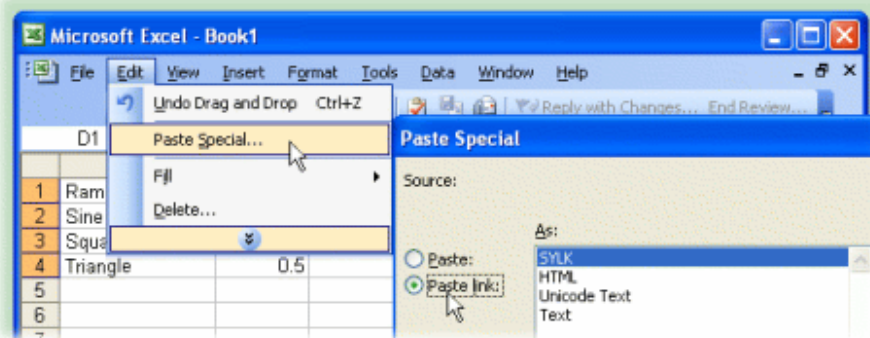
爲了方便起見，我們將只把相同陣列放回Excel。

- 點擊FirstRange點名稱，並拖曳至Excel，然後置放在儲存格D1中。

B	C	D	E	F	G
-0.35		Use Edit->Paste Special->Paste Link			
404508		0	0		
0.5		0	0		
0.3		0	0		

該值不會馬上更新，因爲您必須告知Excel如何在連結中貼上。

- 移至Edit功能表並選取Paste Special。



- 選取Paste link然後點擊OK。儲存格應該會填進正確、更新的資料。

在Excel使用DDE Request

如果您在Excel建立巨集以便從Cogent DataHub讀取資料，您可以使用DDERequest函式呼叫。

這會傳回一個能夠直接寫入試算表裡任何儲存格位置的陣列型態值。如果陣列資料的維度比即將寫入的範圍還大時，那麼陣列裡多餘的資料會被捨棄。如果陣列資料比目標範圍還要小，那麼該範圍裡多餘的儲存格就會填入重複的資料。請參閱下列自動判定的Excel巨集，請參閱下列自動判定目標範圍的Excel巨集，以確保所有的陣列資料已經輸入試算表裡，而沒有重複資料。

5.5.3. 陣列的Excel範例巨集

下面的巨集代表整個巨集設定，它能夠讀取和寫入Cogent DataHub裡單一陣列點的簡單測試試算表。GetData和PutData這兩個函式可以被附加到試算表裡的按鈕以便簡單測試。PutData副程式包含兩個來源範圍的替代表示法，其中一個已經在巨集內添加註解。

```
Sub GetDataArray(Channel As Integer, SheetName As String, DataPoint As String, _
    StartRow As Integer, StartCol As Integer)
    Dim NRows As Integer, NCols As Integer

    ' This sub performs a DDERequest for DataPoint in the DDE Channel and reads in a tab
    ' delimited array with carriage returns at the end of each line. It then fills a range
    ' of cells with the data. The native format for Excel data is tab delimited text with
    ' carriage return at the end of each row of data. If we assign this type of data to a
    ' range of cells using the FormulaArray function, Excel automatically parses the data
    ' and fills it into the specified range. The real trick here is to ensure that the
    ' range is the same size as the incoming data, so we do not have to know the size
    ' a priori.

    dataArray = DDERequest(chan, DataPoint) ' request DataPoint from Channel

    ' find the upper row and column bounds for the variant array

    If StartCol = 0 Then StartCol = 1 ' Starting column where data will go in our sheet
    If StartRow = 0 Then StartRow = 1 ' set the starting row
    NCols = 1 ' set default number of columns to 1
    On Error Resume Next ' ignore errors (error occurs if array has
    ' one dimension)

    ' get upper bound of the array columns
    ' the following line will generate an error if the array is only a one dimensional arr
    ' We just skip this, and use the default 1
    NCols = UBound(dataArray, 2)

    On Error GoTo 0 ' allow errors
    NRows = UBound(dataArray, 1) ' get upper bound of array y dimension

    NRows = NRows + StartRow - 1 ' add offset from StartRow - this is the ending row
    NCols = NCols + StartCol - 1 ' add offset from StartCol - this is the ending column

    ' the following line fills up the cells in the range starting in "StartCol:StartRow" t
    ' "NRows:NCols" with the data from the variant array
    Sheets(SheetName).Range(Cells(StartRow, StartCol), Cells(NRows, NCols)) = dataArray
End Sub

Sub PutdataArray(Channel As Integer, SheetName As String, DataPoint As String, _
    StartRow As Integer, StartCol As Integer, NRows As Integer, _
    NCols As Integer)
    DDEPoke Channel, DataPoint, Sheets(SheetName).Range(Cells(StartRow, StartCol), _
    Cells(StartRow + NRows - 1, StartCol + NCols - 1))
End Sub

Sub PutDataRange(Channel As Integer, DataPoint As String, DataRange As Range)
    DDEPoke Channel, DataPoint, DataRange
End Sub

Sub GetData()
    '
    ' This is a test function assigned to a button. It reads a test point into
    ' an arbitrarily sized matrix starting at A10
    '
    Dim chan As Integer
    chan = DDEInitiate("datahub", "default")
    GetDataArray chan, "Sheet1", "TestArray", 10, 1
    DDETerminate (chan)
End Sub
```

```
Sub PutData()  
,  
,  
, This is a test function assigned to a button. It writes a 3 row x 5 column  
, area of Sheet1 into a single data point in the DataHub. You can use either  
, PutDataArray or PutDataRange, depending on how you wish to specify the range.  
,  
,  
Dim chan As Integer  
chan = DDEInitiate("datahub", "default")  
'PutDataArray chan, "Sheet1", "TestArray", 1, 1, 3, 5  
PutDataRange chan, "TestArray", Sheets("Sheet1").Range("A1:E3")  
DDETerminate (chan)  
End Sub
```

版權所有 © 1995-2011 by Cogent Real-Time Systems, Inc.