

章節 10. Email和SMS

目錄

10.1. 簡介

10.1.1. 運作方式

10.1.2. 有關SMS簡訊的注意事項

10.2. 設定Mail Server

10.3. 發送測試訊息

10.4. 定義Email訊息

10.5. 指派一個Trigger

10.6. 設定觸發條件

10.7. 已設定的動作

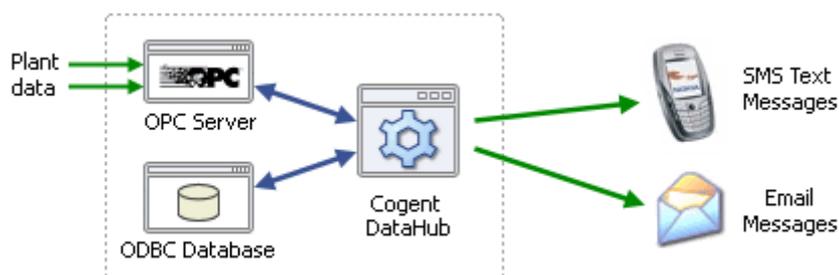
10.8. HTML訊息範例

10.8.1. 包含內嵌資料點的HTML訊息

10.8.2. 包含使用程式碼建立的Table之HTML訊息

10.1. 簡介

Cogent DataHub讓您發送email和SMS簡訊，這是由DataHub事件觸發的，例如點值變更，或者是由計時器觸發的。此email和簡訊可以是純文字格式或是HTML格式，而且它們可以包含DataHub裡面任何資料點的現值。



隨著DataHub的這項功能，您可以：

- 當警報事件觸發時，發送SMS簡訊到手機。
- 可以設計每日總結報表，並在每天早上傳送到主管的email信箱。
- 定期地提供主管人員生產目標更新的email。
- Email的內容可包含來自OPC servers、ODBC資料庫和其它資料來源的資料。
- 減少因為人員手寫生產報表所導致的錯誤。
- 使用DataHub收集重要的報表資訊，將其製成Excel試算表格式，並將此檔以email方式寄送至相關人員的信箱以供檢閱。

10.1.1. 如何運作

DataHub有一個內建的郵寄程式。DataHub指令碼告知程式應該發送哪些訊息，發送到哪裡，以及何時發送。一個典型的郵寄指令碼包含以純文字格式發送email、使用ASP的HTML格式發送email的說明或者是兩者的說明。該範例指令碼，[MailTest.g](#)包含兩種方法的範例。您可以執行該指令碼以測試郵件程式，接著使用下面範例以發送您的個人訊息。

該章節裡的小節將告訴您如何：

1. 設定郵件程式。
2. 發送一則測試訊息。
3. 發送您的個人訊息。
4. 建立HTML訊息。

10.1.2. 有關SMS簡訊的備註

從DataHub發送SMS文字訊息是藉由發送一個小型的純文字email到適當的SMS閘道器的email位址而完成的。大多數的手機服務供應商在新的訂閱上提供email和簡訊的選項。

例如，如果您在加拿大使用Bell Mobility訂閱，且您的新號碼是(416) 123 4567，那麼該手機的email位址是：

4161234567@bellmobility.ca

SMS簡訊位址略有不同：

4161234567@txt.bellmobility.ca

正常情況下，SMS簡訊是從一隻手機發送到另一隻手機，這全都是在手機本身的網路中處理完成。使用SMS簡訊位址，您可以發送定期的email，它會被SMS閘道器(範例中為txt.bellmobility.ca)轉換成一則簡訊並發送到您的手機。當從email轉換至簡訊時可能會有短暫的延遲，但訊息通常會在一分鐘之內送達。

請向您的手機服務供應商查看您手機號碼的SMS簡訊位址。

10.2. 設定Mail Server

在您可以從Cogent DataHub發送email之前，您將需要設定DataHub郵寄server程式如下：

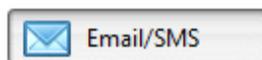


[Click here to watch a video.](#)



1. DataHub執行時，在DataHub System Tray圖示上點擊右鍵並選擇**Properties**。

2. 在Properties視窗中，選取**Email/SMS**



Email and SMS Configuration

[How do I use dynamic Email and SMS messaging?](#)

Configure Outgoing Mail Server

SMTP Server:

Port (default 25):

Sender Email:

User name:

Password:

輸入您想要使用在發送email的資訊。這可以與您的email client程式清單中的SMTP server相同。

SMTP Server:

SMTP server的名稱。

Port:

SMTP連接埠名稱 (通常是通訊埠25)。

Sender Email:

寄件者的email位址。這會顯示在email的From欄位。該位址可以是這兩種類型之一：

- **username@datadomain.com**在email讀取器(client)上將顯示為username@datadomain.com。
- **User Name <username@datadomain.com>**在email讀取器(client)上將顯示為User Name。

User name:

您使用來存取該SMTP帳戶的登錄名稱。

Password:

適用的密碼。

3. 在**Security**區塊中：

Security

Never attempt to connect securely via SSL

Always use SSL (fail if unavailable)

Automatically select most secure connection

Accept invalid or untrusted certificates

選取SSL的三個選項之一，並指定是否想要接受無效或不信任的安全憑證。

4. 點擊**Apply**或**OK**按鈕以送出您的輸入項目。

現在可以使用DataHub郵件程式了。如果您尚未進行此動作，我們建議**發送一則測試訊息**，如下一節中所述。

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

10.3. 發送一則測試訊息

一旦您設定完mail server您可以設定並發送一則測試email，方法如下：

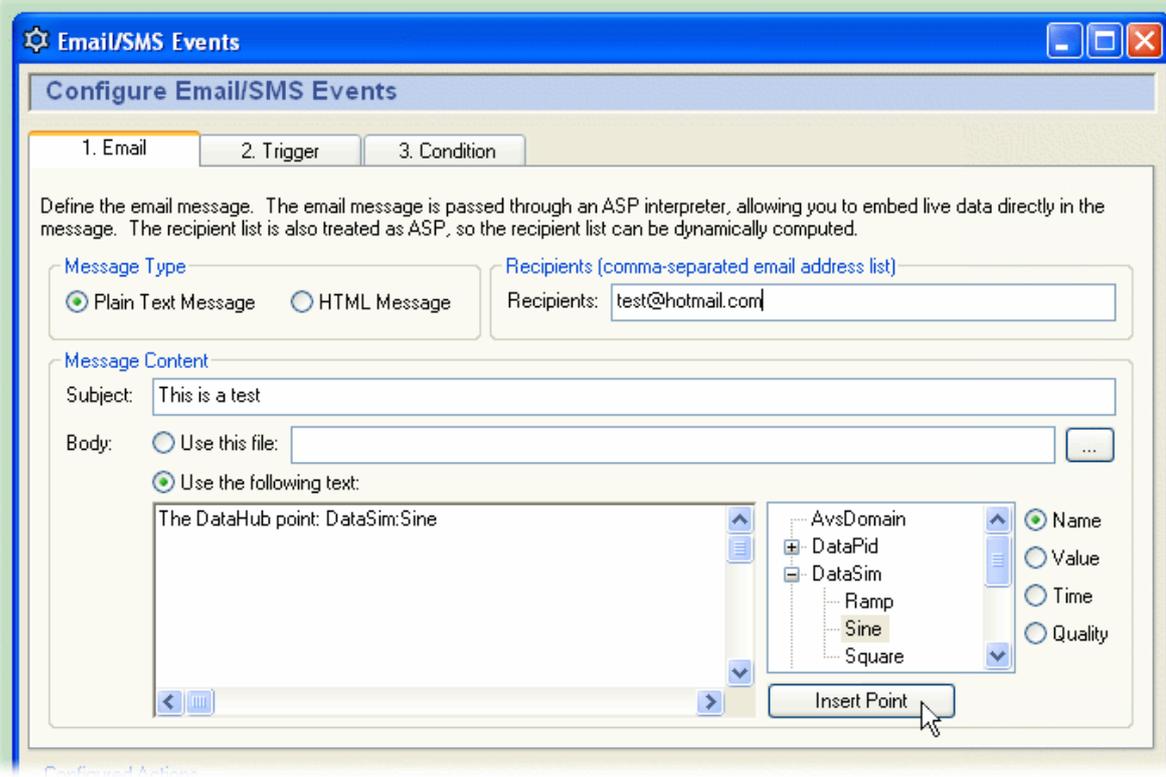


開啓Email/SMS Events視窗

1. 在Cogent DataHub Properties視窗中，選取 **Email/SMS** 。
2. 在**Configure Email and SMS Events**區塊中按下**Configure**按鈕。



這會開啓Email/SMS Events視窗：



定義Email訊息

1. 選取頁籤**1. Email**。
2. 在**Message Type**選項上選取**Plain Text Message**。
3. 在**Recipients**方框中輸入收件人的email位址。您可以輸入多個位址並使用逗號分隔。
4. 在**Subject**方框中輸入主旨。
5. 至於**Body**，選取**Use the following text:**。
6. 若尚未執行，請啓動DataSim程式，並確保它連接到DataHub。
7. 在右側的點選取器清單中，展開DataSim Data Domain並選取點名稱Sine。
8. 點擊點選取器清單右側的**Name**按鈕。
9. 在文字項目欄位，輸入如下：

The DataHub point

10. 點擊**Insert Point**按鈕。您的文字現在應會顯示如下：

```
The DataHub point DataSim:Sine
```

11. 按下**Enter**並繼續輸入:

```
The DataHub point DataSim:Sine
had a value of:
```

12. 點擊**Value**按鈕接著點擊**Insert Point**按鈕，您的文字現在應會顯示如下:

```
The DataHub point DataSim:Sine
had a value of: <%= $DataSim:Sine %>
```

13. 按下**Enter**並繼續輸入:

```
The DataHub point DataSim:Sine
had a value of: <%= $DataSim:Sine %>
at the time:
```

14. 點擊**Time**按鈕，接著點擊**Insert Point**按鈕。您的文字現在應會顯示如下:

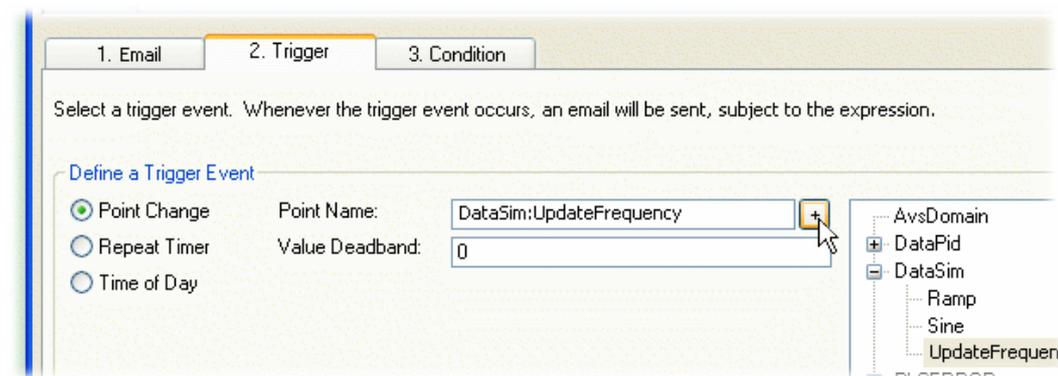
```
The DataHub point DataSim:Sine
had a value of: <%= $DataSim:Sine %>
at the time: <%= PointTimeString( # $DataSim:Sine % ) >
```

訊息已經準備完成。現在您可以指派一個Trigger以及設定條件。

指派一個Trigger

在這個例子中，我們將觸發DataSim:UpdateFrequency點的動作。

1. 選取頁籤**2. Trigger**。
2. 在點選取器中，展開DataSim Data Domain，接著選取點UpdateFrequency。
3. 點擊**Point Name**欄位右方的+按鈕。點名稱DataSim:UpdateFrequency應會填滿。

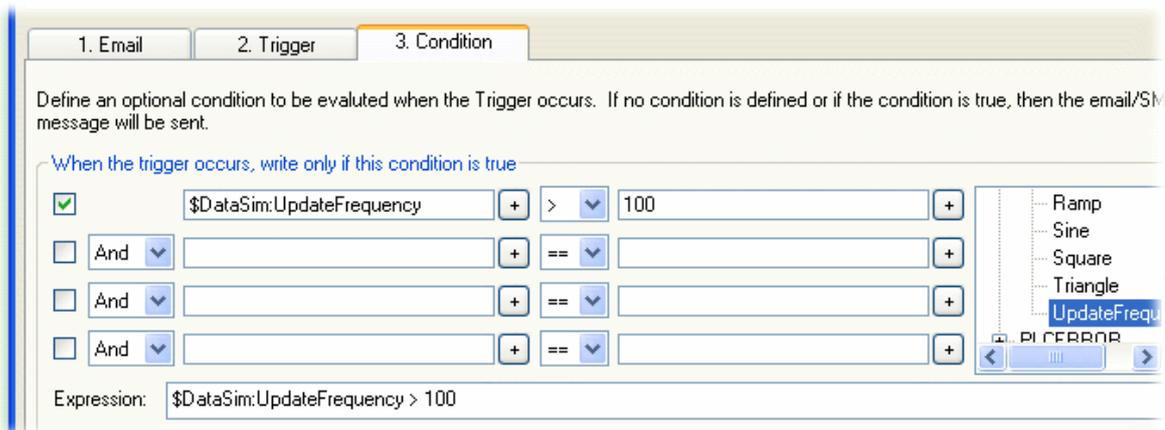


您可以為該觸發程序選取任何點，包括寫入的點，如我們範例中的DataSim:Sine。欲了解觸發程序的更多資訊，請參閱[章節 10.5](#)，“指派一個觸發程序”。

指派一個條件以及設定動作

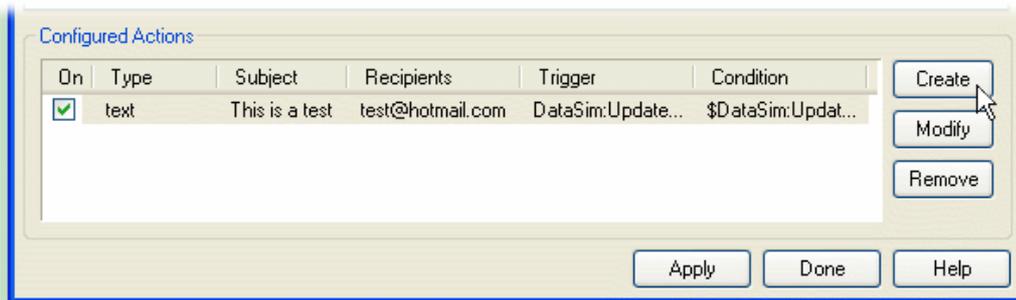
在這個範例中，我們將點DataSim:UpdateFrequency的觸發程序限制為只在值超過100變更。

1. 選取頁籤**3. Condition**。
2. 在第一行中勾選核取方塊。
3. 在點選取器中，展開DataSim Data Domain並選取點UpdateFrequency。
4. 點擊右側欄位的+按鈕。文字\$DataSim:UpdateFrequency應會填入方框裡。
5. 在下拉式方塊中，選取>運算子。
6. 在右側欄位中，輸入數字100。您的螢幕現在應會顯示如下:



您已設定好條件，下方的expression顯示會傳送什麼給Gamma，它是DataHub的內部指令碼引擎。

7. 接著移至**Configured Actions**方框並點擊**Create**按鈕。



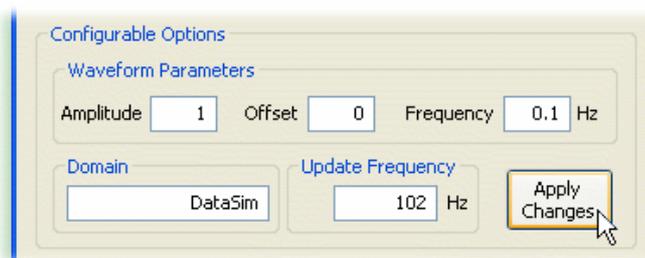
新的設定動作應該會顯示在清單中。這也是您所有動作的摘要資訊。當此清單中的設定動作被選取，您可以在任何頁籤中進行變更以及使用**Modify**修改。您也可以使用**Create**按鈕複製一個設定動作，或是使用**Remove**按鈕移除它。欲了解有關設定動作的更多資訊，請參閱**章節 10.7**，“設定動作”

8. 點擊**Apply**按鈕以啟用設定的動作。現在讓我們看看它是如何運作。

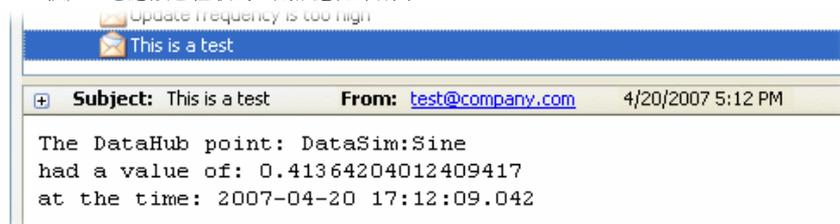
觸發並發送一封Email

您剛剛設定的動作會造成任何時候，只要DataSim Update Frequency變更到大於100的值，DataHub就會發送一封email。欲測試該指令碼，您需要在DataSim中變更該值藉以觸發該指令碼。

1. 在DataSim，按下**More...** 按鈕以檢視**Configurable Options**。
2. 將**Update Frequency**變更至大於100的數字，接著點擊**Apply Changes**按鈕以認可變更。



3. 請檢查收件者的email帳戶，您應該已經收到一則訊息如下所示:



每次您在DataSim中輸入一個數字大於100的新Update Frequency值時，DataHub指令碼就會發送出類似訊息。

您已經設定並測試一個動作，這個動作無論何時，只要UpdateFrequency點的值變更超過100，它就會發送一封email，其內容為DataSim資料網域中點Sine的名稱、值和時間戳記。現在您可以設定其他的email以發送您個人的簡訊或是HTML頁面。該章節的其餘小節有此界面的更多細節說明。

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

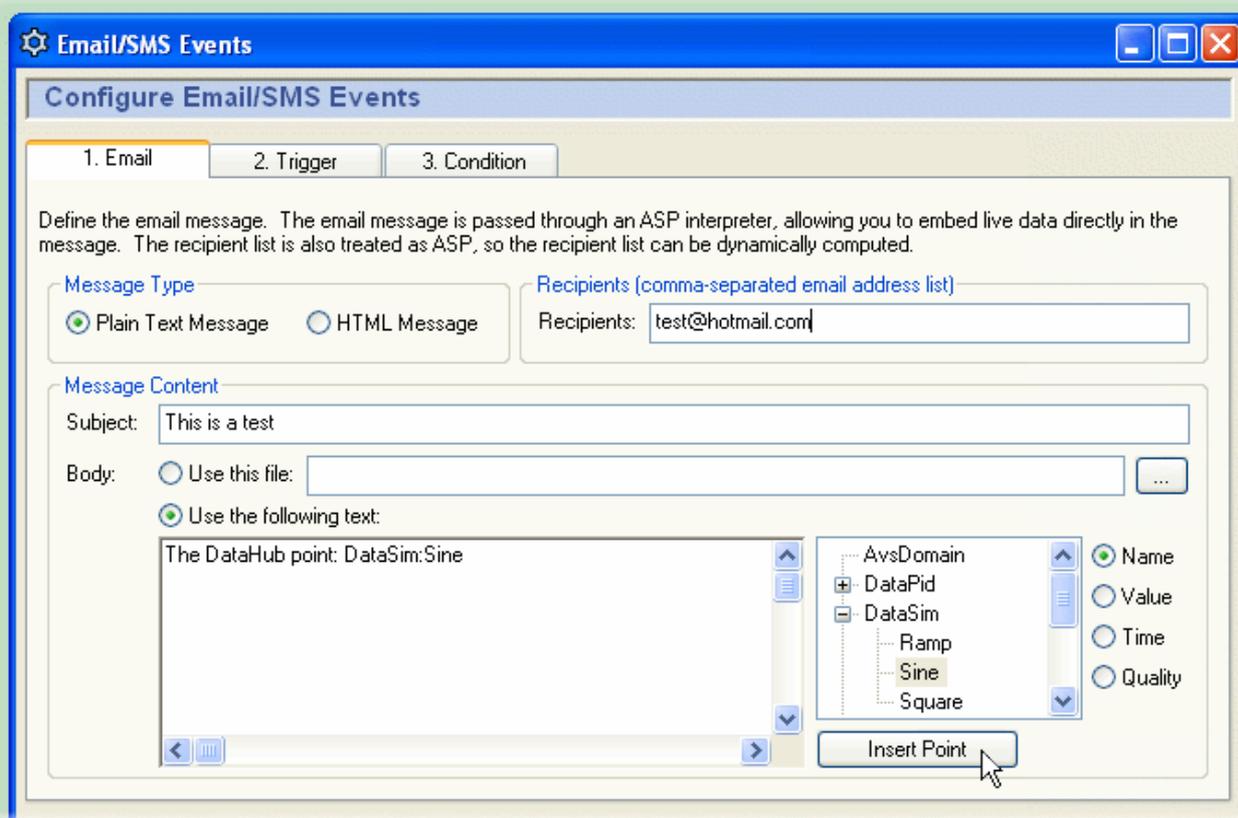
10.4. 定義Email訊息

欲發送一封email，您需要定義email訊息的類型，其收件者、標題和訊息內文。這是在Email/SMS Events視窗中完成的，您可以用此方式存取：

1. 在Cogent DataHub Properties視窗中，請選取**Email/SMS**  **Email/SMS** 。
2. 在**Configure Email**區塊中，按下**Configure** button按鈕。



3. 選取**1. Email**頁籤並設定您的email如下所述。



訊息類型

- **Plain Text Message** 在來源檔案寫入或是在此介面中輸入時，發送文字訊息。會在訊息發送時指派資料點值。
- **HTML Message** 用HTML檔案形式發送此介面的來源檔案或項目。會在訊息發送時指派資料點值。

收件者

這可以是單一的email位址，或是使用逗號分隔每個位址的email位址清單。位址可以是兩種類型之一：

- username@datadomain.com
- User Name <username@datadomain.com>

訊息內容

- **Subject** 輸入訊息的主旨。

- **Body** 您可以使用來自檔案的訊息，或是在編輯方框中撰寫一個訊息。
 - **Use this file:** 讓您插入想要將其作為email內容發送的檔案名稱。這並不是附件，而是訊息的內文。按下... 按鈕以瀏覽您所需的檔案。欲檢視一些HTML的範例檔案，請參閱**章節 10.8**，**“HTML訊息範例”**。
 - **Use the following text:** 讓您撰寫並編輯訊息的內文。欲在點選取器清單中插入點的名稱、值、時間戳記或是品質，請依照需求選取**Name**、**Value**、**Time**或**Quality**按鈕。接著點擊**Insert Point**按鈕。DataHub會將email中所需輸出點名稱的正確語法插入您的文字。



如果您想要發送來自一個檔案的訊息，您仍然可以使用文字編輯器的簡易介面建立它。在編輯器中撰寫訊息，接著複製並將其儲存成一個檔案。

DataHub點的值、時間和質量屬性是使用特殊語法存取的。當您按下**Insert**按鈕時，這會自動套用到文字編輯器。下列是供您參考的語法：

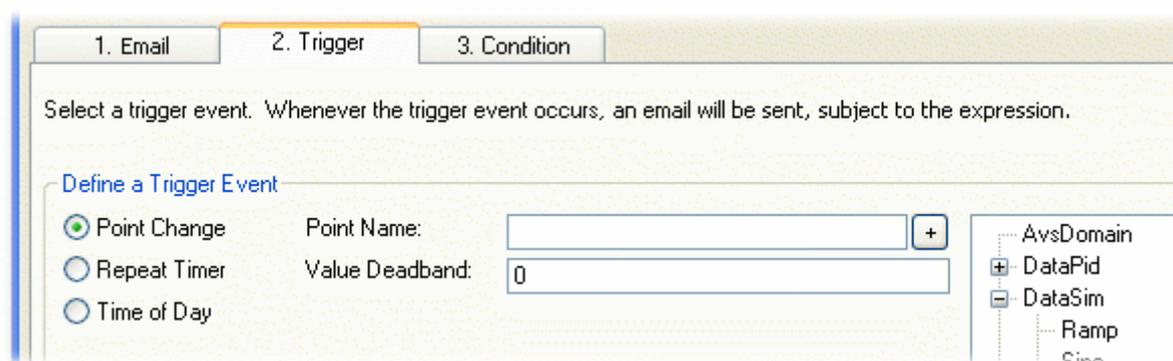
按鈕	語法	範例
Name	<code>domainname:pointname</code>	DataSim:Sine
Value	<code><%= \$domainname:pointname %></code>	<code><%= \$DataSim:Sine %></code>
Time	<code><%= PointTimeString (#\$domainname:pointname) %></code>	<code><%= PointTimeString (#\$DataSim:Sine) %></code>
Quality	<code><%= PointQualityString (#\$domainname:pointname) %></code>	<code><%= PointQualityString (#\$DataSim:Sine) %></code>

在此語法中，特殊符號的使用方法如下：

Character	Use
<code><% ... %></code>	附上的表達式會由Gamma，DataHub指令碼語言進行評估。
<code>\$</code>	替Gamma指出這是一個DataHub點名稱。
<code>PointTimeString()</code>	Gamma函式，它以一種容易閱讀的格式傳回DataHub點的時間戳記。
<code>PointQualityString()</code>	Gamma函式，它傳回DataHub點的品質，以文字字串的方式。
<code>#</code>	保護DataHub點免於被Gamma評估，直到函式被呼叫。

10.5. 指派一個Trigger

Trigger是個事件，它會導致email被發送。觸發事件可以是一個點值變更、計時器事件或是日曆事件其中之一。您也可以為每封email指派不同的Trigger，或者是為任意數量的email指派相同的Trigger。一個動作可以被設定執行在每個觸發事件上，或者您也可以指派觸發條件，無論何時Trigger發生時，它都會評估以確定是否執行動作。



三種Trigger類型如下:

- **Point Change** 無論何時，只要特定的觸發點變更，就會觸發。
 1. 輸入點的名稱到**Point Name**對話框中，或在右側的樹狀資料點擊+號展開以選取點。
 2. (可選)如果您想要過濾掉多餘的資料就輸入一個值的死區。您輸入的號碼會指定一個高和低(加或減)的範圍。任何落於該範圍內的值變更不會導致引發此Trigger。正向或負向的變化大於此值會啟動此Trigger，並會讓此資料行被寫入。

 欲建立一個能自動重設的Trigger，請參閱[章節 10.6](#)，“設定觸發條件”裡的**自動重設的Trigger**。

- **Repeat Timer** 循環觸發，每次秒數流逝時。
- **Time of Day** 在您指定的時間裡觸發。您可以輸入：
 - 一個數字，指出一個特定值。例如，seconds欄位填入0會造成只在分鐘的第0秒觸發事件。
 - 一系列數字，用逗號分隔。例如，在minutes欄位裡輸入0,15,30,45會指出應該在每個小時觸發事件，以及在每個小時的15分、30分和45分之間觸發事件。
 - 一個數量的範圍，由破折號分隔。例如在hours欄位中輸入8-18會指出應該在8 a.m.到6 p.m的每個小時中觸發事件。範圍可以混合列表，如0,4,8-16,20。
 - 星號(*)指出應該要為欄位裡的每個可能的值觸發事件。例如，在seconds欄位填入*號會造成每秒觸發事件。hour欄位填入*號會造成每一個小時觸發事件。

 欲定期紀錄一週裡指定日期的紀錄，請參閱[章節 10.6](#)，“設定觸發條件”。

欄位的範圍是:

Year: 1970-* **Hour:** 0-23
Month: 1-12 **Minute:** 0-59
Day: 1-31 **Second:** 0-59

 年份和月份在此是輸入不同的，至於Gamma localtime函式，如[時間條件](#)所述。

範例:

- 輸入項目為:

Year:	*	Hour:	8
Month:	*	Minute:	45
Day:	*	Second:	0

會造成每天、每個月以及每一年的8:45發送email。

- 輸入項目為:

Year:	*	Hour:	*
Month:	*	Minute:	0
Day:	15	Second:	0

會造成每個月、每一年第15天的每個小時發送一次email。

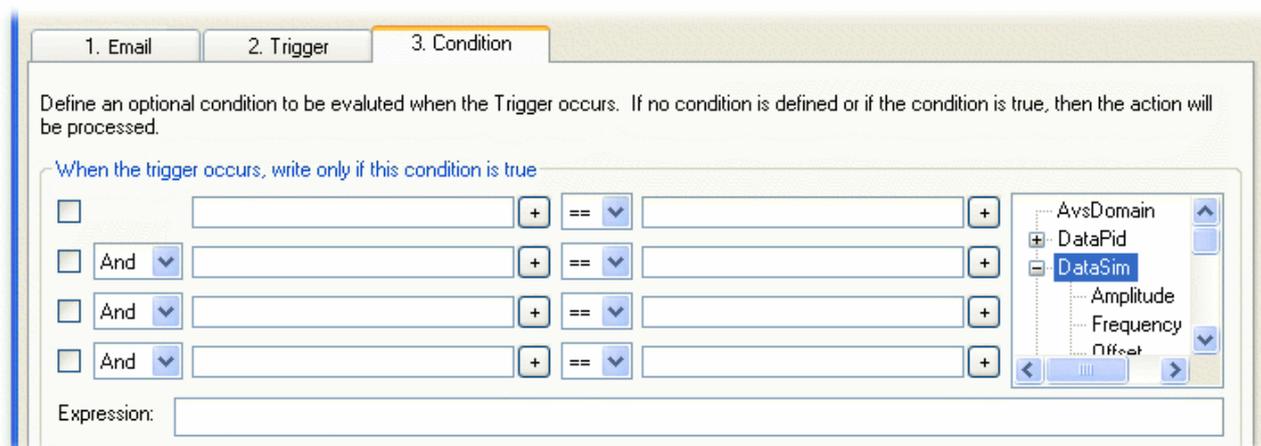
- 輸入項目為:

Year:	*	Hour:	8,10,12,14,16,18
Month:	*	Minute:	0-4
Day:	*	Second:	0

會造成8 a.m.和6 p.m.之間每兩個小時的每5分鐘發送email。

10.6. 設定觸發條件

只要Trigger發生，就會被評估以決定是否執行該動作。



根據以下指南填入條件。勾選條件旁的方框以套用。當您輸入其它項目時，相對應的 *Gamma* 程式碼會出現在顯示中。Gamma 是 DataHub 的內建指令碼語言。在 **Expression** 方框中出現的指令碼是由 Gamma 執行的實際程式碼。"And" 和 "Or" 運算子 (& 和 |) 的優先順序是 And，接著才是 Or。

點值的條件

點的名稱可以被輸入比對雙方的任何一方。它們可以從樹狀資料清單中選取，或者是輸入。每個點名稱的前面需要有一個錢幣符號 (\$) 以指示 Gamma 其為 DataHub 點。您可以把數值放進比對的任何一方。

當您在 **condition** 欄位中輸入一個點名稱，點的現值會被用在評估。例如，您可以定義一個條件，規定只要觸發事件發生時，若其他點值在特定範圍之內，動作才會被執行。

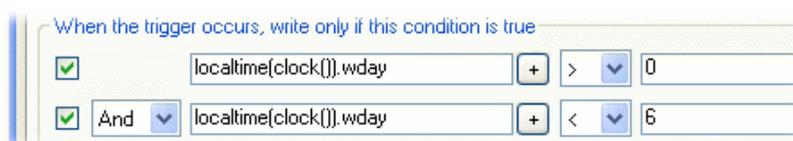
這裡有三個可用的自動變數可與點值一起使用：

- `lasttrigger` - 該 Trigger 最後一次觸發後，Trigger 的點值。
- `thistrigger` - 目前 Trigger 的點值。
- `lastevent` - 最後一次實際執行該事件之 Trigger 的值。

時間條件

當訊息被發送時，這能提供一個額外的方法來限制時間、日期、月份等等。除了在 Trigger 裡的選項之外，您還有 **day-of-week** 條件陳述式，根據一星期內的特定日子，它可給予您更彈性的事件。這些將能與任何類型的觸發事件共同運作。

您可以使用 Gamma 函式 `clock` 和 `localtime` 指定一個星期裡特定的某幾天。例如，這些輸入項目：



會建立此 Gamma 程式碼：

```
(localtime(clock()).wday > 0 && localtime(clock()).wday < 6)
```

這會造成只在週一至週五之間紀錄資料。函式 `localtime` 返回一個類別，其成員包括有關資料的資訊，如下：

- `.sec` 分鐘過後的秒數 (0 - 59)。
- `.min` 小時過後的分鐘數 (0 - 59)。
- `.hour` 午夜過後的時數 (0 - 23)。
- `.mday` 月份的天數 (1 - 31)。

- .mon 從1月開始的月份數量(0 - 11)。
 - .year 從1900年開始的年份數量。
 - .wday 從星期日開始的天數(0 - 6)。
 - .yday 從1月1日開始的天數(0 - 365)。
 - .isdst 如果是在日光節約時間影響之下，為1，如果沒有影響，為0，如果資訊無法供使用，為一負數。
-  年份和月份都輸入和Time of Day觸發條件不同的時間，如[章節 10.5](#)，“指派一個Trigger”所述。

這裡有兩種可用的自動變數可與點值一起使用：

- lasteventtime - 最後一次事件被執行的時間，UNIX時間。
- curtime - 現在的UNIX時間。

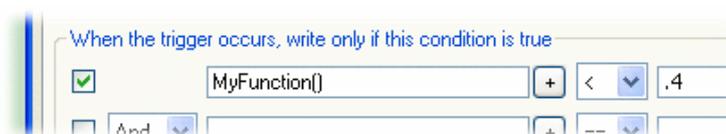
客製化條件

如果您需要滿足的條件已經超出該介面的範圍，您可以使用Gamma函式實際表達出您所需的任何條件。接著您可以插入函式到其中一個條件方框裡，並根據函式裡的傳回值設定條件。

欲做到這一點，您可以建立一個DataHub指令碼(.g檔案)其中只包含您在條件中會使用到的函式，沒有任何類別或方法。例如，以下是此類型檔案的完整內容，名為MyCondition.g:

```
function MyFunction ()
{
  myvalue = $DataSim:Sine;
  princ("Value when the trigger fired: ", myvalue, "\n");
  myvalue;
}
```

這個函式打印出DataSim:Sine點的值，並傳回其值。我們可以從介面裡的其中一個條件方框呼叫此函式，以便使用該函式作為一個條件，如下所示：



當Trigger觸發時，會呼叫MyFunction，而且會檢查傳回值，看它是否小於.4，如果是的話，資料會被紀錄下來。

一個自動重設的Trigger

該指令碼可以將任何DataHub點變成一個自動重設的Trigger。欲使用它，首先您需要載入並執行TriggerFunctions.g指令碼(如下所示，並包含在安裝文件裡)。接著，如果您將此公式：

```
HighWithReset($default:TriggerPoint) != nil
```

放進條件方框中，每當TriggerPoint在DataHub中變更為一個非零的數字，就會引發您的Trigger。指令碼會等待一個millisecond，接著將TriggerPoint重設為0。第二個函式運作方式相同，但它會在變更為0時引發，而不是變更至非0時引發。

TriggerFunctions.g

```
/*
 * This file contains handy functions to perform more complex condition
 * handling in the Condition tab of the data logging and email interfaces.
 */

/*
 * Test a trigger point for a non-zero value. If the point is non-zero,
 * create a delayed event to reset the point to zero, and return true,
 * indicating that the condition has succeeded and the action should proceed.
 * If the value is 0, then simply return nil indicating that the action
 * should not proceed. We need to test for zero because when we reset the
 * trigger point to zero a second data change event will occur.
 *
 * The argument is unevaluated, so the condition should look like this:
 * HighWithReset($default:TriggerPoint) != nil
 */
```

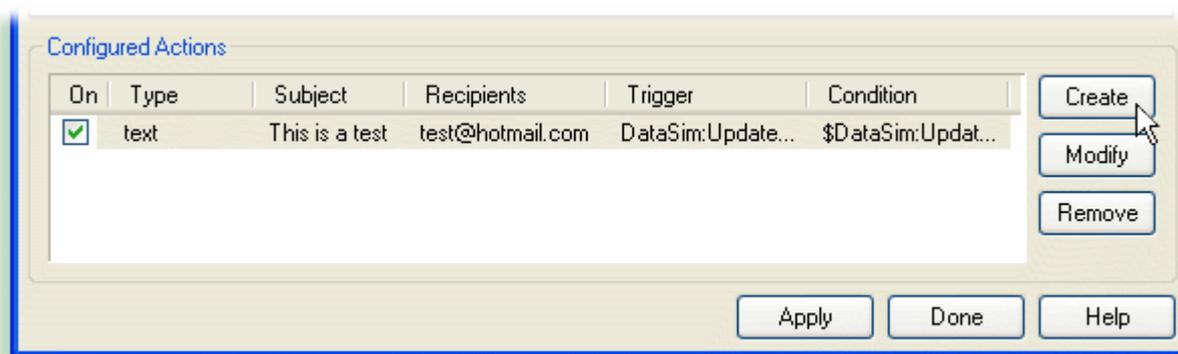
```
function HighWithReset(!triggerpoint)
{
  local value;
  if (!undefined_p(value = eval(triggerpoint)) && value != 0)
  {
    after(0.001, `setq(@triggerpoint, 0));
    t;
  }
  else
  {
    nil;
  }
}

/*
 * This is the inverse of HighWithReset (see above). If the trigger point
 * is zero, perform the action and set the trigger point to 1. If the
 * trigger point is non-zero do nothing and return nil.
 */
function LowWithSet(!triggerpoint)
{
  local value;
  if (!undefined_p(value = eval(triggerpoint)) && value == 0)
  {
    after(0.001, `setq(@triggerpoint, 1));
    t;
  }
  else
  {
    nil;
  }
}
```

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

10.7. 已設定的動作

已設定的動作會造成指定的email根據一個Trigger以及可選的條件被發送。這是此介面中設定動作的最後結果。Configured Actions清單顯示您所設定的動作，並允許您建立、修改或移除動作，以及將它們開啓或關閉。



設定動作的清單顯示您已經設定的動作。從清單中選取一個現有的動作並且自動在**Email**、**Trigger**和**Condition**頁籤中填入資訊。勾選或取消勾選左側**On**方框可以讓您開啓動作或關閉動作。

Create按鈕 為目前輸入到**Email**、**Trigger**和**Condition**頁籤裡的資訊建立一個動作。如果您選取一個已設定動作的同時按下**Create**按鈕，它會建立一個該設定動作的複製，並將其新增至清單中。這是快速設定相同動作的方法。

Modify按鈕 使用目前輸入**Email**、**Trigger**和**Condition**頁籤中的資訊來覆寫選取的已設定動作。

Remove按鈕 移除所有設定動作。

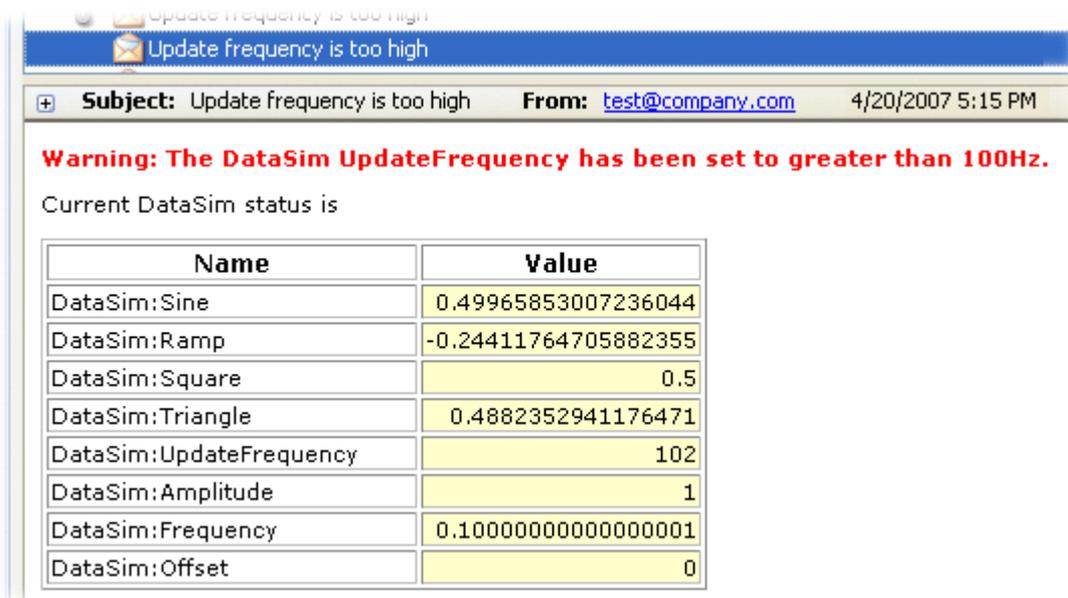
一旦建立或修改一個設定動作，變更不會立即生效，直到您點擊**Apply**或**Done** button按鈕。

10.8. HTML訊息範例

發送一個HTML訊息就如同在Email頁籤上點擊HTML Message按鈕一樣簡單。至於Message Source，您可以選取一個檔案或將HTML程式碼直接寫入文字輸入方框。這裡有兩個範例，說明您要如何將資料嵌入HTML訊息，使用ASP來源檔案。

10.8.1. 包含內嵌資料點的HTML訊息

這是個ASP檔案的範例，它將最新的資料從DataHub點嵌入HTML資料表。ASP檔案命名為EmbedPoints.asp，並與Cogent DataHub/etc/目錄裡的DataHub一起安裝。如果DataHub設定成發送該檔案作為訊息內文，而且DataSim's UpdateFrequency改變，比如說，變成102，DataHub會email一封訊息如下：



ASP檔案的內容

EmbedPoints.asp檔案的內容如下：

```
<html>
<style>
BODY, P, TD{
background-color : White;
font-family : Verdana, Geneva, Arial, Helvetica, sans-serif;
font-size : 8pt;
}
TH{
font-family:Verdana, Arial, Helvetica, sans-serif;
font-size: 9pt;
font-weight: bold;
background-color: #23cce6fe;
}
.highlight{background-color: #FFFFCC; text-align:right;}
.warning{color: #FF0000; font-weight: bold;}
</style>
<body>
<!--
      This is a simple example of an HTML template file which contains
      embedded point values from the DataHub.
-->
<p></p>
<div class="warning">Warning: The DataSim UpdateFrequency has been
      set to greater than 100Hz.</div>
<p></p>
Current DataSim status is
<p></p>
<table border="1">
  <tr>
```

```

    <th width="180">Name</th>
    <th width="80">Value</th>
</tr>
<tr>
    <td>DataSim:Sine</td>
    <td class="highlight"><%= $DataSim:Sine %></td>
</tr>
<tr>
    <td>DataSim:Ramp</td>
    <td class="highlight"><%= $DataSim:Ramp %></td>
</tr>
<tr>
    <td>DataSim:Square</td>
    <td class="highlight"><%= $DataSim:Square %></td>
</tr>
<tr>
    <td>DataSim:Triangle</td>
    <td class="highlight"><%= $DataSim:Triangle %></td>
</tr>
<tr>
    <td>DataSim:UpdateFrequency</td>
    <td class="highlight"><%= $DataSim:UpdateFrequency %></td>
</tr>
<tr>
    <td>DataSim:Amplitude</td>
    <td class="highlight"><%= $DataSim:Amplitude %></td>
</tr>
<tr>
    <td>DataSim:Frequency</td>
    <td class="highlight"><%= $DataSim:Frequency %></td>
</tr>
<tr>
    <td>DataSim:Offset</td>
    <td class="highlight"><%= $DataSim:Offset %></td>
</tr>
</table>
</body>
</html>

```

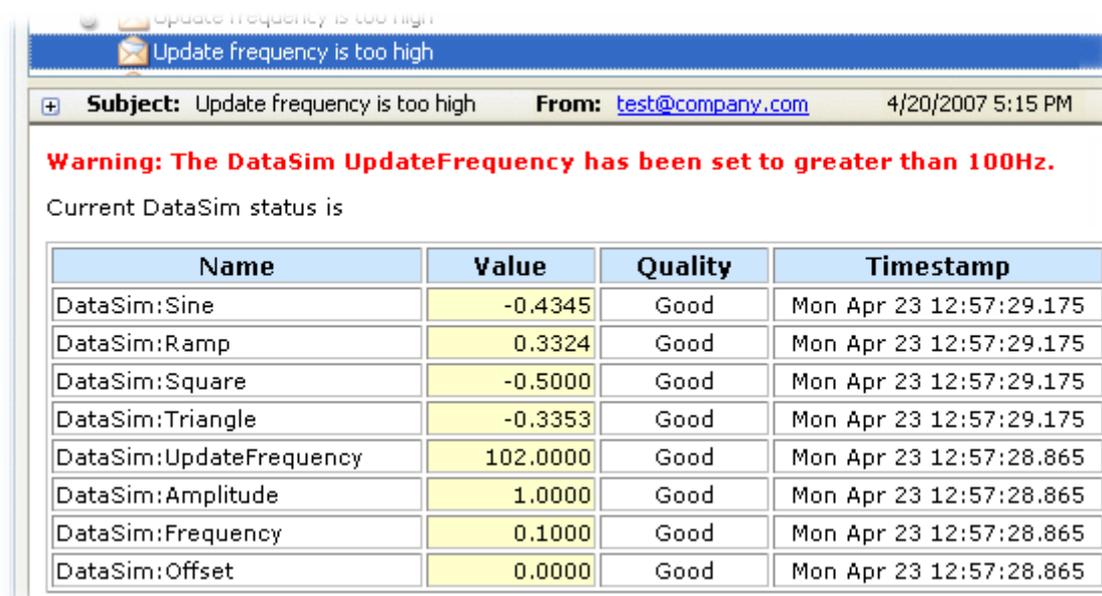
檔案包含HTML程式碼，也穿插著 *Gamma* 程式碼。**Gamma** 是 Cogent DataHub 的指令碼語言。Gamma 程式碼通常用來定義 DataHub 點的值，使用下列語法：

```
<%= $domainname:pointname %>
```

尖括弧和百分比符號(<% ... %>)替 DataHub ASP 解譯器指出此為 Gamma 程式碼。等號(=)會告知 Gamma 評估表達式，而貨幣符號(\$)告知 Gamma 此為 DataHub 點。

10.8.2. 包含使用程式碼建立之 Table 的 HTML 訊息

這是一個 HTML 訊息的範例，隨附使用程式碼建立的資料表，而不是明確地撰寫出。使用程式碼讓資料格式化以及資料表的變更變得更靈活。程式碼是寫入一個名為 `CreateTable.asp` 的 ASP 檔案，它與 DataHub 一同安裝，在 Cogent DataHub/etc/目錄裡。如果 DataHub 是設定為發送此檔案作為訊息內文，以及 DataSim 的 UpdateFrequency 是變更至，比如說 102，DataHub 就會 email 此訊息：



ASP的內容

CreateTable.asp檔案的內容如下:

```
<html>
<style>
BODY, P, TD{
    background-color : White;
    font-family : Verdana, Geneva, Arial, Helvetica, sans-serif;
    font-size : 8pt;
}
TH{
    font-family:Verdana, Arial, Helvetica, sans-serif;
    font-size: 9pt;
    font-weight: bold;
    background-color: #cce6fe;
}
.highlight{background-color:#FFFFCC; text-align:right}
.warning{color: #FF0000; font-weight: bold;}
</style>
<body>
<p></p>
<div class="warning">Warning: The DataSim UpdateFrequency has been set
to greater than 100Hz.</div>
<p></p>
Current DataSim status is
<p></p>
<table border="1">
    <tr>
        <th width="180">Name</th><th width="80">Value</th>
        <th width="80">Quality</th><th width="160">Timestamp</th>
    </tr>
<%
require ("Time");
require ("Quality");

try
{
    local          v, q, tm, ts, info;

    with pt in [ #DataSim:Sine, #DataSim:Ramp, #DataSim:Square,
                #DataSim:Triangle, #DataSim:UpdateFrequency,
                #DataSim:Amplitude, #DataSim:Frequency, #DataSim:Offset ] do
    {
        info = PointMetadata (pt);
        v = eval (pt);
```

```

    if (!number_p(v)) v = 0;
    q = GetQualityName (info.quality);
    ts = PointGetUnixTime (pt);
    tm = format ("%19.19s.%03d", date(ts), (ts % 1.0) * 1000);
%>
    <tr><td><%= pt %></td>
      <td class="highlight"><%= format("%.4f",v) %></td>
      <td align="center"><%= q %></td>
      <td align="center"><%= tm %></td></tr>
<%
  }
}
catch
{
  princ (_last_error_, "\n");
  print_stack (nil, _error_stack_);
}
%>
</table>
</body>
</html>

```

檔案包含HTML程式碼，也穿插著 *Gamma* 程式碼。**Gamma** 是 Cogent DataHub 的指令碼語言。**Gamma** 程式碼通常用來定義 DataHub 點的值，使用下列語法：

```
<%= $domainname:pointname %>
```

尖括弧和百分比符號(<% ... %>)替 DataHub ASP 解譯器指出此為 **Gamma** 程式碼。等號(=)會告知 **Gamma** 評估表達式，而貨幣符號(\$)告知 **Gamma** 此為 DataHub 點。其它用於此範例的 **Gamma** 陳述式和函式包括 **require**、**try**、**local**、**with**、**if**、**format**、**catch**、**princ** 和 **print_stack**。函式 **GetQualityName** 和 **PointGetUnixTime** 分別是來自需要的檔案 **Quality.g** 和 **Time.g**。